ISBN 978-82-326-6545-7 (printed ver.)
ISBN 978-82-326-6169-5 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (electronic ver.)

Doctoral thesis

Madeleine Lorås

Doctoral theses at NTNU, 2021:368

NTNU
Norwegian University of
Science and Technology
Thesis for the degree of
Philosophiae Doctor
Faculty of Information Technology
and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Doctoral theses at NTNU, 2021:368

Madeleine Lorås

Understanding the Relation Between Study Behaviors and Educational Design

Research in Computing Education

NTNU
Norwegian University of
Science and Technology

Madeleine Lorås

Understanding the Relation Between Study Behaviors and Educational Design

Research in Computing Education

Thesis for the degree of Philosophiae Doctor

Trondheim, December 2021

Norwegian University of Science and Technology
Faculty of Information Technology
and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology
and Electrical Engineering
Department of Computer Science

ISBN 978-82-326-6545-7 (printed ver.)
ISBN 978-82-326-6169-5 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (electronic ver.)

Doctoral theses at NTNU, 2021:368

PRINTED IN
NORWAY
NO - 1598

NORDIC SWAN ECOLABEL
Printed matter
2041 0731

Printed by Skipnes Kommunikasjon AS

Abstract

Important learning happens outside organized lectures and labs; however, much of the interaction between these educational design constructs and how students study is unknown. This thesis aims to understand how knowledge about computing students' study behavior can help us design first-year undergraduate computing programs. Previous research has looked at individual courses and specific tools, but the holistic perspective across courses and classes is somewhat missing. Furthermore, the inconsistent use of study behavior terminology and lacking tools to describe educational design makes it challenging to compare findings.

This PhD research took a closer look at the first year of two computing programs, examining the student experience and the relation to all levels of the educational design - from admission systems to course assignments. Through a mixed-method approach in three phases, this research used questionnaires, interviews, and document analysis to further our understanding of how educational design parameters affect how students study.

The results include a mapping of Norwegian computing education programs and a systematic literature review of study behaviors in computing education, producing a framework of educational design elements and a taxonomy of study behaviors. Together, these contribute to an improved understanding of the relationship between study behavior and educational design parameters in computing education and identifying the room for action for educators. Furthermore, a comprehensive investigation of the whole first year found that schedules, assignments, and campus layout facilitates how, when and where students study. A central result was the definition and characterization of the student-driven learning environment, which is based on the individual students' perspective and describes how they navigate the educational design constructs across courses within a program.

Lastly, the findings from this thesis encourage educators, policymakers, and students to consider shifting the focus slightly from the quantity to the quality of learning by better understanding how students study. Re-examining why we do things based on updated research and theories is an important first step. Every parameter and variable should be questioned, looking for the room for action. In addition to increasing the understanding of computing students, this work also contributes to the knowledge about how to understand computing students.

Sammendrag

Det foregår mye læring mellom forelesninger og i diverse kroker på campus, men mye av denne interaksjonen mellom utdanningens utforming og studentenes atferd vet vi lite om. Denne doktorgradsavhandlingen tar sikte på å forstå hvordan kunnskap om studenters studieatferd kan hjelpe oss med å utforme førsteårs IT-studieprogram. Tidligere forskning har fokusert på mange spesifikke tema og verktøy, men det helhetlige perspektivet på tvers av emner og kontekster er mindre utforsket. Det er også en utfordring at forskningen så langt bruker inkonsekvent terminologi for studieatferd og konteksten den gjøres i er mangelfullt beskrevet.

Forskningen presentert i denne avhandlingen ser nærmere på IT-studieprogrammer ved Norges teknisk-naturvitenskapelige universitet (NTNU). Fokuset har vært på studentenes reise gjennom det første året og hvordan de interagerer med ulike elementer i utdanningens utforming, fra opptakssystem til oppgaveløsning. Gjennom kombinerte metoder i tre faser har dette prosjektet gjennomført spørreundersøkelser, intervjuer og dokumentanalyser for å kunne videreutvikle vår forståelse av hvordan utdanningens utforming påvirker studentenes studieatferd.

Resultatene fra denne forskningen inkluderer en kartlegging av IT-programmer i Norge og en systematisk gjennomgang av forskning på studieatferd i IT-utdanning. Ut ifra disse ble det utviklet et rammeverk for å beskrive utdanningens utforming og en klassifikasjon av studieatferd. Sammen utgjør disse nyttige verktøy for å beskrive og forstå sammenhengen mellom hvordan studentene studerer og utdanningens utforming. Videre fant en omfattende undersøkelse av studentenes erfaringer gjennom hele det første året at timeplaner, frister og oppsettet på campus fasiliteter når, hvor og hvordan studentene studerer. Denne relasjonen defineres av det studentdrevne læringsmiljøet, som baserer seg på den individuelle studentens perspektiv og beskriver hvordan de navigerer gjennom forskjellige elementer i utdanningens utforming på tvers av emner og program.

Funnene fra denne avhandlingen oppfordrer lærere, beslutningstakere og studenter til å flytte fokuset fra hva som blir lært til hvordan det blir lært gjennom å forstå studentenes studieatferd bedre. Aller først må vi revurdere hvorfor vi gjør som vi gjør basert på oppdatert forskning og teori. For å identifisere handlingsrommet bør det settes spørsmålstegn ved alle elementer i utdanningens utforming. I tillegg til å øke forståelsen av IT-studenters atferd så bidrar dette arbeidet også til å øke kunnskapen om hvordan vi skal forstå studentene.

Norsk-Engelsk ordliste for sentrale begreper:

Studieatferd - Study behavior | Utdanningens utforming - Educational design

IT-studieprogram - Computing program | Kombinerte metoder - Mixed-method

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) in partial fulfillment of the requirements for the degree of Philosophiae Doctor.

The PhD work was performed at the Department of Computer Science, NTNU, Trondheim, under the supervision of Associate Professor Trond Aalberg (main supervisor), Professor Guttorm Sindre, Associate Professor Hallvard Trætteberg and Professor Letizia Jaccheri (co-supervisors).

During this PhD project, I have been involved in the Excited Center for Excellent IT Education, led by Guttorm Sindre. Excited receives public funding through DIKU, the Norwegian Agency for International Cooperation and Quality Enhancement in Higher Education.

Acknowledgements

For the past four years, I have had the privilege of doing a PhD at NTNU. It has been quite a ride, with obligatory ups and downs. I am so proud of what I have accomplished, not only the research presented in this thesis but also all the interesting initiatives and events I have been a part of not visible in academic publications. There are a lot of people who have supported me on this journey, so here comes a lot of thank you's.

First and foremost, I am eternally grateful for my supportive and invested supervisors Trond Aalberg, Guttorm Sindre, Hallvard Trætteberg and Letizia Jacceri. My main supervisor Trond always knows exactly what to say to make me feel confident and knowledgeable, and I am so very thankful for his wisdom and insight throughout this process. Thank you Guttorm for taking the time to dive deep into my questions. Thank you Hallvard for all the smiles and great conversations. Thank you Letizia for the hugs and for being a role model to me and many others.

Secondly, I want to thank my family. Not just for the unconditional support over the last four years, but for always encouraging me to keep going and believe in myself. To my sister Amanda, I want it noted in this written and formal text, that you are the very best sister in the world. To my parents, thank you for raising me to be curious and independent and for always being there. And to extended family on both sides of the Atlantic, I am very grateful for your support. I especially want to thank the Americans for helping me with my goodly English skills.

I also want to extend my gratitude to all the NTNU people I have had the pleasure of getting to know over the years here. There are so many inspirational and supporting people in the NTNU system. You know who you are: Reidar Lyng, Geir Øien, Inge Fotland, Katja Hakel, Anne Borg, Ole Kristen Solbjørg, Kari Hag, Marius Irgens and all the LUR-people. This also includes my colleagues at IDI, thank you for all the interesting lunchtime chats. Furthermore, I want to especially thank John Krogstie for all the great afternoon discussions. And Kshitij Sharma for always being up for a coffee with way more methodological support than actual beverages. Others who deserve a special thank you are Emma Riese, Rabail Tahir, Katerina Mangaroska, Sofia Papavlasopoulou, Børge Haugset, Tore Sletten Langeland, Randi Holvik and Kristine Lund.

I am also grateful for the community of Excited PhDs and professors: Rune H, Monica, Birgit, Line, Robin, Ottar, Justyna and Abdullah. To my personal coach Vojislav Vujosevic, thank you for the pep talks. And thank you Gunhild Marie Lundberg, for asking so many good questions and for the support during those last months along with Elise and Beate. And to the fabulous Ida Sortland, there are no words to describe the incredible job you did helping me stay strong.

A very special thanks to my friends, who fed me, entertained me and generally kept me sane: Bendik, Tina, Merete, Simon, Ellen, Sindre, Anine, Aslak and many more. And to Helle, thank you for all the good times doing math homework, and also the more fun things we did.

Lastly, I want to extend my deepest gratitude to all the students who participated in the research at some point, and to all the students who emailed me, came by my office, or just engaged in discussions. A very special thanks to all the teaching assistants at Excited over the years, the Catch IDI gang and all my master students. Thank you for always reminding me why I did this PhD.

At this point, I have spent one-third of my life at NTNU. As a student, politician, teacher and colleague I have had the true pleasure of experiencing many sides of this institution. I can honestly say that I am a completely different person now. Am I a better person? Well, that is yet to be decided. But I certainly have more credits.

Thanks, NTNU, see you later!

List of Tables

List of Figures

Glossary

assignment The term assignment in this thesis includes any deliverable, task, course work or problem that is handed in by students to be graded or assessed in any way. First referenced on p. 19

CS1 This refers to the first introductory programming course taken by students. The second course is often referred as CS2. In addition, CS0 is often used to describe computing or programming courses or training offered before the introductory course. First referenced on p . 31

educator The term educator in this thesis includes anyone who teaches; professor (assistant, associate or otherwise), lecturer and docent. However, this definition does not include students who are employed part time as a teaching, learning or student assistants. First referenced on p. 3

program The term program in this thesis refers to wherever the students are enrolled. In some educational contexts, this might be a school of engineering or a major. When referring to a computing program, as in a compilable set of code, that will be specified. First referenced on p . 3

xiv

Acronyms

CS Computer Science. 38

GDPR General Data Protection Regulation. 34

GPA Grade Point Average. 26, 37–39, 55

IDE Integrated Development Environment. 20

IT Information Technology. 18

MOOC Massive Open Online Course. 10, 16

NOKUT Norwegian Agency for Quality Assurance in Education. 26

NSD Norwegian Centre for Research Data. 33, 35

NTNU Norwegian University of Science and Technology. 3, 28

NUCAS The Norwegian Universities and Colleges Admission Service. 26

PBL Project-Based Learning. 10, 16

RQ Research Question. 5, 24, 52, 63

SAL Students Approaches to Learning. 11, 14, 52

SDI Study Day Initiative. 32, 38, 57

SDLE Student-Driven Learning Environment. 15, 29, 41, 43, 57, 58, 60, 64, 67

SPQ Study Process Questionnaire. 28, 61

STEM Science, Technology, Engineering, and Mathematics. 4, 9, 31

TA Teaching Assistant. 31–33, 47, 63

xvi

Contents

Part I

Part I: Synopsis

Chapter 1

Introduction

Computing has become a major educational discipline, and there is increasing demand for qualified graduates [39]. Many approaches have been taken to address this challenge: increasing the diversity of recruitment and making computing more accessible to all [14], as well as a plethora of tools, activities, and interfaces to increase engagement, retention, and throughput [45]. The content and structure of computing education programs represent two aspects of the challenge. Another important aspect is the students and how they interact with, and relate to, the educational design.

The higher education system is built on student independence so that students learn to think and solve problems, develop themselves, become professionals, and grow as human beings throughout the process. Put extremely simply, educators teach, students study, and the outcome is expected to be learning. Educators commonly ensure that learning has occurred with various forms of assessment, often resulting in a grade that quantifies the level of knowledge achieved. The quality of student learning, however, should also receive attention. The quality of learning includes the intent and process of learning, with a focus on outcomes beyond the knowledge and skills measured in exams [12, 33]. Many internal processes and concrete actions take place when a student learns something. These processes and actions constitute the study behaviors of a student. Many researchers agree that study behaviors and noncognitive factors contribute strongly to students' performance and achievements [86, 105, 23].

At the Norwegian University of Science and Technology (NTNU), there are 11 undergraduate computing programs with approximately 650 new students enrolling every year. These programs are all designed and organized in different ways, which makes NTNU an interesting case to investigate. In general, the students are organized into four courses each semester. Some of the courses are small and aimed at computing students; however, more often than not, the courses include students from many different programs. The computing students take an introductory programming course together with over 3000 students. This open enrollment is an example of an educational design parameter that involves some challenges, especially when it comes to providing a stimulating academic learning environment, guaranteeing a sense of belonging, and ensuring that the learning outcomes are met. Making changes to the educational design in this specific example would involve at least 10 other Science, Technology, Engineering, and Mathematics (STEM) programs, an unknown amount of administrative scheduling, and agreement among educators and policy makers from many different fields. In other words, even if the need for improvement is apparent, the room for action may be limited.

This thesis project is part of the Excited Center for Excellent IT Education, at the Department of Computer Science at NTNU. The Excited Center aims to put Norway at the forefront of innovative IT education and to make IT a highly attractive study choice for young people. By focusing on increasing pre-university students' abilities to make informed decisions about studying IT, supporting students throughout their studies, and bringing them into contact with the industry, the Excited Center has over the past four years piloted many projects and implemented changes to educational designs at NTNU.

My motivation for this project stems from my own experiences as a student and a curiosity about the interaction between student life and learning in higher education. I started my "student career" in physics and mathematics and moved into computing only after a formative and inspiring experience in a mandatory programming course. During this transition, I noticed some differences in approaches between the disciplines, as well as between my peers and myself. As an active representative in the student government, I could look "behind the scenes" of higher education. I saw how policies and guidelines, as well as structural and financial constraints, drive educational design choices. When I entered the Excited Center to start my PhD, these experiences and observations were fundamental to my approach to researching first-year computing students.

1.1 Aims and Research Questions

Based on the issues and challenges outlined above, the purpose of this thesis is to help improve the understanding of effective and meaningful first-year computing education. Many initiatives exist in certain courses, there are tools for specific topics, and research is conducted on different student demographics; however, my research is positioned at an aggregate level. By examining the whole first year, not just the different courses and tools, I aimed to look at computing education in an integrated manner by basing my research perspective on studying as opposed to learning. Therefore, the scope of this research is limited to first-year computing education in Norway.

The research objective of this thesis is understanding *how knowledge about computing students' study behavior can help us design first-year undergraduate computing programs*. This objective is divided into the following four research questions:

RQ1: What are the characteristics of educational design in computing education?
RQ2: What is the state of knowledge about study behaviors in computing education?
RQ3: How does educational design impact study behavior during the first year of higher computing education?
RQ4: How can this knowledge be used to improve the educational design of first-year undergraduate computing programs?

The first research question aims to identify the relevant design elements within first-year computing education and to provide an overview of the Norwegian context in which this research was conducted. By defining parameters and linking them to theory, it becomes possible to compare the Norwegian and global educational contexts. The second research question looks at study behaviors and aims to summarize the state of knowledge within computing education. The third question further investigates the connections between educational designs and computing students' study behaviors. By exploring the role of educational design in computing students' study behaviors using the parameters identified in RQ1, we can gain a more comprehensive understanding of the situation. The last research question explores how the knowledge gained by investigating RQ1-RQ3 can be useful to educators, students, and leaders in computing education.

1.2 Research Approach

In an ideal world, all computing students would be self-driven learners who construct knowledge at their own pace, closely guided by teachers and peers. However, in the real world, structural limitations and unpredictable human elements frustrate this ideal situation. This thesis, therefore, uses a pragmatic research approach because I subscribe to the notion that different problems require different solutions. Taking a pragmatic approach means that every step of the research is guided by its own needs and goals and thus may employ approaches and tools from different fields [66, 76]. Accordingly, the ontology and epistemology of this research are defined by the theoretical lens of learning theories. Ontology is concerned with "what is," so the ontology of this research is that learning is the acquisition of competency, gained through study behavior as guided by a student's study processes, strategies, habits, and tactics. Furthermore, the pragmatic research paradigm and the mixed-method research approach determine how knowledge about student learning is acquired.

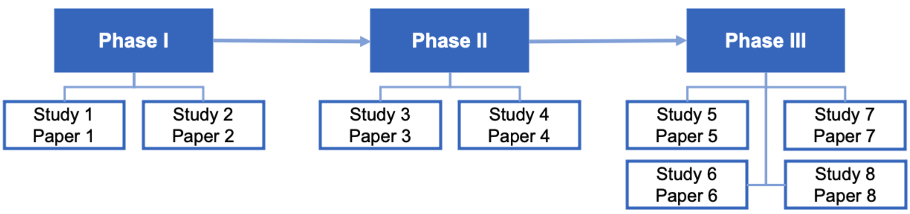


Figure 1.1: Overview of the research process and the corresponding papers

The thesis project had three phases. In Phase I, the focus was on getting to know the context and the student experience. Phase II concentrated on study behavior and the interaction with educational design. Finally, Phase III involved several individual studies and experiments, exploring different aspects of study behavior and educational design. Two study programs were the main focus: the bachelor's program in informatics (a three-year program) and the master's program in computer science engineering (an integrated five-year program). The word "program" is used here to describe the organization of students into a specific field of study, otherwise commonly referred to as major or school.

1.3 Research Contribution

This thesis is built on the eight papers listed below. Each paper corresponds to a phase in the research, as described in Figure 1.1.

Paper 1: Lorås, M., Sindre, G., & Aalberg, T. (2018). First Year Computer Science Education in Norway. *Proceedings of the Annual NOKOBIT Conference 2018*, 26.

Paper 2: Lorås, M., & Aalberg, T. (2020). First Year Computing Study Behavior: Effects of Educational Design. *Proceedings of the 2020 IEEE Frontiers in Education Conference (FIE)*, 1–9. DOI: 10.1109/FIE44824.2020.9274285

Paper 3: Lorås, M., Sindre, G., Trætteberg, H., & Aalberg, T. (2021). Study Behavior in Computing Education—A Systematic Literature Review. *ACM Transactions on Computing Education (TOCE)*, 22, 1. DOI: 10.1145/3469129

Paper 4: Lorås, M., & Aalberg, T. (2021). Characteristics of the Student-Driven Learning Environment in Computing Education.*Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE)*, 11–17. DOI: 10.1145/3430665.3456310.

Paper 5: Lorås, M., Haugset, B., & Trætteberg, H. (2021). The Importance of the Campus—A Study on the Effects of the Covid-19 Pandemic in a CS2 Course. *Proceedings of the 2021 IEEE Global Engineering Education Conference (EDUCON)*, 160-169. DOI: 10.1109/EDUCON46332.2021.9453910.

Paper 6: Hellem, V., & Lorås, M. (2020). The Effect of Mandatory Assignments on Students Learning Outcome and Performance in Introductory Programming Courses. *Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON)*, 704–712. DOI: 10.1109/EDUCON45650.2020.9125198

Paper 7: Lorås, M., & Aalberg, T. (2021). Creating Learning Environments Within the Constraints of Higher Education—A Case Study of a First-Year Computing Program. *Proceedings of the 2021 IEEE Global Engineering Education Conference (EDUCON)*, 170-177, DOI: 10.1109/EDUCON46332.2021.9454036.

Paper 8: Riese, E., Lorås, M., Ukrop, M., & Effenberger, T. (2021). Challenges Faced by Teaching Assistants in Computer Science Education Across Europe. *Proceedings of the 2021 ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE)*, 547–553. DOI:10.1145/3430665.3456304.

These papers made the following contributions:

Contribution 1: An outline of educational design levels, elements, and parameters in computing education in Norway.

Contribution 2: A summary and conceptualization of types of study behavior in computing education.

Contribution 3: An improved understanding of the relationship between study behavior and educational design parameters in computing education.

Contribution 4: A definition and characterization of the student-driven learning environment in computing education.

Contribution 5: Identification of the room for action for educators in computing education.

How the papers and their contributions are related to the research questions is summarized in Figure 1.2.

Papers	RQ1	RQ2	RQ3	RQ4	C1	C2	C3	C4	C5
Paper 1: First Year Computer Science Education in Norway	●				●				
Paper 2: First Year Computing Study Behavior: Effects of Educational Design			●			●			●
Paper 3: Study Behavior in Computing Education - A Systematic Literature Review		●					●		●
Paper 4: Characteristics of the Student-Driven Learning Environment in Computing Education			●			●		●	●
Paper 5: The Importance of the Campus - A Study on the Effects of the Covid-19 Pandemic in a CS2 Course			●	●		●		●	
Paper 6: The Effect of Mandatory Assignments on Students Learning Outcome and Performance in Introductory Programming Courses			●	●		●			
Paper 7: Creating Learning Environments Within the Constraints of Higher Education - A Case Study of a First-Year Computing Program			●	●		●			●
Paper 8: Challenges Faced by Teaching Assistants in Computer Science Education Across Europe			●	●		●			

Figure 1.2: Connections between research papers, research questions (RQs), and contributions (Cs)

1.4 Purpose and Structure of the Thesis

The purpose of this thesis is first and foremost to present the research project that constitutes this doctoral work. Furthermore, the thesis aims to be relevant to educators, policy makers, and students, in addition to researchers in the field. The thesis consists of two parts. Part I provides a synopsis of the research project. The current chapter describes the background, motivation, and aims of this thesis and outlines the approach, results, and contributions. The next chapter presents the theories and definitions underlying this research. Chapter 3 describes the computing education context, both globally and locally, including related research. Chapter 4 presents the research design, data collection, and analysis methods, and Chapter 5 summarizes the results. Lastly, Chapter 6 discusses the results of this research with respect to the research questions, contributions, and implications, and Chapter 7 includes some final remarks and suggestions for future work. Part II contains the collection of research papers included in this thesis.

Chapter 2

Theories and Definitions

The research questions concern the educational design and study behavior in computing education, focusing on the first year. These dimensions can be viewed through various theoretical lenses from many different fields. Education and learning have been theorized and researched over several centuries and have roots in philosophy, psychology, sociology, and social anthropology. Within STEM disciplines, these theories have been further adapted and developed to understand the students' learning experience to solve didactical challenges. This thesis is mainly based on theoretical frameworks from general educational theory. Related studies on aspects specific to computing education will be presented in Chapter 3.

2.1 Theoretical Lens

What learning is and how it happens have been thoroughly discussed and theorized over the years. Generally, such *learning theories* can be categorized under the paradigms of behaviorism, cognitivism, constructivism, and connectivism, although new perspectives and hybrid theories are constantly being proposed [2, 87, 36]. Within these theories, there are numerous models explaining teaching approaches, learning, and educational designs.

Behaviorism is considered to be the oldest theory and defines learning as a change in observable and measurable behavior [5]. Burrhus Frederic Skinner, an influential behaviorist, established the concept of operant conditioning based on the idea that rewards and punishments reinforce particular behavioral responses so that learning occurs [87]. Albert Bandura expanded the work of Skinner and others by introducing the concept of observational learning, implying that students can learn new information and behaviors by watching other people. Based on Bandura's social learning theory, the notion of self-efficacy and the theory of reciprocal determinism were developed, which state that a person's behavior influences and is influenced by personal factors and the social environment [3].

Within constructivism, learning is seen as an active process of constructing knowledge, and teaching is regarded as a process of supporting this construction of knowledge [30].

The opposite perspective, behaviorism, views learning as the acquisition of knowledge and teaching as communication [37]. Furthermore, constructivism is commonly classified into two related but complementary views: cognitive and sociocultural constructivism. Cognitive constructivism (often simply referred to as cognitivism) is based on the work by Jean Piaget [79] and Seymour Papert [77] and focuses on the individual. Cognitive constructivism emphasizes how a student constructs knowledge by making connections between new experiences and established ideas [37]. Sociocultural constructivism emphasizes that knowledge is created through social and cultural activation [36]. Theories that are rooted in this form of constructivism posit that students will engage more deeply with their learning process when they are actively involved and when learning takes place in a group [37]. Also drawing on sociocultural constructivism, Lev Vygotsky's [110] zone of proximal development postulates that learning awakens various internal developmental processes that operate only when one is working with others [37]. The support provided by educators and activities when learners grow within their zone is often referred to as scaffolding. Learning is a mediated process, progressing through dialogue with others. Furthermore, Etienne Wenger's theories on learning communities [111] and the situated learning theories of Jean Lave [54] also emanated from sociocultural constructivism.

With the introduction of the internet and modern technology in education, connectivism was introduced by George Siemens in the 2010s [93]. According to connectivism, knowledge is constantly shifting and changing within a network. Stephen Downes has been influential in the development of connectivist learning theory, emphasizing that learning consists of constructing and traversing the network of knowledge [29].

General higher education will always encompass a mixture of learning theories, and educators will use different teaching methodologies based on various learning theories [13, 92]. For example, traditional lectures are considered a behaviorist approach [5], while scaffolding is regarded as a sociocultural approach [84]. Project-Based Learning (PBL) is a typical cognitive method, whereas Massive Open Online Course (MOOC)s are connectivist methods. Following a pragmatic approach to research and education, this thesis acknowledges that these learning theories can explain different teaching and learning approaches [36, 13, 92], which also informs the discussion of the results Chapter 6.

2.2 Study Behaviors

The research on study behavior is somewhat of a terminology jungle [105]. A common procedure is to focus on one or two related aspects of study behavior, such as time management [20, 72] or motivation and habits [53]. The systematic literature review of study behavior in computing education addresses this terminology jungle and I therefore refer the reader to Paper 3 for a more in-depth presentation and discussion of the topic. However, it is still relevant for this synopsis to clarify some terms and definitions and present some context.

Tressel, Lajoie, and Duffy's review from 2019 defines study behavior as "any actions students make when preparing for, or taking part in, study-based activities" [105, p. 121]. Study behaviors can be further grouped into three categories based on the level of cognition: (1) the study process, which refers to the cognitive level of engagement with study activities; (2) study strategies, which represent the cognitive level of control over study activities; and (3) study habits and tactics, which relate to the consistency and actualization of study activities and students' individual learning tools.

2.2.1 Study Process: Student Approaches to Learning

The study process refers to the cognitive level of engagement with study activities. Early research into the study approaches of higher education students focused on prediction and general laws [11]. As a reaction to this research, toward the end of the 20th century several researchers developed the framework of student approaches to learning (SAL). A common thread was to focus on the quantitatively distinct ways students learn or engage in study activities [11]. A central point in this framework is that the learning approach refers to both the process and the intention of the student [33], further understood to include strategies and motives [11]. At this time, the old perspective on learning approaches was deemed too dependent on the specific context and content of the learning situation. In the "new" research, the central assumption was that there was consistency in approach across context and content [33].

The SAL framework was first introduced by Ference Marton and Roger Säljö in 1976 [65]. This group of researchers, known as the Gothenburg group, was instrumental in developing both the theory and the phenomenographic methodology. The Lancaster group, with Noel Entwistle and Paul Ramsden [34], further developed the theory and published the Approaches and Study Skills Inventory for Students questionnaire (ASSIST) in the 1980s [35].

According to the SAL theory, students' learning and studying processes can be categorized into deep and surface cognitive processing. The deep approach is an internally driven motivation and commitment to learning, in which the intention to extract meaning produces active learning. In contrast, the surface approach is externally driven and just involves coping with various tasks; it is considered a much more restricted learning process. More recently, Biggs described the surface approach as consisting of "activities of an inappropriately low cognitive level, which yield fragmented outcomes that do not convey the meaning of the encounter" and the deep approach as "activities that are appropriate to handling the task so that an appropriate outcome is achieved" [9, p. 42]. Biggs and colleagues developed a questionnaire to measure students' usage of the deep and surface approaches [10], which is often used to evaluate teaching initiatives and student learning approaches. The terms achieving and strategic are commonly used to describe students who employ both deep and surface approaches depending on what is required [34]. The revised two-factor Study Process Questionnaire has been adapted and validated across countries and cultures; for example, I and colleagues at the University of Agder adapted the questionnaire for the Norwegian context [114].

Although several researchers have adapted and revised the SAL framework over the years, one important distinction is the perspective on consistency across contexts. As Entwistle [33] accounts, the SAL framework originated as a theory to explain student learning across contexts. However, today, many researchers have landed on a middle ground between the viewpoint that different contexts and subjects require completely different learning approaches and the notion that there is one unifying theory. As Biggs and Tang [13, p. 28] describes it,

> *Students do have predilections or preferences for this or that approach, but those predilections may or may not be realized in practice, depending on the teaching context. We are dealing with an interaction between personal and contextual factors, not unlike the interaction between heredity and environment.*

Later in this synopsis, I will further explore the connection between approaches to learning, study behavior in general, and the environment.

2.2.2 Study Strategies: Metacognition and Self-Regulation

In this thesis, study strategies are defined as the level of cognitive control over study activities. This definition embraces the terms metacognition and self-regulation. Metacognition and self-regulation stem from higher education research focused on cognitive psychology. Like the SAL theory, cognitive psychology also developed a significantly new perspective at the end of the 20th century [89]. As Dale H. Schunk outlines in his contribution to a special issue on metacognition, self-regulation, and self-regulated learning in 2008, "cognitive theories shifted the focus of human functioning away from environmental variables and onto learners—specifically, how they encoded, processed, stored, and retrieved information. Rather than being passive recipients of information, learners were active seekers and processors of information" [89, p. 1]. As the theories on metacognition and self-regulation developed further, it has become common to differentiate metacognition as the mental knowledge behind human actions and self-regulation as the process of executing the actions [28, 81]. The exploration and application of these theories within higher education were led by Paul P. Pintrich, who developed the Motivated Strategies for Learning Questionnaire (MSLQ) [80, 88], which was later expanded by Zimmerman and Schunk [116].

2.2.3 Study Habits: Time Engagement and Tactics

Study habit is a loosely defined term in the literature [105, 23]. Tressel et al. [105] argue that study habits should be defined by the consistency of study behaviors, regularity in the use of study strategies, and the study environment. Accordingly, study habits are informed by study processes and strategies but are related to explicit behaviors. In this thesis, study habits are defined as the consistency and actualization of study activities. The interaction with the study environment has been left out of this definition because the environment and context are also important in other study behaviors.

An important aspect of study habits is that it is related to the activities students partake in when studying. Whereas study processes and strategies are related to cognitive processes, study habits and tactics are concrete and directly observable. Furthermore, study tactics are defined as "the individual learning tools a student uses during their studying" [105, p. 120]. Examples of study tactics are note-taking, self-testing, and viewing videos. Research on tactics has revealed that students' success is related to the awareness of using certain tactics and the breadth of tactics used [41]. Like habits, tactics are aspects of what students actually do; however, the choice and use of specific tactics are connected to the cognitive levels of engagement and control. Furthermore, tactics are often discipline specific and for computing include some unique tools, such as debugging and pair programming.

2.3 Educational Context and Design

The students' study behaviors happen in close relation to the educational context, which is defined as organized teaching and learning activities, the learning environment, and the curriculum [12]. The educational context involves physical, cultural, and social aspects and is inherently linked to cognitive and concrete aspects of study behaviors [3, 27]. Table 2.1 specifies a framework inspired by the micro-, meso-, and macro-level perspectives on educational structure in higher education in, for example, Lock et al. [57] and Dysthe and Engelsen [31]. The reason for introducing this framework is to communicate the essential elements of the design used in this thesis to educators and researchers from other educational contexts; the framework has been used in several papers.

Table 2.1: Summary of the design elements of higher computing education

Level of control	**Design Elements**	**Design Parameters**
Institution	Admission	Prerequisites, enrollment structure
Rector/pro-rectors,	Learning environment	Campus layout and infrastructure
central administration	Scheduling and timetables	Lecture and lab time slots
	Quality assurance system	Evaluation and feedback routines
Program	Program design	Number of semesters
Program leaders, dean		Weight of a course (number of credits)
		Enrollment practice
		Parallel vs. modular courses
Course	Course structure	Open or closed enrollment
Course teacher,		Number of students
department head	Learning activities	Pedagogical design
		Number of lectures
		Number of assignments and/or projects
		Individual or group-based activities
	Assessment	Type of assessment and exams

The institution level is the central or highest level, which varies in size and control. In higher education, disciplinary characteristics are mostly stable across countries and exert macro-level influences [107]. The program level refers to the place where students are enrolled. In some educational contexts, this might be a school of engineering or a major; however, in the case studied in this thesis, students are organized into a study program. At this meso-level, the top-down influences may be strong, and educators on the program level has little power to control it [31]. Lastly, the course level is perhaps a universal construct. Although the focus in this thesis is on the levels of educational design elements, a direct comparison can be made to instructional design and learning outcomes. For example, Biggs's outcomes-based teaching and learning framework states that learning outcomes exist at three levels: graduate, program, and course outcomes [13].

Each level of control has different design elements and parameters. The admission system, the physical learning environment, and the scheduling scheme are at the institution level, as they are controlled by the institution as a whole or by a government system, as is the case for admission in Norway. Each of the design elements has a number of parameters, that is, the different "variables" considered. For example, regarding the learning environment, the campus layout can be designed with group rooms, study halls, auditoriums, and the like. At the program level, the program design is the main design element, involving several design parameters, such as the number of semesters, the weight of a course, and the course structure. Lastly, the course level includes the elements of a course: its structure, learning activities, and assessment.

Higher education institutions are organized in many ways, and this framework aims to incorporate most designs, thereby highlighting the interconnected complexity [57]. When discussing students' study behaviors, it is essential to make a distinction between the levels of context [58]. As Lonka et al. [58] describe it, students' study processes act as mediators between their backgrounds and strategies and their habits and tactics. They further describe how the institution level relates to the general behavior, how the program level relates to the discipline-specific approach, and how the course-specific task relates to the situational approach.

2.3.1 Learning Environments

The term learning environment is, for the lack of a better description, academically iffy, and it is challenging to find a specific and coherent definition. Formally distinguishing the learning environment from just a classroom or a group of students learning is challenging. Within higher education, learning happens everywhere, at different times, at different levels of organization, and with a variety of people. Educational glossary [74] defines a learning environment as,

> *[T]he diverse physical locations, contexts, and cultures in which students learn. Since students may learn in a wide variety of settings, such as outside-of-school locations and outdoor environments, the term is often used as a more accurate or preferred alternative to classroom.*

Within learning sciences, the learning environment "is an artifact designed in a historical context, in response to cultural constraints and expectations, which is intended to bring about societally desirable learning outcomes" [68, p. 7]. The learning environment is also a factor in Biggs's work on SAL in the 1980s [12]. In his Presage, Process, and Product (3P) model of learning in higher education, he describes how "students undertake, or avoid, learning for a variety of reasons; those reasons determine how they go about their learning, and how they go about their learning will determine the quality of the outcome" [12, p.5]. An important part of the presage is the teaching context, which, in addition to the learning environment, includes the curriculum and the assessment and teaching methods. These factors have in common that the institution controls them, whereas the other aspect of presage, the student characteristics, exists prior to the learning and relates to the student. The final two parts of the model, process and product, are related to the students' approaches to learning and the learning outcome, respectively. As the 3P model suggests, learning environments are present within each course as well as at the program and institution level. These interactions constitute the *student-driven learning environment (SDLE)*, which is based on the individual students' perspective and describes how the students navigate, and interact with, the educational design constructs across courses within a program. It is a student-driven environment because it is the student who must navigate between organized activities and independent study, prioritizing and balancing the course load, managing his or her time, and using physical study spaces.

In the rest of this thesis, the SDLE will be the main focus, as characterized in Paper 4 and explored in Papers 5 and 7.

2.4 Learning, Studying, and Educational Design

So far in this chapter, I described and defined theories and concepts related to learning theories, study behaviors, and educational design. However, the connection between these three domains is not necessarily straightforward. Table 2.2 provides an overview of these connections, which will be further explained in the following paragraphs.

In the connection between learning and studying, learning can be viewed as the successful outcome of studying. A student can exhibit study behaviors that may lead to learning; however, this outcome is not guaranteed. Conversely, learning implies that a student has engaged in study behavior. The different learning theories put different emphasis on behavior [36]. Behaviorist theories focus on observable behaviors and how they are conditioned by the teaching activities and environment. Cognitive theories, on the other hand, focus on mental processes, while constructivism argues that behaviors are situationally determined. Lastly, connectivist theories view behaviors in a network of information and peers. A commonly adopted definition of learning that incorporates all these perspectives is that "learning is an enduring change in behavior, or in the capacity to behave in a given fashion, which results from practice or other forms of experience" [36, p.45]. As Ertmer and Newby [36] point out, not all learning theorists agree on this, and they argue that the important differences between theories lie more in the interpretation of how the learning theory is relevant for educational design and the design of instructions to facilitate learning.

Regarding the connection between educational design and learning, Ertmer and Newby [36] state that the role of learning theories in educational design is to shed light on the relationships among instructional components and to indicate how these components best suit specific learners. Furthermore, they emphasize that the crucial question for educators is not "Which is the best theory?" but "Which theory is the most effective in fostering mastery of specific tasks by specific learners?" [36, p. 61]. A similar perspective is presented by

 16

Table 2.2: The role of behavior and environment in learning theories

Role of	Behaviorism	Cognitive constructivism	Social constructivism	Connectivism
Behavior	Observable and measurable	Mental connections and processing	Situationally determined	Traversing the network
Environment	Conditioning	Facilitating	Interacting	Informal
Examples	*Group work*	*Feedback and reflection*	*Scaffolding and PBL*	*MOOCs*

Biggs and Tang [13], who argue against the notion that educators should aim for one learning theory to "rule them all" and rather focus on the contexts in which students learn. The context or environment plays different roles in different learning theories. In behaviorist theories, the environment conditions the change in behaviors, while in cognitive theories, the environment facilitates the learning processes. In constructivist theories, it is the interaction between the learner and the environment that is important. Lastly, connectivist theories emphasize the role of the informal environment [93].

Chapter 3

Contextualizing the Research

In the previous chapter, I explored the general education theories and definitions that underlie this thesis. In the following chapter, I will take a closer look at these theories and definitions within the context of computing education research. First, I will introduce the computing education research discipline and clarify some terminology for the global and Norwegian contexts. Then, I will summarize related work within the domains of study behavior and educational design in computing education.

3.1 Computing Education

The first generation of modern computers was developed in the 1940s, and the computing education field has been around nearly as long as the computing discipline. In Norway, the first computer was installed in 1954 [73], and the first computing education programs followed in the 1970s, along with the first university computing departments. In the United States and Europe, as in Norway, the computing departments and programs originated from the mathematics and engineering disciplines. Parallel to this, the computing education field developed and is now considered a separate discipline, with various subdisciplines [25, 46, 45]. Throughout the years, computing education research matured, debating methodological, conceptual, and epistemological approaches. Before the introduction of personal computers, computing education research focused on programming language design to (1) teach programming to novices and (2) learn through programming [45]. Over the years, this debate has taken different forms, but Guzdial and du Boulay [45] argue that the recent focus on computing for all and computational thinking represents a new era for the learning-through-programming discourse. Teaching programming to novices is also an ongoing discussion, with a large body of research into introductory programming courses [61]. Furthermore, computing education research in the 2020s is characterized by some growing pains, with increasing student numbers and a push to broaden participation [39]. In addition, changing demography, national curricula [39], and increasing demand for computing competency in the labor market are topics of recent interest [14].

In Norway, the term Information Technology (IT) is often used as an umbrella term for all things related to computing and computer technology. Regardless of accuracy or personal preference, it is important to have a common understanding of the terms. In this synopsis, the term computing means what in Norway is called IT: computing, informatics, and information and computer science. However, this terminology is not used consistently throughout the papers included in this thesis. I used computing and computer science somewhat interchangeably because I was unaware of the more recent shift to computing [1]. Computing education in Norway was the focus of Paper 1. For a more detailed description of the context of this research, I refer the reader to Paper 1.

3.1.1 Learning Theories in Computing Education

In 2019, an ITiCSE working group produced a quantitative analysis of how learning theories are adapted for the computing education research communities [97]. The group identified three main theory communities: social theories, experiential theories, and theories of mind. The most prevalent topics were student attitudes, errors and misconceptions, and learning styles. The working group also found that affordance theory, analytical behaviorism, and latent learning were among the least used theories. Regarding the role of learning theories and theories in general within computing education, several researchers have pointed out that the development of computing-specific theories, as well as the adaptation of general educational and pedagogical theories, is essential to the maturation of computing education research as a scientific discipline [63, 94].

Among behaviorism, constructivism, and connectivism in computing education, the constructivist perspective has dominated [97]. Cognitive constructivist theories have developed alongside computing education through, for example, Papert's work on the educational programming language Logo and later studies on constructionism [45, 84]. Regarding social constructivism, computing education research has emphasized the role of social, cultural, and historical contexts in learners becoming computer scientists [84]. Furthermore, arguments have been made that the central tenet of constructivism—that knowledge is constructed by the student—is closely aligned with the epistemology of computing [7]. As for the behaviorist perspective, both Robins et al. [84] and Szabo et al. [97] point out gaps in computing education research; however, one could argue that some traditional designs in higher education, such as lectures, fall within the behaviorist perspective on knowledge transfer. Robins et al. [84] suggest that learning analytics and the extraction of information about patterns of behaviors are emerging designs. Lastly, Szabo et al. [97] found connectivism to be common in computing education and closely linked to social constructivist theories such as scaffolding.

3.1.2 Study Behavior in Computing Education

Research on study behaviors in computing education has found that students exhibit many different behaviors when studying and learning computing concepts [67, 112, 8] and that differences between effective and ineffective students can often be explained by their behaviors [83]. A literature review found that "the most significant differences between effective and ineffective novices relate to strategies rather than knowledge" [83, p. 165].

Additionally, several studies confirm that students use many different strategies and habits when learning and understanding computing concepts [67, 112, 8].

Previous research on computing students' study behavior has identified that the classroom experience is not always the central aspect of a student's study day [91]. Instead of depending on lectures and teachers, students tend to rely more on online resources and their own independent work. The behaviors of higher-performing students have been found to be characterized by soliciting help, seeking out extra resources, taking extensive course notes [55], starting assignments early, working incrementally [38], attending lectures [19], keeping to an average workweek [112], and applying consistent behaviors throughout the semester [42]. In contrast, lower-performing students are more inclined to memorize material, seek answers from others without understanding them, not work on assignments after the deadline [55], use the internet, work with others, and rely on tutorials and model solutions [19].

Many studies are focused on introductory-level courses [19, 91, 55, 112]. One common underlying motivation for these studies is to understand how computing students study and predict their performance. Previous programming experience and lecture attendance have been found to improve exam performance, while internet usage, non-lecturer instructors, working with others, and the use of tutorials and model solutions did not [19]. In addition, more recent research has also focused on understanding behaviors, as opposed to only tracking and modeling. Prather et al. [82] examined the role of metacognition and self-regulation in programming education and found an increasing interest in cognitive control in computing education research.

The 2018 ITiCSE working group Luxton-Reilly et al. [61] found that gathering and analyzing behavior data to identify difficulties, design interventions, encourage change, and predict success and performance has become a focus area for research on introductory programming. Various perspectives and definitions, as well as many different research methods, seem to be in use. Many studies employ the data-driven approach [55, 42], meaning that behaviors are defined by the available data rather than by theoretical frameworks. As for the research methodology, questionnaires and interviews are widely used. More recent studies have used log-file and submission data as well [38, 42, 55].

For a more in-depth exploration of study behavior in computing education, I refer the reader to the systematic literature review in Paper 3. In this paper, we reviewed the research on study behaviors and discussed the definitions of study behavior, process, strategies, habits, and tactics presented in the previous chapter.

3.1.3 Educational Design in Computing Education

The design of computing education varies across the world. Different countries have different organizations, funding schemes, admission systems, program designs, and assessment regimes, and I do not aim to summarize them all. However, it is relevant to highlight some of these differences. Following the design elements framework presented in Table 2.1, I first present an example of different admission systems. Nordic countries have a government-run admission system that is mainly based on performance in upper secondary school. In Norway, students upload their transcripts, and the system takes care of the rest. No application statements or letters of recommendation are needed. In the United States and several European countries, there are different application and admission systems for each university or college. Some countries, such as India and the United States, also use specific tests for admission to different fields. At the program level, there are differences in the declaration of majors (United States), which often happens after admission, whereas in Norway, a student is enrolled in a specific program from day one. At the course level, many different rules and norms may govern the educational design. Assessment is perhaps the most influential one, with different countries using different regulations for how assessment is done and by whom. Another potential difference is the role of attendance, with some institutions having mandatory attendance or a points system in which attendance is one aspect. Norway has strict rules for assessment, but attendance does generally not count.

The educational system includes various interrelated institutions and mechanisms that shape and support computing education teaching and learning; each component is linked to and influences the other components [14]. Many studies have been carried out on different design elements and parameters within computing education. At the course level, the way that students study and several educational design parameters, such as mandatory and individual assignments, seem to be strongly connected [44]. For example, assessment practices have been found to drive individual learning even when peer learning is advocated by students [44]. Also, mandatory tutorials have been found to increase submissions and early starts on assignments [112]. At the program level, research has found that both the social and the academic learning environment benefits from students having access to informal learning spaces where they can collaborate with their peers [56]. Furthermore, the overall design of each year and the combination of courses, as well as teaching and learning activities, have been found to play an important role in students' performance [48]. The number of courses per semester, parallel vs. modular courses, the weight of courses, and the alignment between courses are some other aspects that have been investigated [104, 78]. Regarding the choice of an Integrated Development Environment (IDE) and technologies for use in computing courses, research has found that there is room for broadening students' abilities. For example, though the use of version control systems, web-based platforms, and professional IDEs [109].

Regarding the learning environment, research has found that students benefit from being part of a learning community [17] and that a focus on all aspects of the learning process and environment is valuable for students and educators [101]. The structure and teaching of a course define the learning environment, and educators should consider the implicit messages that these factors convey to students [104]. This point is also made in Szabo et al. [97], who visualize the interactions among individuals, groups, and artifacts in computing learning environments, emphasizing the differences between direct, indirect, and reflective interactions.

Theories and context summarized

This concludes the introductory chapters on theories, definitions, and the context of computing education. So far, I have explored the research on study behaviors and educational design, both in the general education literature and within the computing education context. Study behaviors can be hard to define. Various approaches to exploring student behaviors in computing education exist; however, the exploratory perspective has been somewhat neglected. In relation to educational design, the plethora of contexts and the lack of appropriate terminology for comparing design features between different countries provide a challenge. Regarding the related work on both study behaviors and educational design, the focus has been on course- and content-specific aspects. Learning theories were introduced to establish a theoretical connection between these concepts. In the next chapter, the focus is on methodology, but these theories, definitions, and findings from related work will be revisited in the discussion.

Chapter 4

Methodology

This chapter presents the research design, methods, and analysis underlying this thesis. First, I describe the mixed-method research approach and how I implemented it to answer the research questions. Next, I will go through each phase of this PhD project and describe the different studies.

4.1 Research Design

The overall design of this research was based on a mixed-method approach. Mixed-method research originated from evaluation research in the late 1980s when researchers started to combine qualitative and quantitative methods [24, 98]. At first, researchers utilized data collection and analysis tools from both domains, but later, they combined all phases of the research process and developed a methodological orientation. Qualitative and quantitative methods have different strengths and weaknesses, and researchers should exploit the strengths of both methods to understand social phenomena better [76]. Furthermore, a major strength of mixed-method research designs is that the different types of data can achieve multiple objectives, satisfying different stakeholders [47].

After several iterations, Creswell and Clark [24] landed on a definition of mixed-method research, incorporating many viewpoints with four core characteristics. In mixed-method research [p. 5] , the four key characteristics are that the researcher

- collects and analyzes both qualitative and quantitative data rigorously in response to research questions and hypotheses,
- integrates the two forms of data and their results,
- organizes these procedures into specific research designs that provide the logic and procedures for conducting the study, and
- frames these procedures within theory and philosophy.

Regarding philosophical foundations, choosing a mixed-method research design leaves the options open for several ideologies [24]. Mixed-method researchers must be aware of their philosophical assumptions, make them known, and be aware of and acknowledge other perspectives. The research in this thesis has been based on the pragmatic paradigm, and a deliberate choice was made to adopt a pluralistic orientation toward data collection and analysis [102]. Pragmatism, as a research paradigm, accepts multiple realities and orients itself toward solving practical problems [24]. An important axiom in the pragmatist paradigm is the notion that the research question should determine the method. Within education, the context might also be a factor that determines the method. In researching a course, some existing elements can be utilized in the research, such as exams and assignments, which should also be considered when designing research. In pragmatic research in education, one of the goals is to create vocabularies and descriptions that are useful in criticizing and developing educational practices [50].

Given the exploratory nature of the research questions, a mixed-method research design provided an open and adaptable approach to answering them. Eight studies were performed using qualitative and quantitative data collection and analysis methods. The studies were intricately connected and built on each other and the overall RQs. For several studies, different data sources were combined, as described in the next subsection. This PhD project has been guided by empirically based curiosity, constantly adapting future plans on the basis of current findings and fitting into the pragmatic worldview. Furthermore, some central principles were guiding the research process. First, the naturalistic setting was essential [85], meaning that the research was to be conducted in real-world scenarios. The goal was to align data collection to existing educational constructs, leaving out laboratory-based experiments. Secondly, the notion that research and practice should be connected was central. In practice, this principle meant that I aimed to be involved in the teaching, which ruled out many multi-institution approaches. Lastly, the application of these two principles meant that the research focused on one specific educational context instead of comparing several contexts.

4.1.1 Overview of the Research Process

The research process for this thesis work can be divided into three phases. In *Phase I*, the goal was to get to know the students and the educational context. This work resulted in two studies: a mapping of the computing education programs in Norway (Study 1) and an exploratory case study on the student experience (Study 2). Based on what I learned from these studies, *Phase II* focused on study behaviors and educational design. This work resulted in two studies: a systematic literature review of study behaviors in computing education (Study 3) and a case study on computing students' study behaviors throughout the first semester (Study 4). Lastly, *Phase III* included research on individual cases or ongoing initiatives conducted throughout the thesis period. This phase contained four studies (Studies 5-8) on different perspectives of study behaviors and educational design, which will be further described in Section 4.4. Figure 4.1 lists the studies, illustrates the phases, clarifies the connection to the research questions, and provides the data sources.

Although the term *phase* suggests a linear timeline, some of these studies were done in parallel. As illustrated in Figure 4.2, the data collection for Phase III was carried out at the same time as Phases I and II. However, the results of Phase III were analyzed in light of the results of Phases I and II.

Studies	PI	PII	PIII	RQ1	RQ2	RQ3	RQ4	Data Sources
Study 1: Mapping computing education programs in Norway	●				●			National databases; Study program information; National student survey
Study 2: Exploring the student perspective			●				●	Study Process Questionnaire; Student interviews over one year
Study 3: Reviewing research on study behavior in computing education		●				●		Research papers; Study program information
Study 4: Characterizing study behavior in computing			●				●	Student self reports
Study 5: Exploring the effect of the pandemic on computing students			●	●			●	Course information; Student interviews; Student survey QUAL+QUANT
Study 6: Experiment with voluntary assignments			●	●			●	Pre/post test; Student interviews; Exam results
Study 7: Designing and implementing the Study Day Initiative			●	●			●	Evaluation forms; Educational system information; Staff observation
Study 8: Exploring the role of teaching assistants			●	●			●	Self reflection essays

Data sources: *Database* *Documents* *Questionnaire* *System* *Interviews* *Test* *Observation*

Figure 4.1: Overview of studies and the connection to phases (Ps), research questions (RQs), and data sources

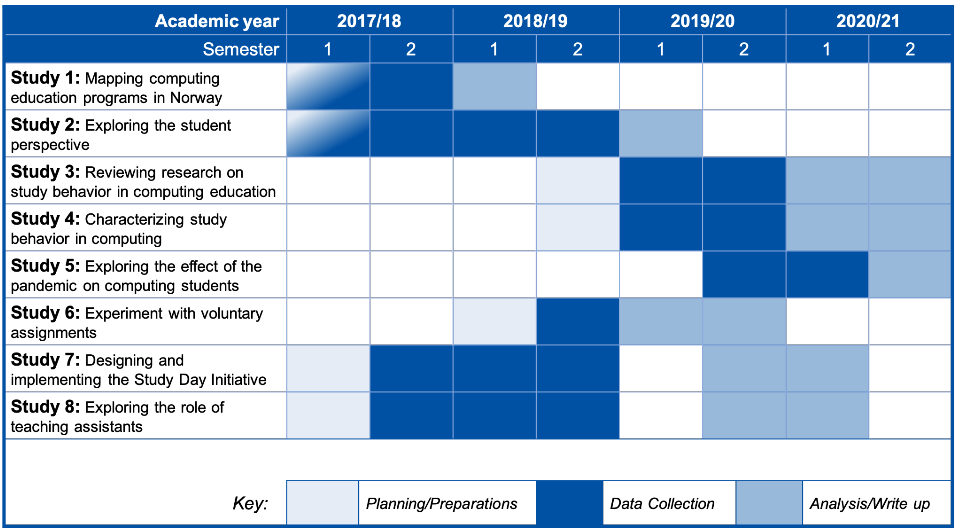


Figure 4.2: Timeline of planning, data collection, and analysis processes

In the following sections, I will describe these phases, focusing on the motivation and reasoning behind each decision and the link between studies. Furthermore, this description will include the data collection, participants, and method of analysis for the individual studies, expanding on the description in the published papers.

 26

4.2 Phase I: Getting to Know the Context and Students

Phase I adopted an exploratory approach to the first year of computing education in Norway. This phase was intended to start the inquiry into RQs 1 and 3, finding out what knowledge was needed to answer them. Phase I included two studies, which resulted in Papers 1 and 2. One study explored how study programs were designed, and one study explored the students and their experiences.

4.2.1 S1: Mapping Computing Education Programs in Norway

In this study, the goal was to obtain an overview of the undergraduate computing education programs in Norway. This included defining computing education, selecting the programs, and characterizing them according to some predefined variables.

A list of study programs was compiled using the The Norwegian Universities and Colleges Admission Service (NUCAS) and Norwegian Agency for Quality Assurance in Education (NOKUT) databases, selecting the computing study programs by name and database code. These databases provided information on Grade Point Average (GPA), admission requirements, student numbers, gender balance, and survey data from the National Student Survey "Studiebarometeret" [71]. Based on this list, we surveyed the first year of these study programs to categorize the various courses and their content. For each course, the name and number of credits were documented, as well as a category indicating what kind of course it was (computing, programming, mathematics, or other). The variables mapped and their source are described in Table 4.1. The result of this mapping provided a way to position the computing programs at NTNU that I focused on in the following research.

Design parameter	**Variables**	**Source**
Admission criteria	GPA and admission requirements	NUCAS
Student body	Number of students, gender balance, and overall satisfaction	NUCAS/NOKUT
First-year composition	Number and type of computing courses and programming language used	Manual documentation
Time engagement[1]	Organized teaching activities and self-study	NOKUT

Table 4.1: Design parameters and variables in Study 1

4.2.2 S2: Exploring the Student Perspective

In this study, the goal was to explore the student perspective in the first year. At this early stage in the research, I was curious about the pressure points in the first year, how students developed, and how their perspectives might change. Therefore, I initiated a longitudinal

[1]In Paper 1, we use the term *time commitment*, but following the updated terminology presented in Chapter 2, I will use *time engagement* in this synopsis.

interview study in which I followed the students throughout their first year. Based on the results from the interviews, a questionnaire study was designed. According to Creswell and Clark [24], this constitutes an exploratory sequential design, with a qualitative element (interviews) followed by a quantitative measure (questionnaire).

Longitudinal Interviews

We chose to recruit students from one specific program because (1) I had access to the program through teaching duties and (2) we wanted to understand the educational context and connect the students' statements to a specific context. The students were recruited at a voluntary weekly study day. All attendees were invited, ten people signed up, and six were chosen based on diversity and background. Table 4.2 summarizes the participants and their background.

Three in-depth interviews were conducted with each student over the first year. Each interview was based on a semi-structured approach, with different focus topics as listed in Table 4.3. The interviews varied in length, from 20 minutes to one hour. The interview guides for all three interviews are included in Appendix A.

Participant	Gender	Previous higher education	Experience with computing
P1	Female	Degree in a different discipline	None
P2	Male	Gap year before university	Upper secondary school
P3	Male	3 years in a different discipline	None
P4	Male	None	Upper secondary school
P5	Female	1 year in a different discipline	None
P6	Male	None	Upper secondary school

Table 4.2: Participants in longitudinal interviews

Interview	Timing	Topics
Interview 1	November 2017	Previous educational experiences, motivation and choice of study program, and first impressions of the social and academic environments.
Interview 2	April 2018	Evaluation of the first semester, how the new semester is going, focusing on study habits, strategies, and differences between the first and second semesters and pre-university.
Interview 3	November 2018	A retrospective evaluation of the whole first year. Presentation and discussions of the findings so far.

Table 4.3: Overview of timing and content of the longitudinal interviews

The interviews were transcribed and analyzed with an approach to thematic coding inspired by grounded theory [85, 22]. The data was analyzed by coding in three phases, as described by Corbin and Strauss [22]: open, axial, and selective coding. For a more detailed description of this process, I refer the reader to Paper 2.

Questionnaires

Toward the end of the interview study, a clear shift was identified in how the students viewed studying and their approaches to learning. Based on this, we decided to investigate this further with a quantitative questionnaire. The hypothesis was that computing students have a study process at the end of the first year that is different from the one they had initially. To test this, we used the Biggs revised two-factor Study Process Questionnaire (SPQ), which can indicate whether a student has a deep or a surface approach to learning [10]. The SPQ was translated into Norwegian, validated [114], and sent out to the students at the beginning and end of the first year during the academic year 2018/19.

The first iteration of the survey was sent out to first-year students in all NTNU computing study programs and had a response rate of 30%. For the second iteration, the study process questionnaire was part of a larger survey sent out to all students (in all years). The response rate for first-year students in the second iteration was 13%. For both iterations, the proportion of female respondents was around 30%, which slightly above the proportion in the class. The deep and surface scores were calculated following the revised two-factor method described in Biggs et al. [10], adapted for the Norwegian context as described by Zakariya et al. [114]. The questionnaire data was analyzed with Kernel density plots, and the hypothesis was tested with two-sample t-tests [43]. For all the quantitative analyses, I used the statistical software Stata MP [95].

4.3 Phase II: Focusing on Behavior and Educational Design

Based on the results of the exploratory studies in Phase I, Phase II focused on study behaviors and the connection to educational design. Examining the research on study behaviors in computing education revealed a fragmented field, with a plethora of definitions and research approaches in use. Therefore, I felt the need to systematically review the field to synthesize and structure the state of knowledge (Paper 3). Simultaneously, I wanted to develop a study to investigate the study behaviors of the students I was following at NTNU. This study continued to explore the close links between a specific educational context and study behaviors, resulting in the characteristics of the student-driven learning environment as described in Paper 4.

4.3.1 S3: Reviewing Study Behavior in Computing Education

Many theses start with a systematic literature review; however, in my case, I first had to identify what exactly needed to be reviewed. When I started searching for related work on study behaviors and computing education, I noticed the fragmented domain. Many terms and methods are in use, describing and analyzing the same behaviors with different words and tools, respectively. Initially, the idea was to find out what works best for computing education; however, it became clear from the start that that would be impossible. Therefore, the review focuses on definitions and aims to provide a useful overview of the domain. The procedure of this review follows Kitchenham et al. [49] and Bearman et al. [6], further described in Paper 3. The results of this review put some of the studies from Phase I into a slightly different light and have shaped the remaining work. However, it should be noted that the final version of the published paper was finished rather late in the research process, and in retrospect, I would have changed some terms and definitions as a result of the systematic literature review.

4.3.2 S4: Characterizing Study Behavior in Computing Education

The second study in Phase II ran in parallel to the systematic literature review. It aimed to explore the behaviors of computing students at NTNU in a broader sense than in Phase I. Study 4 was designed as a case study aimed at describing and explaining aspects of how first-year computing students study [21, 113]. The case can be viewed as the first year of a computing program, and the phenomenon of studying is researched holistically by following a group of students throughout their studies [4]. Initially, the goal of the study was to follow the students throughout their whole first year and to follow up with some interviews. However, because of the COVID-19 pandemic disrupting the whole educational context in March 2020, the data from the second semester is not comparable to the other data. Therefore, only data from the first semester was used in the intended way, while the data from the second semester was used for a study on the effects of the pandemic (Section 4.4.1).

Study 4 was conducted during the common introductory programming course, but the research perspective was the whole semester, including the other courses the students took. The students participating in this study were all enrolled in a computing study program: computing engineering, informatics, technology management, engineering and ICT, communication technology, or teaching and computing. There were 544 students, of whom 203 (37%) consented to participate in the research study. The male-to-female ratio in the course was approximately 70:30, and that in the participation group was 60:40. The students' age and ethnicity were homogeneous, with an average age of 20.

Along with weekly assignments in the introductory programming course, participants handed in a report in which they recorded when, where, and how they had worked on the assignments. In addition, a report was also handed in one week after the assignments, during the exam preparation week. From these weekly reports, the participants' study behaviors were modeled and tracked using the variables of the Student-Driven Learning Environment (SDLE) described in Table 4.4. The design of the reports and the wording of the questions were based on related work and the results from the literature review. The full self-report questionnaire can be found in Appendix B.

The analysis of the reports consisted of two parts: descriptive analysis and cluster analysis. The descriptive analysis was done by graphing the various study behavior dimensions by week. In addition, we wanted to examine the interconnections between the various elements of the dimensions. Using the statistical software Stata MP, we performed a cluster analysis on the different study behavior variables. K-median clustering with random initial group centers was run until a fitting model was found, exploring the number of clusters from 1 to 20 as described by Makles [62].

30

Dimension	Description	Variable
Organization	How students interact with organized learning activities and manage their independent study	*Time spent in/with:* Lectures Teaching assistant Alone Alone with peers Collaborating
Independent study	What tactics the students employ outside organized learning activities	*Use of:* Book Internet Note-taking Lecture examples Self-made examples Assignment Videos Memorizing
Planning and priorities	How students manage the course load	*Time spent compared with CS1:* Calculus Discrete Mathematics Scientific Philosophy
Time engagement	What days and what times of the day students study	What days of the week Time of the day
The study environment	Where the students study	*Use of:* Home Computing labs Study area Library Cafeteria Off campus

Table 4.4: Overview of the dimensions and variables of the SDLE

4.4 Phase III: Individual Studies

In Phases I and II, the research was driven by exploring, explaining, and developing theoretical frameworks, as well as by furthering our understanding in a broader sense. In addition, some studies were conducted in parallel to this core research. These studies serve as examples or quasi-experiments, which test or explore some specific elements of study behavior and educational design.

4.4.1 S5: Exploring the Effects of the COVID-19 Pandemic

When the pandemic hit Norway in March 2020, I had to change the plans for my research according to the rapid digital transformation in the educational landscape. At this point, I was planning to wrap up the data collection for Study 4 with interviews and a post-survey.

In the days after the campus-based education at NTNU was shut down, it became clear to me that it is was more important to follow up on the effects of the pandemic on the students. Following the pragmatic research paradigm, I took the opportunity to investigate such a game-changing event. Therefore, I abandoned the original plan for the spring semester of 2020 and initiated research on the shift from campus-based to online learning.

The change from campus-based to online education provided a natural, although unplanned, quasi-experiment. Building on the data and research from Study 4, the pandemic case study followed a group of 841 students in a CS2 course during the online transformation. These students spent the first eight weeks of the semester following a traditional campus-based course. During week nine, the course was changed into a completely online course. Since I could not set up a rigorous experimental study, with control groups and a random selection of participants, this study's research design can be viewed as a mixed-method, quasi-experimental investigation of a course [47, 106].

The original plan for Study 4 was to collect weekly reports on the assignments in the CS2 course, in the same way as in the CS1 course in the first semester. When the pandemic hit, I continued to collect those reports. In addition, I tracked the students' interaction with the digital teaching and learning activities and held interviews with seven students. Four of the interviews were held via written chat, while three were held via video chat. All interviews were directed by an interview guide based on findings from a preliminary survey among students and educators in the first weeks after the online transformation [60]. The interview guide can be found in Appendix A. The analysis of these three data sources was conducted following the model from Study 2, comparing the students' behaviors in the online learning environment with those in the campus-based learning environment.

4.4.2 S6: Experiment with Voluntary Assignments

One result from Study 2 was that assignments were important drivers for students. The mandatory weekly assignments used in most computing and STEM courses seemed to be the center point of the participants' study behaviors. Therefore, a study on the role of these mandatory assignments was developed along with a master's student who was interested in the influence of assignments as well.

A quasi-experimental research design was set up and implemented to investigate the effect of an intervention on a research population that was not randomly selected [21, 18]. The intervention was providing voluntary assignments, and the aim was to measure the effect on learning outcome and performance. Among over 700 students taking the course, 40 students volunteered to be part of the experiment, either as part of the experimental group with voluntary assignments or as part of a control group. The students in the experimental group were exempt from doing the mandatory assignments and were allowed to choose their learning resources. They were, however, required to attend biweekly meetings with a TA in which they had to describe what they had learned in the previous weeks and to explain how they reached the learning objectives for that week. These meetings, along with the pre- and posttest, served as the experimental group's qualifying activities for the exam. The students participating in the experiment were from various study programs within computer science. The male-to-female ratio of the participants was 50:50 for both groups.

To measure the effect of voluntary versus mandatory assignments, we looked at learning outcomes and exam performance. The learning outcome was measured as learning gain by subtracting the posttest score from the pretest score for each student. To deal with the quasi-experimental designs with nonrandomized participants and the low number of participants, an adjusted pretest score was also computed to compensate for the nonequivalent groups design in a Reliability Corrected Analysis of Covariance model [106]. The performance was measured by the grade of the final exam, ranging from 0 to 5, where 0 is an F and 5 is an A. Consequently, *posttest score* and *exam grade* were the dependent variables. The independent variable, *group*, differentiated between the experimental and the control group. Additionally, *adjusted pretest score* acted as a covariate in the analysis of learning outcome, and *grade in introductory programming (CS1)* as a covariate in the analysis of performance. To test the difference in learning outcome and performance, t-tests and ANCOVA models were run using posttest scores and exam grades as dependent variables, respectively. Interviews were also held with students in the experimental group after the posttest was performed to document their experiences without mandatory assignments.

4.4.3 S7: Designing and Implementing a Study Day Initiative

From the first week of my PhD project, I was involved in planning and implementing a Study Day Initiative for first-year computing students. The Study Day Initiative (SDI) is an all-day weekly study session in a reserved room, where first-year computing students are invited to work on assignments and courses, with TAs present to help them with anything they need. Throughout the years, evaluation surveys were collected; however, only later did it become clear how this intervention was relevant for this thesis. To explore educational design innovation in computing education, we performed a case study analysis on navigating the constraints of the educational system at a large university to improve the students' learning environment [113, 4].

The case study of the Study Day Initiative (SDI) had a holistic and evaluative design, spanning across three years [4]. The unit of analysis was the development of the SDI, and the case investigated was the population of students who participated. The case study was reflective in nature, looking back at various data points in an integrated way, providing opportunities for transforming teaching and learning practices. The data came from three sources. To describe the challenges and solutions, we relied on the educator's descriptions of the process. Questionnaire responses from the students were used to measure the student experience. In addition, we had a set of structured observations made by TAs during the last semester.

The analysis was divided into two parts: (1) the design and implementation of the SDI and (2) the evaluation of the initiative. Part 1 was analyzed using the design tensions paradigm developed by Tatar [99]. This paradigm helps understand design decisions in complex systems while emphasizing the balance of considerations in producing an entire system, especially the user group experience. Specifically, the paradigm highlights what *is* and what *ought to be* and illustrates the constraints in getting from one to the other. Part 2 looked at student feedback and observations to evaluate the student perspective.

4.4.4 S8: Exploring the Role of Teaching Assistants

From the first semester as a PhD student, I was responsible for the department's TA training. The task involved training TAs for specific computing courses, and I developed a curriculum and activities as described in [59]. Throughout the years I was teaching this course, I collected reflection essays and obtained consent to use them for research purposes. Together with some fellow TA trainers from two other institutions, these reflection essays were later used in a study on the challenges faced by TAs in computing across Europe. The question we asked the TAs was the following:

> *Describe an interesting situation or interaction you have experienced as an assistant. It can be something that you found challenging, an ethical dilemma, or just something that has been on your mind. Reflect on how you handled the situation. What did you do well? What would you have done differently? Is there something you would like to give feedback on, or do you have any questions?*

I collected essays from TAs at NTNU throughout the introductory TA training from 2017 to 2019. In addition, my co-authors collected the same data from their institutions during 2020. The data collection consists of 180 essays: 119 from NTNU, 32 from KTH (Royal Institute of Technology, Sweden), and 29 from MUNI (Masaryk University, Czech Republic).

The essays were analyzed using a thematic analysis [15], aiming to identify common challenges the TAs had written about. We followed the six steps outlined in Braun and Clarke [15], but with some adaptions for the specific data set at hand. The analysis was first carried out for each institution separately and then for all institutions together. For the essays from each institution, I and one co-author coded all essays independently and summarized the initial codes and identified themes. We then discussed and compared the findings of the independent analyses. This resulted in an agreement on the final themes and codes for each subset of data. Once the analysis of all three subsets was completed, we created a complete data set by merging the codes and themes, which was also carried out by both researchers independently, followed by a discussion resulting in the final themes. We then revisited the essays and previous codes to validate our findings and identify the origin of the themes.

4.5 Ethics and Considerations

Throughout the research, I was guided by ethical principles and considerations. First, ethical approval was granted from the Norwegian Centre for Research Data (NSD) for all the relevant studies. A relevant study was defined as any study that processed personal data. Personal data were, for example, student interviews, IP addresses obtained via questionnaires, or e-mails on contact forms. Table 4.5 provides an overview of the data and studies for which ethical approval was needed and the NSD reference[2]. In the next sections, I will further detail the considerations regarding the data collection.

[2]NSD applications and approval forms can be made available upon request.

Data source	Studies	Personal data	NSD reference
Interviews	S2	Voice recordings	56875
Questionnaires	S2	Indirect personal information and IP	391298
Reports	S4 + 5	Indirect personal information and email	841439
Experiment tests	S6	Name and email	281255

Table 4.5: Overview of personal data collected and NSD applications

4.5.1 Interview Strategy

For all the interviews, the participants were given information about the study, what it would entail to participate, and how to obtain or delete their data. They were also informed that participation was completely voluntary, that it would not affect their assessment in any way, and that they could withdraw at any time. This information was provided in written form prior to the interview and as an oral summary at the beginning of the interview. Participants signed paper-based consent forms and gave oral confirmation. Only after this introduction did I start recording. The recordings were kept on a hard drive (not connected to the internet) until the transcription was finished and then deleted. The transcriptions were anonymized by redacting names, places, and other identifiers.

During the interviews, I was mindful of making the experience pleasant for the students. Since I was planning on asking them personal questions, I spent some time building trust. I did this by sitting next to them, as opposed to directly in front. I also shared some of my experiences as a student where appropriate, often referred to as probes [85].

The participants in the longitudinal interviews (Study 2) were given a gift card as a token of appreciation. The value of the gift card was the equivalent of minimum wage for 2 hours of work.

4.5.2 Questionnaire and Written Material Strategy

For the quantitative studies, all the participants were given written information about the study, what it would entail to participate, and how to obtain or delete their data (i.e., the same procedure as for the interviews). This includes the questionnaires in Study 2, the reports in Studies 4 and 5, and the tests in Study 6. I always tried to make a personal appearance when informing or recruiting students, making myself visible and approachable for questions. For the online questionnaires, I used software provided by NTNU, making sure GDPR guidelines were followed. When processing the data, I avoided cloud-based storage and password protected all key documents. For other written materials, such as reflection essays in Study 8, I redacted names and references to courses before analysis.

For the questionnaires in Study 2 and the reports in Study 4, five randomly chosen participants received gift cards. The participants in the experiment on voluntary assignments (Study 6) were also given gift cards as a token of appreciation.

Studies 1 on 3 were based on documents, systems, or program data that were not traceable to any individual students and, therefore, did not need any ethical approval. Even though NSD approval was not needed, there were still ethical considerations. At each step of the research, I reflected on the potential ramifications of a data leak, the impact of such a leak, and the consequences for the people involved and took the appropriate precautions.

36

Chapter 5

Results

This chapter presents the results of the research in this thesis. The results have been published in eight papers in peer-reviewed journals or conference proceedings. In Part II, the papers are reprinted in full length, with permissions from the appropriate parties. In the following sections, I will summarize the studies' main results related to the overall research questions of this thesis. Hence, not all the results from the different studies will be included here.

5.1 S1: Computing Education Programs in Norway

The results from Study 1, on the first-year composition of computing programs, revealed several important aspects of Norwegian computing education and described the NTNU programs included in this research. In this summary, I will report on the main findings for the Norwegian context and highlight how the computing programs at NTNU are positioned. The full data set is available via Appendix C.

Regarding the admission requirements and GPA of incoming students, we found that most programs in Norway have mathematics requirements for entry. Students must have taken certain math courses in upper secondary school, which, for some programs, must be completed with a certain grade. Not a single computing program requires computing or IT courses from upper secondary school. The GPA of enrolled students is often used to assess the popularity of a program as well as the level of academic success for incoming students. Most computing programs in Norway had an admission threshold, implying that there is competition to get in. Furthermore, the NTNU programs rank high on this list, with five of the top ten programs. This indicates that NTNU's computing programs are popular with students and that incoming students scored high grades in upper secondary education.

Concerning first-year composition, we found that the number of programming, computing, and mathematics courses varied considerably in the first year. Furthermore, we found that Java was by far the most popular programming language, followed by Python and web-

 38

based languages such as HTML, CSS, and JavaScript. Regarding the student body, we found that most programs have 100-250 students in total, which makes most of the NTNU programs large in comparison. NTNU has the most programs and the highest total number of students, accounting for 31% of all computing students in undergraduate programs. On average, 17% of the students in computing education in Norway were female.

As regards students' time engagement, students in computing study programs studied between 20 and 52 hours per week. The average was 35 hours, which was in line with the national average for all students in Norway. Notably, for a large majority of study programs, students spent more time on self-study than in organized teaching activities.

	Program A	**Program B**
General description	3-year bachelor's program	5-year integrated master's program in engineering
First-year composition		
CS in first year	50% programming 12.5% computing 24% mathematics	37.5% programming 12.5% computing 37.5% mathematics
Programming language	Python, Web, Java	Python, Java
Student body		
Number of students	481 total, 140 per class	692 total, 140 per class
Percentage of female students	16%	20%
Admission criteria		
GPA	53.2	58.5
Requirements	Mathematics	Higher-level mathematics, physics
Time engagement		
Organized time	10.9 h/week	10.9 h/week
Self-study	21.2 h/week	24.2 h/week

Table 5.1: Summary of educational design parameters for two NTNU computing programs

In addition to an overview of Norwegian computing education, this study also provided a detailed description of the study programs that this PhD project focused on. As an example, I will highlight the two programs in which most of the participants were enrolled: (A) the bachelor's program in informatics and (B) the master's program in computer science engineering. All the students participating in Study 2 were from Program A, and in Studies 4, 5, and 6 most participants were from Program A or B. The Study Day Initiative was specially designed for these two programs. In addition, most TAs in Study 8 worked in introductory or first-year computing courses, in which at NTNU, students from both these programs were enrolled.

Table 5.1 provides an overview of these two programs as well as the variables included in this mapping study. Compared with the other computing programs in Norway, Program

A has more programming courses in the first year. Program B, on the other hand, has more mathematics courses. Both programs are high on the scale of GPA, as the maximum number of points in Norway is 64.0 for Program A and 66.0 for Program B. In Norway, the application and admission to higher education are centrally organized and involve complex point systems and quotas, as described in more detail in Paper 1. Programs A and B are the two largest computing programs in Norway in terms of student numbers, accounting for 14% of all computing students. Gender balance was in line with the average for computing programs in Norway, as was the time engagement.

Note that these numbers are from the academic year 2017/18, the same year as investigated in the qualitative part of Study 2. Since then, some of the variables have changed. Notably, the number of students in each class has increased to 155 for both programs, and the GPA for admission has increased to 56.2 for Program A and to 61.6 for Program B (as of 2020). Regarding time engagement, the organized time remained 10.9 hours per week; however, the self-study time has increased to 25.4 hours per week (Program A). For Program B, organized time has increased to 12.2 hours per week, and self-study time has decreased to 21.9 hours per week. Also, the proportion of female students has increased to 23% for Program A and to 29% for Program B.

5.2 S2: Exploring the Student Perspective

The results from Study 2 have two parts: the longitudinal interview results and the questionnaire results. The interviews revealed several important connections between study behaviors and educational design. Regarding independent study organization, the interviews indicated that deadlines, lectures, and assignments were important drivers. Further, the interviews showed that the way the participants studied was closely linked to the course design parameters. The findings from the interviews resulted in a model of student behavior and educational design, as shown in Figure 5.1. In this version, I have updated the model to include terminology and concepts from later findings.

The interviews also illuminated how the students developed over time. All the students described decreased motivation and, in their own words, “worse” study behaviors in the second semester. Taking shortcuts, struggling with balancing their social life, and the increased workload were all mentioned as negative aspects of the second semester. The quantitative questionnaire study confirmed that students changed their study behaviors, especially their approaches to learning throughout the first year. We tested the difference in deep and surface scores from the beginning of the first semester to the end of the second semester and found a significant difference in both surface and deep scores.

- H1: There is a significant difference in the surface scores between the fall and spring semesters. Confirmed: $t(241) = 2.06, p = 0.041, d = 1.25$
- H2: There is a significant difference in the deep scores between the fall and spring semesters. Confirmed: $t(241) = 9.16, p > 0.001, d = 5.68$

In other words, the students were more inclined to adopt surface approaches to learning in the second semester than in the first semester. Likewise, students were less inclined to adopt deep approaches to learning in the second semester than in the first semester. The data from this analysis is available via Appendix C.

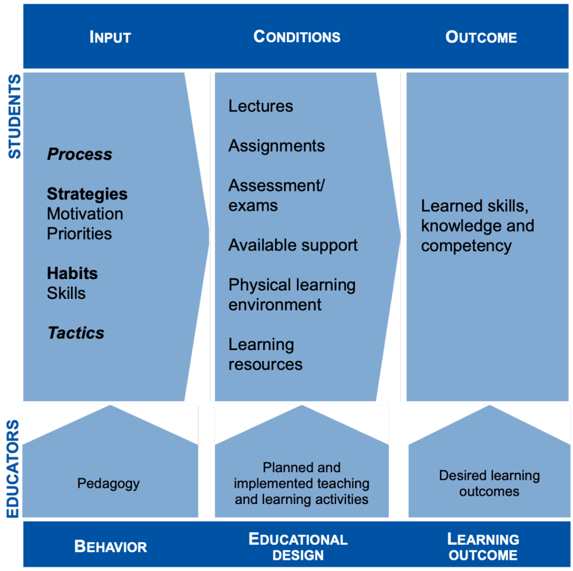


Figure 5.1: Model of study behaviors and educational design with revised terminology

5.3 S3: Reviewing Study Behavior in Computing Education

The systematic literature review in Study 3 comprised of 107 publications. The first main finding is that the same terms are used to describe substantially different study behaviors, and the lack of standard terminology makes it difficult to compare findings from different papers. Besides, 75% of the papers fail to define their terminology clearly or use self-defined terms where more established definitions are available. Data rather than theory seems to drive research on study behaviors, hampering the comparison of behaviors across courses, institutions, and nations. Based on these results, we developed a study behavior taxonomy for computing education, as illustrated in Figure 5.2

Furthermore, we found that study behaviors were used in many ways. Some papers used study behaviors to explain or predict academic performance, while other papers treated study behaviors as explanatory variables to investigate dropout and retention. A common element was that behaviors were often used to explain the quantity of learning. There were few examples of understanding study behaviors and how they affect the quality of learning. Several papers discussed “good” and “bad” behaviors without any further specification. Lastly, we found that study behaviors were investigated mostly at the undergraduate level, mainly in introductory programming. Therefore, there is a need to investigate more educational contexts, take a more holistic approach across courses and levels, and include more aspects and issues of study behaviors specific to computing education.

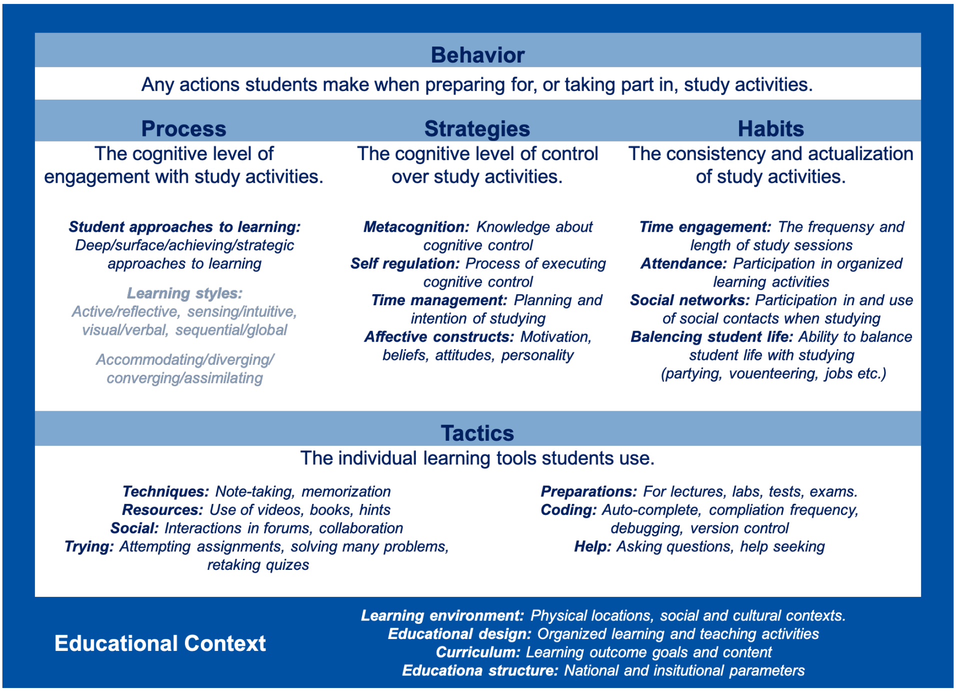


Figure 5.2: Taxonomy of study behaviors in computing education

5.4 S4: Characterizing Study Behavior in Computing Education

For Study 4, which followed a group of students throughout their first semester with weekly reports, the results were explored with descriptive graphs and exploratory cluster analysis. The anonymized raw data can be found via Appendix C. By comparing different aspects of student behavior to design parameters at both the program and the course level, we gain a deeper understanding of the Student-Driven Learning Environment (SDLE). This analysis revealed close relationships between the educational design and when, where, and how students study.

The SDLE has five dimensions: organization, independent study, planning and priorities, time engagement, and study environment. The graphs in Figure 5.4 illustrate the mean level of activity for each dimension, as described in Table 4.4. These graphs are an extension of the results presented in Paper 4 and cover the whole year. The first 11 weeks represent the assignments of the first semester (August-December), and the remaining weeks the second semester (January-June). Note that in the second semester, some assignments lasted over two weeks. The solid vertical line at 11.5 indicates the split between the semesters. Furthermore, the dotted red line between weeks 16 and 17 indicates the arrival of the COVID-19 pandemic, which will be further discussed in the results from Study 6. The cluster analysis is visualized in Figure 5.3, in which the cluster size reflects the frequency and the location of clusters illustrates their connections.

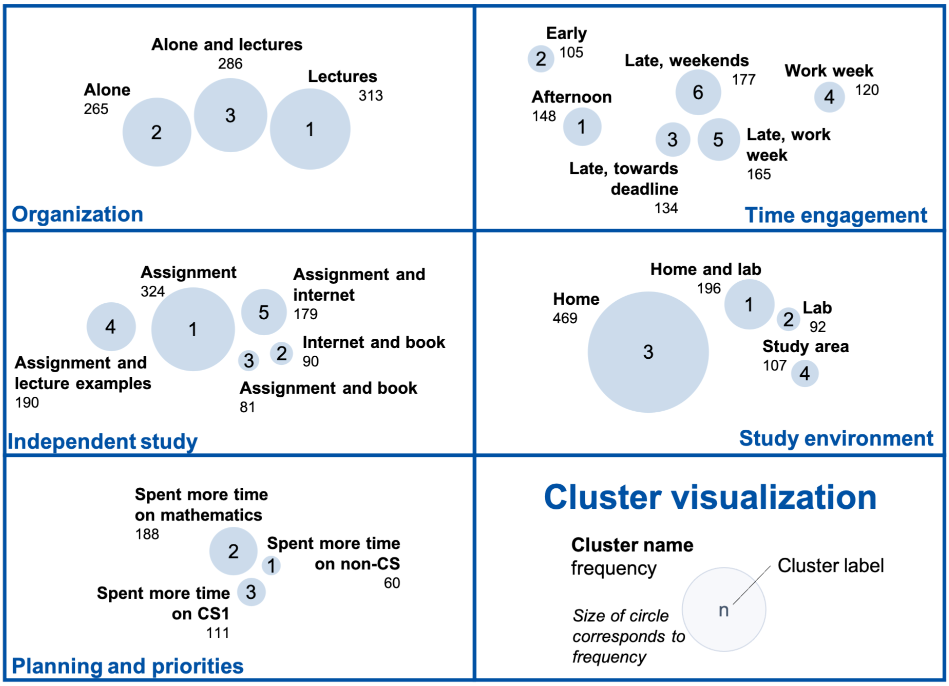


Figure 5.3: Visualization of cluster analysis

Based on the graphs and cluster analysis, three main characteristics were identified: the home-alone tendency, the executive action factor, and the organized activities component. The home-alone tendency represents the fact that studying alone and at home is the most prevalent aspect of the organization and study environment dimensions. The executive action factor represents the way students manage their time and handle their course load. We found that the students avoided mornings and weekends and prioritized mathematics over CS1. Lastly, the organized activities component represents how the students were driven by organized activities. In this case, assignments were found to be more important than anything else, including lectures. This is evident in all the dimensions in Figure 5.4 for weeks 4, 8, and 11. During these weeks, there were mock exams or exam preparations, no ordinary assignments, as reflected by a significant dip or peak in the graphs.

Although the pandemic changed the educational context of the second semester completely, there are some interesting developments as well. Regarding organization (5.4a), time spent alone increases in the second semester while participation in lectures goes down. As for independent study (5.4b), using the internet and focusing on assignments and rest are distinctly separated. Concerning priorities (5.4d), CS2 was consistently prioritized over other courses in the second semester, which is quite different from the prioritization in the first semester. Use of the study environments (5.4c) was much the same, with spending time at home increasing slightly compared with the other components, even before the pandemic. The time engagement dimension (5.4e-f) shows a large spike in the first weeks of the semester, followed by a steady decline.

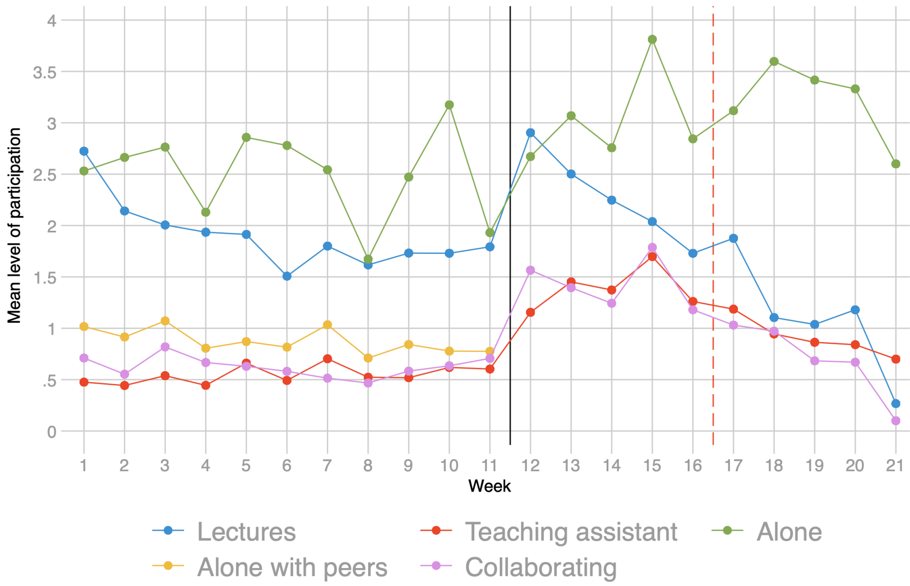
5.5 Phase III Results

The results from the Phase III studies focus on specific aspects of educational design and study behaviors within the SDLE characterized in Phases I and II.

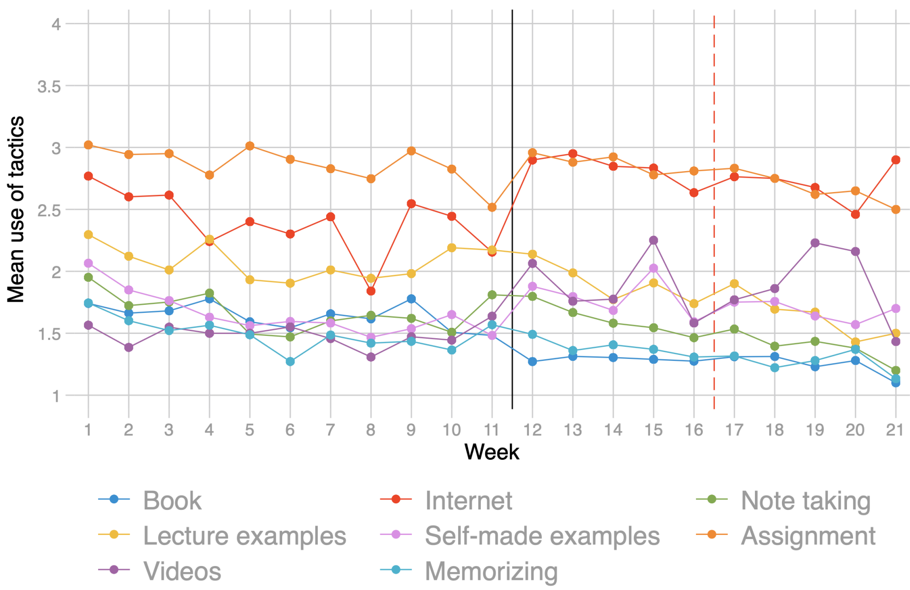
5.5.1 S5: Exploring the Effects of the COVID-19 Pandemic

We identified three main findings. First, informal learning spaces were essential to students, but at the same time, it was challenging to transfer effectively to the online environment. Normally, students would meet between lectures and when hanging out in the labs, but these informal spaces did not exist in the online environment. Many different digital platforms were used, but students had no place to informally meet their classmates, older students, or peers. Secondly, the scaffolding for effective study behavior provided by the on-campus environment's schedule and structure was valuable for students. Without set lecture times and organized labs, students seemed to struggle with their study routines. Lastly, the differences between struggling and successful students seemed to increase in the online environment. The following list provides a summary of the main findings:

- **Informal spaces:** Although most students joined the various platforms initiated, the number of active participants was low. The course included in the case study used two platforms for interactions and two for information flow. Half of these platforms were new to the students. Different courses seemed to use different platforms, and the total number of online tools and sites in use for each student was somewhat overwhelming. According to the interviews, students found it challenging to get a clear overview.
- **Structure and routines:** Students seemed to prefer asynchronously recorded lectures because they could regulate their viewing. Watching the videos at the time and speed they liked and the ability to go back and re-watch some sections were highlighted as positive aspects of video lectures. However, students also reported struggling with keeping up and structuring their day, two elements that the campus-based environment normally took care of.
- **Larger differences:** Students who did well before the pandemic also seemed to do well in the online environment. Conversely, students who did not perform as well in the campus-based environment seemed to struggle a lot more online. The structure provided by the schedules and informal spaces on campus apparently helped these lower-performing students study and master their courses. However, with most of that structure gone, they were in trouble. In summary, the online transformation seemed to create larger gaps. Furthermore, proportionally more female students than male students used formal help systems.

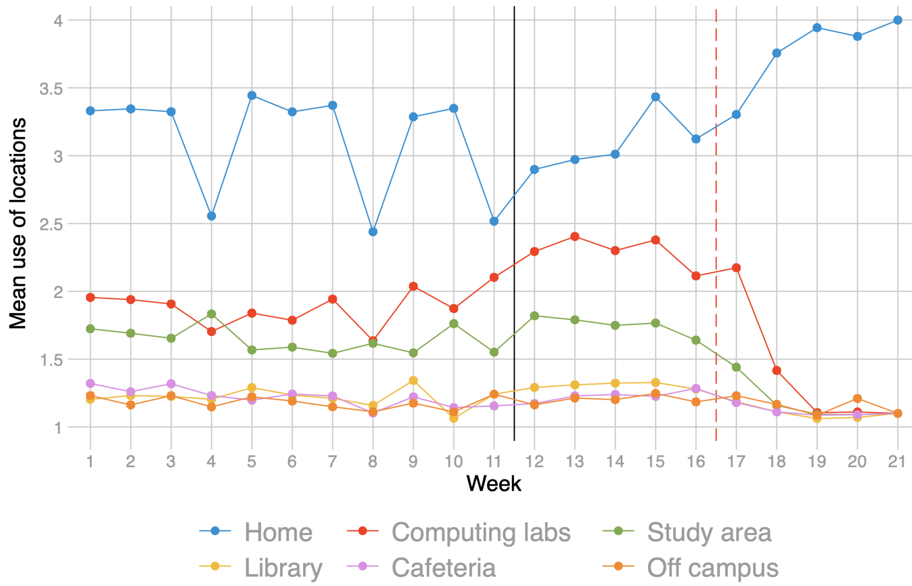


(a) Organization

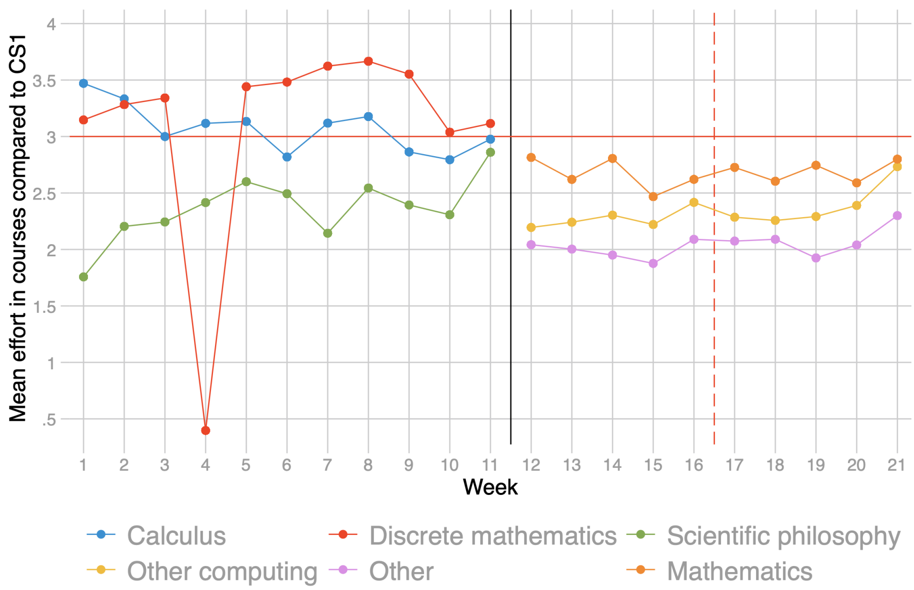


(b) Independent study

Figure 5.4: Graph of means for the dimensions organization and independent study

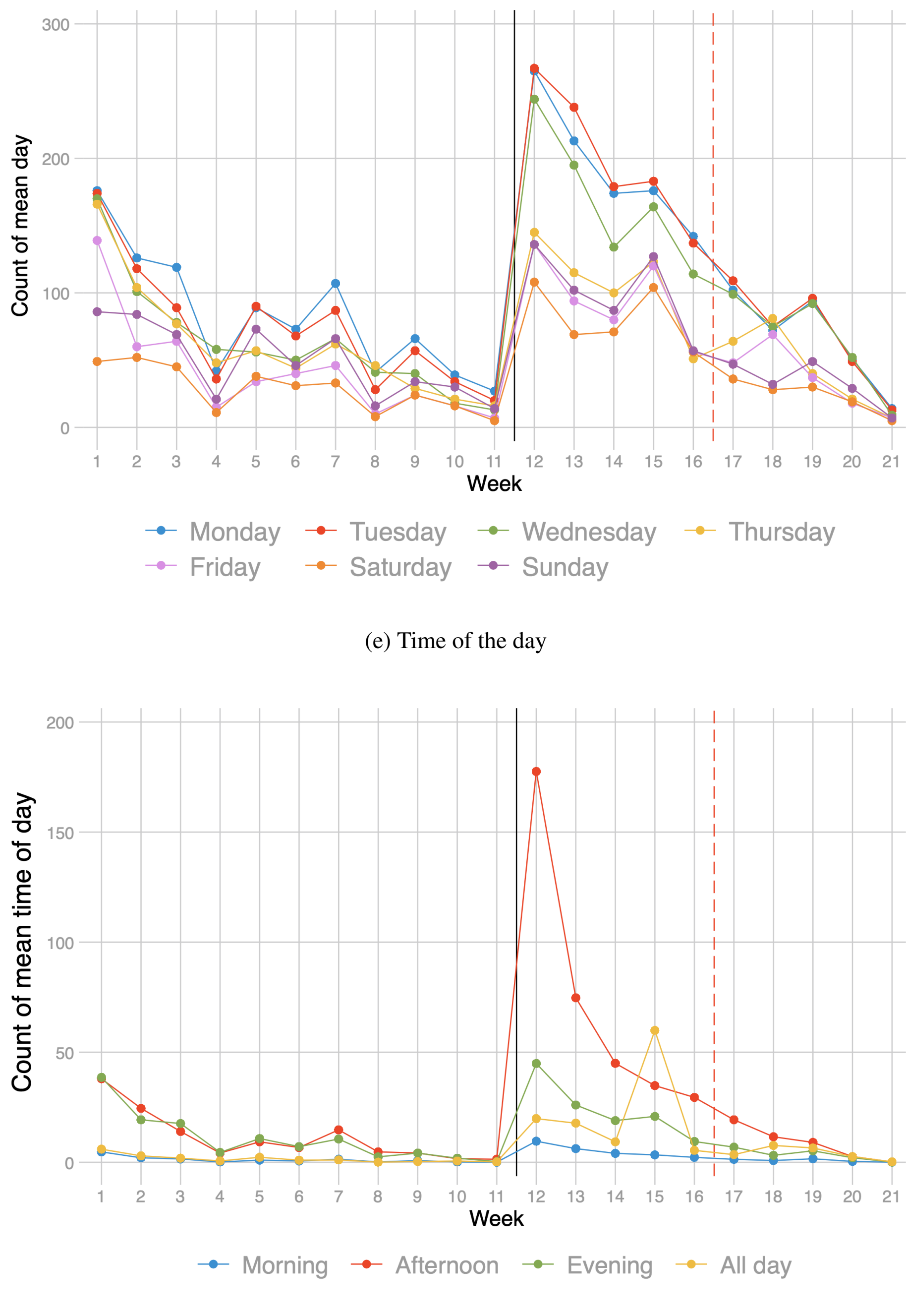


(c) Study environment



(d) Priorities

Figure 5.4: Graph of means for the dimensions study environment and priorities



(e) Time of the day

(f) Time of the week

Figure 5.4: Graph of means for the dimension time engagement

5.5.2 S6: Experiment with Voluntary Assignments

In summary, Study 6 found no statistically significant differences in learning outcome and performance between students with voluntary assignments and those with mandatory assignments. Students with voluntary assignments were given more autonomy to obtain the necessary course knowledge. The results further indicated that just checking whether a student has done the assignments is not necessarily more helpful than assisting them with the assignments or projects and letting them learn the way they prefer. Most students in the experimental group reported in interviews that they were happy to be exempt from mandatory assignments and felt this better suited their study behaviors. Their reported motivation was to do the assignments, even when this was not mandatory. Those who decided to do the assignments wanted to be sure they learned everything related to the exam and not to miss out on anything. This shows a considerable focus on the exam and the given grade. Additionally, many students mentioned that it was more fun to do the assignments when they were not compulsory so that they did not have to complete everything and could rather focus on the learning goals.

5.5.3 S7: Designing and Implementing the Study Day Initiative

The results from Study 7 on the Study Day Initiative (SDI) indicate that changing the educational design at the program level was (1) challenging within the constraints of a large institution but (2) very helpful to the students. The SDI provided a space where students could come together and work on assignments, learn, and get support. According to the evaluation questionnaires, students were more efficient and motivated and forged stronger academic and social bonds. The SDI is not the most revolutionary innovation; providing time, space, and support for students in the same place every week is similar to many traditional designs. However, operating the SDI within the systems and constraints of a large university was challenging.

5.5.4 S8: Exploring the Role of Teaching Assistants

Although Study 8 was focused on the TA perspective, the results of this study provide some insights into the students' interactions with the educational design parameter of TAs. The thematic analysis of the reflection essays from TAs identified five themes: becoming a professional TA, student-focused challenges, assessment, defining and using best practice, and threats to best practice. These themes shed light on the TA-student relations and describe the students from the TA perspective. Communication and language were identified as challenging areas for some TAs, which probably means that the students regarded these areas as challenging as well. Furthermore, the TAs highlighted several student-focused challenges, including students with individual challenges, different content knowledge, emotional presence, and different mindsets and abilities. All these bear on the TA-student relationship, which was especially challenging when dealing with cheating and plagiarism. Moreover, support and assessment posed a dilemma, as TAs needed to help the students and afterward had to assess their work. Together with the considerable time constraints described by many TAs, the TA-student relationship seemed to be constrained by both structural and human factors. In summary, these findings indicate that the TA-student re-

 48

lationship can be challenging for TAs, but previous studies indicate that TAs are essential to students.

Chapter 6

Discussion

This chapter addresses the research questions, elaborates on the contributions, discusses the implications, and describes the limitations of the research conducted in this doctoral work. The chapter is structured by first answering the research questions and then describing the corresponding contributions. Thereafter, I describe the implications for practice, policy, and research before, lastly, presenting an evaluation of the research process.

6.1 RQ1: Characteristics of Norwegian Computing Education

An important step in meeting the overall research aims of this thesis was to gain a better understanding of the educational design of computing programs. The first research question was concerned with identifying characteristics of educational design in computing education. Starting with the Norwegian context, the results presented in Paper 1 provide the first systematic attempt to map Norwegian computing education, based on the definitions from the computing education research discipline. Additionally, this work plays an important role in positioning the computing programs investigated in this thesis within the national and global contexts.

In Paper 1, we identified what Norwegian computing education programs have in common and in what aspects they differ. The main differences were found in the size of the programs in terms of the number of students admitted annually, the number of programming and computing courses, and the admission thresholds (GPA). The two programs studied in this research represent large programs with many students and a high admission threshold. The NTNU programs attracted students who were typically among the top of their class when graduating from upper secondary education. These students are likely to be better prepared for higher education studies in general; however, we do not know whether this is also true for computing. Previous research has indicated that prior experience with programming is an important predictor of academic success [19], but since prior programming experience was not a requirement for NTNU programs, we have no way of confirming this.

The large number of students sets boundaries to the educational design that smaller institutions than the NTNU do not necessarily have. When welcoming over 300 new students a year, educators are necessarily limited as to what they can do to aid the transition from school to university. Moreover, the push to increase the number of computing students graduating is not expected to end soon. Both in Norway [32] and in the United States [69], calls have been made to find ways to handle the increasing number of computing students.

Paper 1 also identified some common elements. All computing education programs in Norway have a significant mathematics component, with 75% of the programs having a math requirement for admission and 59% having at least one math course in the first year. Program B represents a math-heavy engineering program, and Program A represents a programming-heavy computer science program. We also found some trends in what programming languages were in use, with Java, Phyton, and web-based languages (JavaScript, CSS, and HTML) being the three most used language groups. Lastly, we can conclude that computing education is an attractive field of study for young Norwegians.

6.1.1 C1: Educational Design Elements for Computing

The first contribution is the framework of the educational design elements, which has been developed and utilized throughout the research of this thesis. The final version of the framework, with the identified variables for computing education, is illustrated in Figure 6.1. The specifics will, of course, vary from institution to institution; however, the elements and parameters should be present universally. Therefore, the variables outlined will always need to be decided on. We have shown that these variables have an impact on the study behaviors of computing students, so these variables should be considered with care in the educational design.

Based on the findings in Paper 1, the design elements framework was used as a tool to describe the educational context in Papers 4, 5, and 7. In Paper 8, which covered several institutions and countries, the framework provided the starting point for describing the different contexts of TA training and use. Although the findings show that the systematic framework provides a useful tool to describe computing education, further validation is needed by other studies in different contexts. Previous research in computing education has called for more detailed descriptions of educational contexts [61], and the design elements framework may be a good tool for that purpose.

In addition to using the framework as a tool to describe educational contexts, it can be used to help understand students and their study behaviors because it highlights the complexities of educational design [57]. As Lonka et al. [58] points out, the general behaviors at the program and institution levels can be differentiated from the discipline-specific behaviors at the course level. The findings from this thesis support the notion of differentiating behaviors according to the levels of education; however, I would argue that discipline-specific aspects of students' behavior are present at all levels. For instance, the widespread use of team projects [16] or pair programming [108] in computing education implies the need for a collaborative study environment, preferably as physical workspaces. Although the pandemic seems to have increased the digital literacy of students [100], the findings from Paper 6 suggest that group work and social learning were a big challenge for students in the online learning environment [60]. Nevertheless, the study environment is an institution-level design element, which supports the argument that discipline-specific behaviors exist at each level. As Biggs and Tang [13] pointed out the interaction between study behavior and educational design is not unlike the interaction between heredity and environment, and we need to consider both elements together if we want to expand the understanding of student learning.

Levels of control	Design Elements	Design parameters	Variables
Institution	Admission	Prerequisites	Mathematics requirement Computing requirement
	Learning environment	Campus layout and infrastructure	Spaces for individual study Spaces for collaborative study Informal meeting places
	Scheduling and timetables	Lecture and lab timeslots	Course or program perspective
	Quality assurance system	Evaluation and feedback routines	Course and program surveys and reports
Program	Program design	Semester structure	Number of semesters in a year Modular or parallel courses
		Course balance	Number of credits per course Number of programming, mathematics and programming courses
Course	Course structure	Enrollment practice	Open or closed enrollment Number of students
		Pedagogical design	Example: flipped, PBL, TBL etc.
		Educators	Number of lecturers/professors Number of teaching assistants
	Learning activities	Lectures	Type Frequency and length of lectures Mandatory attendance
		Assignments	Type Number of assignments Individual or group Mandatory or voluntary
		Labs/seminars	Type Frequency Mandatory attendance
	Assessment	Type of assessment and exams	Formative vs summative
	Computing aspects	Tool and technology	IDEs and version control Programming language

Figure 6.1: Educational design elements framework for computing

6.2 RQ2: Study Behaviors in Computing Education

The systematic literature review on study behaviors in Paper 3 is the main source for answering the second research question: What is the state of knowledge about study behaviors in computing education? Research on study behaviors in computing education is receiving increasing interest [61]; however, the lack of generally agreed terminology limits a systematic discourse. The theoretical perspective used and the educational contexts investigated need to be broadened. We found that many studies have been conducted on different aspects of study behaviors in specific courses, but the holistic perspective is often absent, regarding both study behaviors and study programs. Additionally, theories specific to computing education and the role of social learning remain poorly studied.

6.2.1 C2: Taxonomy of Study Behaviors

The second contribution of this thesis is the summary and conceptualization of study behaviors in computing education presented in Paper 3. The taxonomy of study behaviors is depicted in Figure 5.2. Importantly, the taxonomy does not attempt to model students' learning or behavior or to provide a unifying theory. Rather, it is intended as a tool to (1) help researchers and educators navigate study behavior terminology, (2) support the understanding of students, and (3) illustrate the connection to computing education design. In addition, the taxonomy can help separate the notions of learning and studying. The results of this thesis work do not allow drawing definite conclusions on what behaviors lead to learning. Therefore, the taxonomy illustrates all study behaviors and requires local interpretation and adaptation within different educational contexts.

Furthermore, the taxonomy contributes to the understanding of students by breaking down the different behaviors. The student population is constantly changing, as are their expectations, values, and demographics [39]. When educators welcome new classes of students into computing, they can use the taxonomy to help understand the students and their reactions to the teaching environment [13]. Like the learning theories, the cognitive aspects of study behaviors have evolved from passive to active perspectives [89]. The overlapping and interconnected nature of these behaviors is also valuable. The fact that use of different tactics is considered an effective approach to learning [41] highlights the relation between processes, strategies (cognitive), habits, and tactics (concrete).

Some perspectives are not evident in this taxonomy, even though they seem relevant at first glance. One example is the various motivation theories, which are often closely linked to behaviors and approaches. For example, the achievement goal theory [90] is similar to the SAL theory in the way it speaks about approaches. However, the taxonomy is intentionally abstract, keeping motivation as one aspect of study strategies, which has its own set of theories. Another example is student engagement, which can be viewed as a proxy for quality [52]. Student engagement can be seen as an evolving construct that captures a range of institutional practices and student behaviors related to student satisfaction and achievement, including time on task, social and academic integration, teaching practices, and how institutions affect student engagement [51].

Lastly, the taxonomy provides a valuable tool to identify the limitations of the different behavioral constructs. Cognitive engagement and control are aspects of behavior that can be observed only indirectly, through questionnaires and tests. On the other hand, habits and tactics have a directly observable element. This distinction is important, especially with the rise of behavioral informatics and learning analytics.

6.3 RQ3: Educational Design and Study Behaviors

The third research question asked how educational design impacts study behavior during the first year of higher computing education. To answer this, we need to combine the contributions from RQ1 and RQ2. The framework of computing education design elements systematizes the educational design elements for the investigated institution. Used together with the taxonomy of study behaviors in computing education, the framework helps understand and discuss the relations between educational design and study behaviors.

Study Behavior and Course Elements

Course-level educational design elements were found to have a significant impact on the students' study behavior. Paper 2 found that the students' study behaviors were to a large extent based on course elements. Students mentioned mainly lectures, assignment deadlines, and exams when describing their study process, strategies, and habits. Furthermore, Paper 2 identified a shift in focus during the first year. Students tended to be more *content-driven* in the first semester and more *task-oriented* in the second semester. This was also confirmed statistically with the Study Process Questionnaire. In Paper 4, these findings were further expanded. The identified *organized activities component* indicates a strong reliance on lectures and assignments in students' weekly study activities. In other words the organized activities form students' strategies, habits and tactics. Previous research has suggested that computing students rely less on lectures and teachers and more on independent study and peers [91]. Sheard et al. [91] found that most students studied alone most of the time. The researchers pointed to the individual nature of the assignments as a possible explanation, which supports the finding that organized activities are important drivers for study behavior [44].

The behaviors of high-performing students identified in previous research are to some extent related to course elements. Soliciting help and seeking out extra resources can be interpreted as course-, program-, and institution-level behaviors. Students at NTNU reported a strong relationship with older students as well as peers in their class. Nevertheless, taking extensive course notes [55], starting assignments early [38], and attending lectures [19] are only behaviors at the course level. Furthermore, the mandatory assignments prevalent in all the NTNU courses act as drivers, as found in [44] and [112]. The findings from Papers 2 and 4 confirm that the relationship between assignments and students' study behavior is strong, in particular habits and tactics. Furthermore, Papers 5 and 6 indicate that assignments help students structure their study behaviors and may improve their quality of learning.

On the other hand, the identified behavioral characteristics of lower-performing students are predominately at the course level. Memorizing code, obtaining answers from others without understanding them, not working on assignments after the deadline [55], using the internet, working with others, and relying on tutorials and model solutions [19] are all closely related to course assignments and activities. Again, one could argue that social aspects are also program related. Nevertheless, the indicators from previous research highlighted here do not include detailed context descriptions, and any inferences drawn from these indicators alone should consider the broader perspective.

Study Behaviors and Program Elements

The role of program-level design elements was found to be concerned mainly with the management of the course load. In the study programs investigated, all courses have the same credits; however, it is evident from this research that students will assign their own priorities. In Paper 4, the *executive action factor* identified how students balanced their time between courses. For this group of students, mathematics was prioritized in the first semester, and Paper 5 shows that programming was the main focus in the second semester. Several reasons might underlie this initial focus on mathematics and later shift. Students might fear failing mathematics more than failing programming, or students might learn how to trick the system by strategically retaking the programming course in the second year from older students. However, our findings do not allow drawing any definite conclusions yet. Therefore, further research into these potential explanations is needed, specifically looking at self-regulation and metacognition across courses on the program level. Another interesting avenue to pursue is to examine how the executive action factor develops as students mature and progress beyond the first year.

The findings from Paper 3 show that a lot is known about introductory programming courses; however, the program level is mostly absent in the research on study behaviors. Previous research has found that soliciting help [55], working incrementally [38], keeping to an average workweek [112], and applying consistent behaviors throughout the semester [42] are characteristics of higher-performing students in computing. These strategies, habits and tactics are aspects of the executive action factor, closely related to balancing course load and asking for support.

Study Behaviors and Institution Elements

The role of institution-level design elements is related to the learning environment. Paper 2 identified that the social and academic learning environments are important drivers of computing students' study behaviors. In both Paper 2 and Paper 5, we found indications that students would seek help and support from their friends and that the social and academic aspects of the learning environment were closely related. On the other hand, Paper 4 identified a strong *home-alone tendency*. At first glance, a divide might exist between students who study mostly on campus and students who study mostly at home, with the former group more engaging in social learning than the latter. We also found that the home-alone tendency increased in the second semester, even before the pandemic. These two findings, of students relying on their peers and collaboration and at the same time indicating a home-alone tendency, seem to contradict each other. One interpretation is that these two categories represent different students: social learners and individual learners. However, placing students into categories often turns out to be wrong and potentially harmful [70]. Another interpretation is that the students belong to both types of learners and that explanations must be sought in the educational and situational contexts. The educational design of the second-semester courses might be a factor here; however, it is difficult to see what effect this factor might have, as the levels of individual assignments and the general course organization are very similar. One could hypothesize that students are developing their own personal behaviors as a reaction to the increased difficulty and possibly unexpected results on first-semester exams. As students progress and become exposed to more team- and project-based designs, this tendency might decrease, or it might become too challenging for students to overcome.

As for the physical aspect of the learning environment, the findings from Paper 5 provide some insights. When the students were forced to shift from a campus-based to an online learning environment, we learned a lot about the importance of the campus. First, the informal learning spaces created between lectures and during lab hours were essential to these students. In addition, we found that seeking help was a challenge because students relied to a large extent on their peers and to some extent on older students [60]. Furthermore, the findings seem to suggest a drop in the students' experienced success in the online environment, indicating that the campus-based learning environment provides valuable scaffolding for students' study behaviors. These data must be interpreted with some caution because many other factors affected the students during the pandemic.

Another aspect of the institution level is the admission system. For computing education, previous experience with programming has been found to strongly affect performance and success in higher education [19]. The students in our studies all had high GPAs relative to the average in Norway; however, we do not have any indicators of previous computing and programming experience. Informal surveys during lectures suggest that about half of the students have taken computing courses or programming classes prior to university.

6.3.1 C3: The Ecosystem of Learning

The findings on the connection between course-, program-, and institution-level design elements contribute to an improved understanding of the holistic relationship between study behavior and educational design parameters in computing education. The term *holistic* refers to "dealing with or treating the whole of something or someone and not just a part" [26, p.1]. It is thus essential to look at the course, program, and institution elements together. The relationship between study behaviors and educational design is complex and dynamic. To illustrate these complex relations, I have borrowed a metaphor from biology: the ecosystem. In the ecosystem of learning for computing students shown in Figure 6.2, the population is the class of students within a course. The entire student group in one program makes up the community. The ecosystem, therefore, consists of the individual student, the class, the community, and the surrounding infrastructure, referred to as the biome in the original metaphor. The edusphere is obtained with the addition of the institution and national systems.

This metaphor is used to illustrate that students' behaviors are intricately linked to their surroundings. We found that computing students rely on lab hours, weekly assignments, peers, and informal collaboration spaces. Therefore, aiming to establish universal indicators of "good" and "bad" study behavior should not be a goal. Researchers should be reluctant to rely on prediction and analytics tools that do not consider the context when it comes to study behaviors and, in many cases, learning. As shown in this research, the ecosystem of learning is a complex system in which changing one parameter can have unintended effects. In conclusion, the ecosystem of learning contributes to the knowledge about understanding computing students.

Although the focus in this research has been on the first year, the ecosystem metaphor provides perspectives on higher education in general. Nevertheless, there are some important considerations specifically for first-year students. Transitioning to higher education studies can be challenging, and previous research has identified many issues and ways to address them [115]. A study by Zarb et al. [115] on computing students' transition to higher education found that the major concern was failing exams. Additionally, time and workload

 56

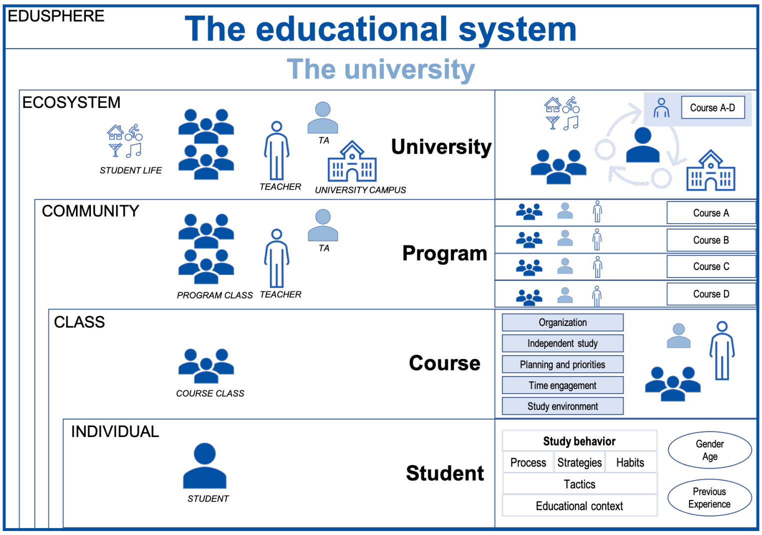


Figure 6.2: Ecosystem of learning: connections between educational design and study behaviors

management, preparedness, the availability of academic staff, and the prospects of securing good employment were also ranked high. These are aspects that can be addressed with a holistic perspective on the first year. Building a sense of belonging and promoting self-efficacy are difficult to do in one course alone, and the students' experiences during the entire first year lay important groundwork for later studies [103].

6.4 RQ4: Educational Design for Study Behaviors

Research question four is aimed at identifying how the findings presented so far can be used to improve the educational design of first-year computing programs. How can knowledge from RQs 1-3 be used to improve the educational design of first-year undergraduate computing programs? In this thesis, I present four examples from this research:

Course level: *Mandatory assignments or not (Paper 6).* In the Paper 6 study, we found that whether the assignments were mandatory or voluntary did not affect the learning outcome for this group of students. The results from Papers 2 and 4 show that the assignments provide important structure and constraints to the study behaviors. Although these results seem contradicting, it is important to point out that learning outcome was not measured in the other studies. On the other hand, the experiment with voluntary assignments was conducted with a nonrandom subset of the students, who mentioned in the interviews that doing the weekly assignments anyway was important for their progression in the course. Viewed together, it seems that not the final submission of an assignment but the studying for an assignment provided the scaffolding and produced the learning.

Course level: *The role of the TA (Paper 8).* In the research on the challenges of TAs, we also learned about the challenges of the students they interacted with. We found that students sometimes have problems with communication and language and that they struggle with personal issues not related to the course content. Furthermore, we found that the design of the TA-student space very often provides a time constraint. TAs reported that they had not enough time to explain concepts, to debug students' code, and to follow up the students throughout the semester. This time constraint indicates that the educational design parameter of the number of TAs per student was too high or that the number of lab hours was too low. More time and improved circumstances for the TA-student relation can facilitate for students to learn more about how to learn. By being exposed to more study strategies, habits and tactics, student can develop their study behaviors more comprehensively.

Program level: *The Study Day Initiative (Paper 7).* In the evaluation of the SDI, we found that creating a learning environment within a program can be hugely beneficial to the students; however, the constraints of the university system can limit the possibilities. In our case, it was room allocations and the scheduling of lectures and labs that made it impossible for us to change the educational design parameters via the formal system. This meant that we had to use informal channels to negotiate the changes needed, which is not a sustainable approach to educational innovation. Nevertheless, the behavioral scaffolding created by the SDI seemed to benefit the students both in the social and academic dimension.

Institution level: *The campus-based learning environment (Paper 5).* In the research on the effects of moving to an online learning environment during the COVID-19 pandemic, we learned a lot about how valuable the campus is for some students and what aspects of the campus-based learning environment we should aim to improve or understand better in the future. In particular the value of informal learning spaces found on campus, where students collaborate and socialize outside formal instruction. In other words, both the obvious and hidden aspects of the campus provides important scaffolding for students' study process, strategies, habits and tactics.

These four examples highlight how some aspects of the educational design elements framework relate to study behaviors; however, it does not provide a complete picture. For example, this research has not examined important design parameters such as team- and project-based designs, formal assessment, or the use of specific tools. More research into elements and parameters not mentioned here is needed to get a full picture.

6.4.1 C4: The Student-Driven Learning Environment

The fourth contribution of the thesis is the definition and distinction of the Student-Driven Learning Environment (SDLE) in computing education. The SDLE is a general pedagogical concept, but the results of Papers 2, 4, and 5 indicate that it can provide some important suggestions for computing education. In Paper 7, we confirmed that time, space, routine, and support contribute positively to the students' perception of the learning environment, which has been found to strongly affect learning positively [33, 96]. How students learn—in other words, their study behavior—will determine the quality of the outcome [13].

For the reasons that the SDLE is so important to students, we can turn to the learning theories presented in Chapter 2. Table 2.2 illustrates how the different learning theories view the learning environment. According to the behaviorist perspective, the SDLE is conditioned by the educational design parameters, while in the cognitive perspective the SDLE is facilitated by these parameters. The distinction lies in the students being passive or active learners, respectively. According to Bandura's reciprocal determinism, students' behaviors are a product of personal factors and social behaviors [3], with the understanding that learning can be done by observing. In computing and programming courses, pair programming is a classic example [19]. Also, the physical environment in which students are studying becomes important, which can explain why the SDLE was experienced so differently in the online environment.

From the perspective of Vygotsky's sociocultural theories, scaffolding and the zone of proximal development provide a different lens [110]. In social constructivism, the environment provides the space for interaction where knowledge can be constructed. The educational design parameters act as scaffolding for students' study behaviors within the SDLE. As in the cognitive constructivist perspectives, the student is an active participant who develops internally when working with others. Connectivism takes this a step further, emphasizing the importance of the informal learning environment, which was found to be essential for the students in Paper 5.

6.4.2 C5: The Room for Action

The fifth contribution of the thesis is related to the work on identifying the room for action within the educational design of a computing program. The room for action refers to the elements of the educational designs that can be adjusted and the potential impact on students. By using the parameters and variables outlined in the educational design elements framework and the model of study behaviors and educational design, educators and researchers can identify the room for action in other educational contexts.

Figure 5.1 illustrates the structure of constraints, educational design, and relation to study behaviors. The model also emphasizes the plans of educators (intended learning outcomes and planned learning activities) on the one hand and the results of the students (activities they undertake and actual learning outcomes) on the other hand. The cause-effect relationship between design elements and student behaviors will never be straightforward, which should be kept in mind when identifying the room for action and evaluating any actions taken. In Paper 7, we used the design tensions framework [99] to further analyze the room for action. In addition, the educational design is not carved in stone and then offered to the students. There is often some dialogue between educators and students throughout the semester, resulting in adjustments and new actions. In other words, the educational design has some degree of plasticity, albeit to a limited extent in courses with a teacher-driven delivery style or in large or complex courses in which the overhead of making changes is extensive.

The NTNU Case

As an example, I want to summarize the educational design of the NTNU and outline the room (or rather, lack of room) for action. First, we must consider the starting point and the current focus of Norway's and NTNU's computing education. The Excited Center for Excellent IT Education was established in 2017 and in many ways exemplifies the main focus and challenges of Norway's computing education. The Excited Center has five main projects, with the first one focused on increasing the knowledge of IT and computing of pre-university students. The second project is aimed at supporting new students to become successful students and, in the longer term, computing professionals. In the third project, learning through construction, the aim is to maintain and further develop students' interest and excitement by the creative design of artifacts. The fourth focus is on developing highly efficient cross-campus learning spaces, and the fifth project focuses on career readiness and aims to strengthen and expand the education-work connectivity. In addition, there is the divide between "computing for computing" and "computing for all," with the Excited Center belonging to the first category, but other centers exist that fall into the second category. Based on the main topics at international conferences and in the computing education online discourse, the Norwegian focus is somewhat different. For instance, a focus on diversity beyond gender, retaining computing students, and the use of various tools are not yet very present in Norway's computing education.

In addition, we must consider some of the national higher education events. In 2016, a large reform led to several institutions merging. NTNU merged with three colleges, all with computing programs and departments, increasing the complexity of the educational system and creating some cultural challenges. Many of my colleagues would agree that we just passed the "making-it-all-work phase" and can only now start with the actual innovation. In other words, our room for action has been heavily constrained by factors outside our control. Much time and many resources have been spent on committees for merging physical allocations, study programs, and learning outcome goals. Another example is an upcoming regulation change, requiring all graded assessments to have an external as well as an internal examiner. According to many educators, this rule will reverse many active learning initiatives. For the computing education community, this rule will affect our ability to run project-based courses and severely limit our options of formative assessment.

In summary, it is my opinion that in some aspects first-year Norwegian computing education is behind the international front line, especially regarding assessment and active learning initiatives. However, as I have outlined, there are reasons for this which largely acquits the computing education community from much of the "blame." Concurrently, Norway's computing education has large international potential in some areas. The governmental admission system and free education ensure that students have equal opportunities, and although Norway must work on closing the gender gap, it has a good starting point. Lastly, I want to stress that when discussing computing education it is important to be mindful that the notion of 'one size does not fit all' also applies to the educational context, not just the students.

 60

6.5 Implications

The research contributions presented above have potential implications for educators, practitioners, and policy makers in higher computing education. Based on these, I have outlined some general design guidelines aimed at supporting decision-making and identifying the room for action. Every educator or policy maker needs to interpret the frameworks and models identified in this research in his or her own context, and these guidelines may support that process.

1. **Outline the design levels, elements, parameters, and variables** for your educational context. Besides the formal design elements, such as assignments and campus layout, also consider the informal elements, such as the social dimensions across levels.
2. **Identify the room for action** by evaluating the costs and benefits of changing the variables. Consider the dimension of time (short term/long term), resources (initiate/maintain), and impact (students/educators). Question everything, even the most ingrained traditions and systems. Also, consider the indirect effects of design elements and variables, as well as exposing students to a variation across courses and classes.
3. **Implement and maintain the change**. Some innovations may need only an implementation phase; however, many innovations will need to be maintained. Consider ways to avoid changes being dependent on one educator alone by aiming to embed changes into educational systems and policies.
4. **Evaluate the effect and impact**, and consider aspects besides students reaching learning outcome goals and their performance on assessments. Other factors may include student satisfaction, engagement [52], dispositions [40], and behavior development, as well as educator contentment, enjoyment, and time commitment.

These guidelines are meant not just for practitioners starting a new course or working on major changes but also for mid-semester evaluation and dealing with problems. Additionally, these guidelines can be useful for policy makers and decision takers, as well as students, who might be frustrated by the lack of instant gratification. This list is evolving and should be further developed and nuanced in future research.

The research contributions also have potential implications for theory. For computing education research, arguments have been made for the need to use more educational theories and to develop theories specifically for computing education [97, 64]. Building on the insights from Contributions 2 and 4, I support the advancement of this effort. The taxonomy of study behaviors in computing education provides a theory-based approach to understanding computing students beyond their academic performance. The SDLE is a tool to extend the understanding of the connection between these behaviors and the educational design elements in computing education. Furthermore, this research demonstrated how learning theories can be used to understand study behavior and educational design, complementing the existing body of knowledge on performance. The exploration of general education theories and definitions within the computing education discipline contributes to contextualizing theory. Theories are approximate and cumulative, and this research is in many ways only a baby step on the way toward a mature research discipline.

6.6 Evaluating the Research

Evaluating research is an important procedure to ensure the validity of the data and results and their interpretation. Many classifications and definitions exist for describing the evaluation of mixed-method research; however, I have chosen to follow the outline of Creswell and Clark [24]. In mixed-method research designs, with both quantitative and qualitative elements, the evaluation of validity differs for the different elements and their integration [24]. In the following, I will consider the dimensions of research quality that relate to different data collection methods; however, I will not cover every study in detail (see individual papers). The goal of this section is to highlight the main threats to the project as a whole and how I worked to limit them.

6.6.1 Validity of Quantitative Research

In quantitative research, the researcher is concerned with validity and reliability. Validity is often further differentiated into construct, internal, and external validity as discussed in the following.

> **Construct validity** is concerned with whether the research measures the concept that it is intended to measure. Threats to construct validity include using inexact or confounding definitions, mono-operation bias, hypothesis guessing, and reducing the levels of measurements [21, 18]. The construct validity of the questionnaires used in Studies 2 and 4 and that of the experimental setup in Study 6 are the main points of concern for this evaluation. By using and validating the SPQ in Study 2, the threat of confounding measurements was reduced. In Study 4, I developed the questions for the weekly reports based on theoretical definitions and previous results from this research, which limited the threat of inexact measurements somewhat. In addition, I piloted the reports on a group of TAs and had several other researchers review the questions. For the quasi-experiment in Study 6, we used both test and exam results as measurements, limiting the risk of mono-operation bias.
>
> **Internal validity** is concerned with the degree of confidence in the results and making sure that they are not influenced by casual relationships or other variables [21]. Threats to internal validity include history, instrumentation, and subject effects. Since many of the studies were longitudinal, the threat of history effects was present, especially because of the real-world context of this research. However, my involvement in the courses and programs ensured that I was informed about most events regarding the educational design. Nevertheless, unforeseen events might happen. In this case, the global pandemic is perhaps the best example of such a disruption, which was managed by changing the scope of the ongoing study. Furthermore, much of the research relied on self-reported questionnaires or reports (Studies 2, 4, 5, and 8), which might lead to instrumentation and subject effects. Students could have been dishonest in their reporting or unmotivated to answer. Research based on surveys and questionnaires always involves these concerns; however, efforts were made to ensure that students felt comfortable that their responses were anonymous and would not affect their assessment in any way, hence facilitat-

ing reporting negative experiences or “bad” behaviors.

External validity is about to what extent results from a study can be applied to other situations [21]. Threats to external validity include selection bias and differences in contexts (real vs. experimental world). In our research, selection bias was mainly due to the participation being voluntary. Both for ethical and educational reasons, it was not possible to perform any true random sampling for questionnaires or experiments; however, we did randomize groups within the volunteers for the experiment in Study 6 and the analysis in Study 4. As all the research was conducted in naturalistic settings, the difference in context with a lab environment was not an issue. Nevertheless, computing programs and students at only one institution in Norway were investigated, which limits the generalizability to other institutions. However, efforts were made to generalize the terms and findings to provide adequate transferability of the design elements framework. In addition, the educational context has been described in detail, both at the local level and in relation to the national and international perspectives.

Reliability is concerned with the reproducibility of research and, consequently, the transparency of the research process [21, 24]. One way to ensure reliability is to document the methodology and analysis in a manner that can be reproduced. In the papers and this thesis, all the procedures and steps have been documented. Furthermore, the data from several studies have been made available via Appendix C so that other researchers might check for errors.

6.6.2 Validity of Qualitative Research

In qualitative research, the focus is on validity rather than reliability [24]. It is also common to talk about trustworthiness or authenticity in qualitative validity [22]. Qualitative validity can be evaluated by assessing whether the information obtained is accurate, credible, transferable, dependable, and confirmable. Established strategies exist for determining and enhancing validity, which I will describe in the following. Creswell and Clark [24] recommend that researchers employ at least three of these strategies in their research.

Member checking is discussing the main findings with the participants to find out whether they are an accurate representation of their experiences. This was done in Study 2 by discussing the findings from each interview in the next interview and via email correspondence after the full analysis was performed. In Study 1, my supervisors, as experienced computing educators, provided member checking of the coding of study programs. For Studies 5 and 7, I relied on informal discussions with students and TAs throughout the process.

Triangulation is a strategy in which the researcher draws data from several sources or individuals. Throughout the qualitative research, I recruited students from both investigated programs and, in some cases, other programs, providing triangulation of participants. In addition, the nature of mixed-method research provides triangulation between qualitative and quantitative data sources: for example, in Study 2 between questionnaires and interviews, in Study 6 between reports and interviews, and in Study 7 between questionnaires and observation. In Study 8, data sources from three different European countries could be triangulated.

Reporting disconfirming evidence, which is evidence contrary to the main findings, is also an important strategy to ensure validity. In several instances, I pointed out discrepancies and conflicting perspectives in the qualitative data: for example, the home-alone tendency and the importance of social learning found in Studies 4 and 2, respectively.

External examination of the data is another validity-enhancing strategy. I always involved my supervisors in the data analysis to provide oversight. Similarly, letting two researchers do the actual analysis of qualitative data is also a strategy, which was applied in Study 8. Furthermore, data from Studies 1 and 8 have been made available via Appendix C.

For the overall quality of the qualitative elements of this research, the trustworthiness of the participants is essential [22]. For this research, this relates to the interviews in Studies 2 and 5 and the reflection essays in Study 8. For the interviews, I employed several strategies to build trust with the participants. In their answers, they were not afraid to be critical of lecturers or honest about “bad” behavior, such as procrastinating and getting help from their friends, indicating that they were honest. Similarly, the reflection essays included reflections of a personal nature and descriptions of situations that were unfavorable for the TA.

6.6.3 Quality of Mixed-Method Research

The quality of the quantitative and qualitative elements of a mixed-method study can be addressed separately; however, beyond that, specific expectations and standards for mixed-method research exist. The standard evaluation criteria for mixed-method research are still being debated, and several lists and guidelines exist [75]. Following Creswell and Clark [24], as I have done throughout the research process, the criteria determining the quality of a mixed-method study are connected to the four key characteristics presented in Chapter 4. The first characteristic is the quality of the research questions. The RQs in this research were exploratory and inductive, and the findings do not attempt to confirm or state anything for certain. Furthermore, the scope of the RQs is limited to computing students in the first year, focusing on the Norwegian context as an example, which limits the inference transferability [75]. The quality of the quantitative and qualitative components has been covered in the two preceding sections.

The second characteristic is related to the quality of the integration of conclusions made on the basis of the findings from different types of data, often referred to as inference quality [75]. To assess the interpretive rigor, we examine the efficacy and credibility of the conclusions [98]. Inference quality is also concerned with the degree to which the findings and inferences of various strands of a mixed-method project are effectively integrated to yield a more advanced understanding of the phenomenon investigated. One example of this is the inference of results on assignments with the interviews in Study 2, the reports in Study 4, and the experiment in Study 6. Since the reports include only the weeks during the

 64

semester with organized activities, the interviews complement these by providing insight into the time before and after the assignments.

The third characteristic, design quality, is related to the organization of the procedures into specific research designs that provide the logic and procedures for conducting the study [98]. One aspect of design quality is the appropriateness of the data components; in our case, the quantitative and qualitative elements overlapped and complemented each other. In Study 2, the questionnaire answered some questions that were raised in the interviews. These questions were further explored with a different quantitative approach in Study 4.

The fourth, and last, characteristic is concerned with the role of theory [24]. Even though this was challenging at times, the data collection and analysis were always grounded in theoretical definitions and established frameworks. One example is Study 3, in which general education theories provided the search terms and grounds for analysis. Where needed, definitions and frameworks have been adapted and revised, as seen in Study 4 with the SDLE.

General Considerations

Several general considerations of this research need to be addressed, first of all the role of the researcher. As a "native" at NTNU, I have my own experiences as a student there. Therefore, the threat of confirmation bias is always present. Further, cultural bias and wording bias might also be an issue. I have been very aware of these threats throughout the research. To mitigate the effects of bias in general, I have reflected on my reaction during interviews and on how and when questions were asked. I have also challenged my preexisting assumptions and hypotheses. My mental model when doing interviews or designing questionnaires has been to reveal experiences and behaviors that were different from my own, so I have been driven by curiosity and openness, as opposed to aiming to confirm my own experiences. In addition, the pros of having insight into the educational design and culture were considered to outweigh the potential threats.

Furthermore, there are some limitations related to the participants. First, the number of participants was sometimes an issue. The response rate for the questionnaires was rather low; however, many students participated in the reports. For the interviews in Studies 2 and 5, the data provided saturation even though the number of students interviewed was not particularly large [85]. Secondly, as I relied on volunteers for both qualitative and quantitative studies, some student perspectives might not be present. For example, I have struggled with reaching students who generally do not participate much in learning activities or in general student life. I have gathered background information during interviews and surveys and have taken that into account during the analysis. Lastly, the time and structural constraints of a PhD program also play a role; however, such limitations did not significantly affect the trustworthiness of the results.

Finally, I must consider how the threats to validity, quality, and reliability impact the findings and conclusions of this research. One major concern is the representativeness of the participants. To counter this, I have described the participants thoroughly and taken their potential differences from the general population into account when interpreting the results. For example, the students were all above-average performers in upper secondary school, and they were generally very active in academic and social life, which must be considered when evaluating the findings. Another concern is the representativeness of the educational context, which has also been described in detail to provide the tools for generalizing. The presented findings should not be seen as a complete list of characteristics and impacts but must be viewed in light of the limitations of the studies.

Chapter 7

Conclusions and Future Work

The overall research objective of this thesis was understanding how knowledge about computing students' study behavior can help us design first-year undergraduate computing programs. This was investigated in a four-year research project with a mixed-method approach. The project had three phases: exploring the student perspective, narrowing down on study behaviors and educational designs, and conducting individual studies. Through questionnaires, interviews, and system and document analysis, this research has explored the different ways in which students engage with the educational design of computing programs at NTNU. Culminating in eight papers, the results show that educational design structures and scaffolds students' study behavior, both directly through, for example, assignments and schedules and indirectly through campus layout and informal learning spaces. Through the perspective of learning theories, the Student-Driven Learning Environment provides a deeper understanding of how students navigate through the educational design.

At this point, it is tempting to want a concrete answer on how first-year computing education should be designed to induce effective study behaviors; however, we do not have the empirical grounds to provide such answers. And there is reason to believe that we never will. The student population is changing in demography, previous experience, and expectations [39]. The structure, policy, and funding of higher education are being challenged, and perhaps even the overall goal of education [14]. As a result of the COVID-19 pandemic, we have experienced how education needed to transform rapidly, and we should be prepared for future disruptions. In conclusion, the educational system, with its surrounding policy and constantly changing student population, must be prepared to adapt and innovate. When creating and developing higher computing education, educators and other stakeholders tend to focus on the following types of questions: Are the students doing the right things? How well are they doing? How can I change my design so that students do better? The main argument of this thesis is to consider *why* the students do what they do and, consequently, to shift the focus slightly from the quantity to the quality of learning.

Based on the findings presented in this thesis, I have outlined some possible avenues to

 68

explore. Re-examining why we do things based on updated research and theories is an important first step. Every parameter and variable should be questioned, looking for room for action and innovation. In addition to increasing the understanding of computing students, I hope that this thesis may contribute to the knowledge about *how* to understand computing students.

The work presented in this thesis suggests several focus areas for future research. One perspective to explore further is understanding why students do what they do. Among other aspects, following up on the home-alone tendency, investigating social study behaviors, and exploring demographic and gender differences are interesting areas to examine further. Future plans could include conducting research on the effects of different assessment regimes, not only in terms of learning outcomes measured in tests but also in terms of effective and productive behaviors for increased computing competency, thereby focusing on the disposition component [40]. The effects of different tools, such as IDEs and version control, might also be explored.

I have more questions now than I had at the beginning of this project, and I hope that this research may provide others with a starting point for deepening our understanding of the relations between study behaviors and education design in computing education.

Bibliography

[1] ACM Computing Curricula Task Force, editor. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, Inc, January 2013. ISBN 978-1-4503-2309-3. doi: 10.1145/2534860.

[2] Francis Ashworth, Gabriel Brennan, Kathy Egan, Ron Hamilton, and Olalla Sáenz. Learning Theories and Higher Education. *Level3*, (2), 2004. doi: 10.21427/WGCF-ZP04.

[3] Albert Bandura. The self system in reciprocal determinism. *American Psychologist*, 33(4):344–358, 1978. ISSN 1935-990X(Electronic),0003-066X(Print). doi: 10.1037/0003-066X.33.4.344.

[4] Michael Bassey. *Case Study Research In Educational Settings*. Open University Press, Buckingham, May 1999. ISBN 978-0-335-19984-6.

[5] A. W. (Tony) Bates. *Teaching in a Digital Age: Guidelines for Designing Teaching and Learning*. February 2018. ISBN 978-0-9952692-1-7.

[6] Margaret Bearman, Calvin D. Smith, Angela Carbone, Susan Slade, Chi Baik, Marnie Hughes-Warrington, and David L. Neumann. Systematic review methodology in higher education. *Higher Education Research & Development*, 31(5): 625–640, October 2012. ISSN 0729-4360. doi: 10.1080/07294360.2012.702735.

[7] Mordechai Ben-Ari. Constructivism in Computer Science Education. *Journal of Computers in Mathematics and Science Teaching*, 20(1):45–73, 2001. ISSN 0731-9258.

[8] Anders Berglund and Mattias Wiggberg. Students learn CS in different ways: insights from an empirical study. In *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education*, ITICSE '06, pages 265–269, New York, NY, USA, June 2006. ACM. ISBN 978-1-59593-055-2. doi: 10.1145/1140124.1140195.

[9] John Biggs. What the student does: teaching for enhanced learning. *Higher Education Research & Development*, 31(1):39–55, February 2012. ISSN 0729-4360. doi: 10.1080/07294360.2012.642839.

[10] John Biggs, David Kember, and Doris Y. P. Leung. The revised two-factor Study Process Questionnaire: R-SPQ-2F. *British Journal of Educational Psychology*, 71 (1):133–149, 2001. ISSN 2044-8279. doi: 10.1348/000709901158433.

[11] John B. Biggs. *Student Approaches to Learning and Studying. Research Monograph*. Australian Council for Educational Research Ltd, 1987. ISBN 978-0-85563-416-2.

[12] John B. Biggs. Approaches to the Enhancement of Tertiary Teaching. *Higher Education Research & Development*, 8(1):7–25, January 1989. ISSN 0729-4360. doi: 10.1080/0729436890080102.

[13] John B. Biggs and Catherine So-kum Tang. *Teaching for quality learning at university: what the student does*. SRHE and Open University Press imprint. McGraw-Hill, Society for Research into Higher Education & Open University Press, Maidenhead, 4 edition, 2011. ISBN 978-0-335-24275-7.

[14] Paulo Blikstein and Sepi Hejazi Moghadam. Computing EducationLiterature Review and Voices from the Field. In Sally A. Fincher and Anthony V. Robins, editors, *The Cambridge Handbook of Computing Education Research*, Cambridge Handbooks in Psychology, pages 56–78. Cambridge University Press, 2019. doi: 10.1017/9781108654555.004.

[15] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, January 2006. ISSN 1478-0887. doi: 10.1191/1478088706qp063oa.

[16] Jürgen Börstler and Thomas B. Hilburn. Team Projects in Computing Education. *ACM Transactions on Computing Education*, 15(4):1–5, December 2015. ISSN 1946-6226. doi: 10.1145/2808079.

[17] Candido Cabo and Ashwin Satyanarayana. Promoting students' social interactions results in an improvement in performance, class attendance and retention in first year computing courses. In *2018 IEEE Frontiers in education conference (FIE)*, Frontiers in education conference, 2018. ISBN 978-1-5386-1174-6.

[18] Donald Thomas Campbell and Julian Cecil Stanley. *Experimental and quasi-experimental designs for research*. Houghton Mifflin Comp, Boston, 2. print; reprinted from "handbook of research on teaching" edition, 1967. ISBN 978-0-395-30787-8.

[19] Donald Chinn, Judy Sheard, Angela Carbone, and Mikko-Jussi Laakso. Study Habits of CS1 Students: What Do They Do Outside the Classroom? In *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103*, ACE '10, pages 53–62, Darlinghurst, Australia, Australia, 2010. Australian Computer Society, Inc. ISBN 978-1-920682-84-2.

[20] Felly Chiteng Kot. Gap between actual and expected time allocation to academic activities and its impact on undergraduate academic performance. *Tertiary Education and Management*, 25(1):65–82, March 2019. ISSN 1573-1936. doi: 10.1007/s11233-018-09015-8.

[21] Louis Cohen, Lawrence Manion, and Keith Morrison. *Research Methods in Education*. Routledge, London, UK, 5 edition, 2002.

[22] Juliet Corbin and Anselm Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, Inc, Los Angeles, 4 edition, December 2014. ISBN 978-1-4129-9746-1.

[23] Marcus Credé and Nathan R. Kuncel. Study Habits, Skills, and Attitudes: The Third Pillar Supporting Collegiate Academic Performance. *Perspectives on Psychological Science*, 3(6):425–453, November 2008. ISSN 1745-6916. doi: 10.1111/j.1745-6924.2008.00089.x.

[24] John W. Creswell and Vicki L. Plano Clark. *Designing and Conducting Mixed Methods Research*. SAGE, 3 edition, 2018. ISBN 978-1-4129-7517-9.

[25] Peter J. Denning. Is Computer Science Science? *Commun. ACM*, 48(4):27–31, April 2005. ISSN 0001-0782. doi: 10.1145/1053291.1053309.

[26] Cambridge Dictionary. Holistic. In *Dictionary.Cambridge.org*. April 2021.

[27] Daniel L. Dinsmore. Toward a Dynamic, Multidimensional Research Framework for Strategic Processing. *Educational Psychology Review*, 29(2):235–268, June 2017. ISSN 1573-336X. doi: 10.1007/s10648-017-9407-5.

[28] Daniel L. Dinsmore, Patricia A. Alexander, and Sandra M. Loughlin. Focusing the Conceptual Lens on Metacognition, Self-regulation, and Self-regulated Learning. *Educational Psychology Review*, 20(4):391–409, December 2008. ISSN 1573-336X. doi: 10.1007/s10648-008-9083-6.

[29] Stephen Downes. What Connectivism Is. 2007.

[30] Thomas M. Duffy and Donald J. Cunningham. Constructivism: Implications for the Design and Delivery of Instruction. 1996.

[31] Olga Dysthe and Knut Steinar Engelsen. Portfolio practices in higher education in Norway in an international perspective: macro-, meso- and micro-level influences. *Assessment & Evaluation in Higher Education*, 36(1):63–79, January 2011. ISSN 0260-2938. doi: 10.1080/02602930903197891.

[32] Fernanda Winger Eggen, Jonas Måøy, Rolf Røtnes, Marthe Norberg-Schultz, and Jørgen Ingerød Steen. Norges behov for IKT-kompetanse i dag og framover. Technical report, Samfunnsøkonomisk analyse AS, Oslo, Norway, 2021.

[33] N. J. Entwistle. Approaches to Learning and Perceptions of the Learning Environment: Introduction to the Special Issue. *Higher Education*, 22(3):201–4, 1991. ISSN 0018-1560.

[34] Noel Entwistle and Paul Ramsden. *Understanding Student Learning (Routledge Revivals)*. Routledge, August 2015. ISBN 978-1-317-51358-2. doi: 10.4324/9781315718637.

[35] Noel Entwistle, Velda McCune, and Hillary Tait. The Approaches and Study Skills Inventory for Students (ASSIST). Technical report, Centre for Research on Learning and Instruction, University of Edinburgh., Edinburgh, Scotland, 1997.

[36] Peggy A. Ertmer and Timothy J. Newby. Behaviorism, Cognitivism, Constructivism: Comparing Critical Features From an Instructional Design Perspective. *Performance Improvement Quarterly*, 26(2):43–71, 2013. ISSN 1937-8327. doi: 10.1002/piq.21143.

[37] Katrina Falkner and Judy Sheard. Pedagogic Approaches. In Sally A. Fincher and Anthony V. Robins, editors, *The Cambridge Handbook of Computing Education Research*, Cambridge Handbooks in Psychology, pages 445–480. Cambridge University Press, 2019. doi: 10.1017/9781108654555.016.

[38] James B. Fenwick, Jr., Cindy Norris, Frank E. Barry, Josh Rountree, Cole J. Spicer, and Scott D. Cheek. Another Look at the Behaviors of Novice Programmers. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09, pages 296–300, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-183-5. doi: 10.1145/1508865.1508973.

[39] Sally A. Fincher, Josh Tenenberg, Brian Dorn, Christopher Hundhausen, Robert McCartney, and Laurie Murphy. Computing Education Research Today. In Sally A. Fincher and Anthony V. Robins, editors, *The Cambridge Handbook of Computing Education Research*, Cambridge Handbooks in Psychology, pages 40–55. Cambridge University Press, 2019. doi: 10.1017/9781108654555.003.

[40] Stephen Frezza, Mats Daniels, Arnold Pears, Åsa Cajander, Viggo Kann, Amanpreet Kapoor, Roger McDermott, Anne-Kathrin Peters, Mihaela Sabin, and Charles Wallace. Modelling competencies for computing education beyond 2020: a research based approach to defining competencies in the computing disciplines. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE 2018 Companion, pages 148–174, New York, NY, USA, July 2018. ACM. ISBN 978-1-4503-6223-8. doi: 10.1145/3293881.3295782.

[41] Maribeth Gettinger and Jill K Seibert. Contributions of study skills to academic competence. *School Psychology Review*, 31(3):350–365, 2002.

[42] Seth Copen Goldstein, Hongyi Zhang, Majd Sakr, Haokang An, and Cameron Dashti. Understanding How Work Habits Influence Student Performance. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer*

Science Education, ITiCSE '19, pages 154–160, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6301-3. doi: 10.1145/3304221.3319757.

[43] Frederick J. Gravetter and Larry B. Wallnau. *Statistics for the Behavioral Sciences - Standalone Book*. Cengage Learning, Australia ; United States, 9 edition, 2013. ISBN 978-1-111-83955-0.

[44] Sheryl Guloy, Farimah Salimi, Diana Cukierman, and Donna McGee Thompson. Insights on supporting learning during computing science and engineering students' transition to university: a design-oriented, mixed methods exploration of instructor and student perspectives. *Higher Education*, 73(3, SI):479–497, March 2017. ISSN 0018-1560. doi: {10.1007/s10734-016-0097-6}.

[45] Mark Guzdial and Benedict du Boulay. The History of Computing Education Research. In Sally A. Fincher and Anthony V. Robins, editors, *The Cambridge Handbook of Computing Education Research*, Cambridge Handbooks in Psychology, pages 11–39. Cambridge University Press, 2019. doi: 10.1017/9781108654555.002.

[46] Mark Guzdial and Briana Morrison. Growing computer science education into a STEM education discipline. *Communications of the ACM*, 59(11):31–33, October 2016. ISSN 0001-0782. doi: 10.1145/3000612.

[47] Burke Johnson and Larry Christensen. *Educational Research: Quantitative, Qualitative, and Mixed Approaches*. SAGE, November 2010. ISBN 978-1-4129-7828-6.

[48] Päivi Kinnunen and Lauri Malmi. CS minors in a CS1 course. In *Proceedings of the fourth international workshop on computing education research*, ICER '08, pages 79–90, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-216-0. doi: 10.1145/1404520.1404529.

[49] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, 51(1):7–15, January 2009. ISSN 0950-5849. doi: 10.1016/j.infsof.2008.09.009.

[50] Osmo Kivinen and Pekka Ristela¨. From Constructivism to a Pragmatist Conception of Learning. *Oxford Review of Education*, 29(3):363–375, September 2003. ISSN 0305-4985, 1465-3915. doi: 10.1080/03054980307442.

[51] George D. Kuh. Assessing What Really Matters to Student Learning Inside The National Survey of Student Engagement. *Change: The Magazine of Higher Learning*, 33(3):10–17, May 2001. ISSN 0009-1383. doi: 10.1080/00091380109601795.

[52] George D. Kuh. The national survey of student engagement: Conceptual and empirical foundations. *New Directions for Institutional Research*, 2009(141):5–20, 2009. ISSN 1536-075X. doi: 10.1002/ir.283.

[53] Archana Kumari and S Chamundeswari. Achievement motivation, study habits and academic achievement of students at the secondary level. *International Journal of Emerging Research in Management and Technology*, 4(10):7–13, 2015.

[54] Jean Lave and Etienne Wenger. *Situated learning: Legitimate peripheral participation*. Situated learning: Legitimate peripheral participation. Cambridge University Press, New York, NY, US, 1991. ISBN 978-0-521-41308-4. doi: 10.1017/CBO9780511815355.

[55] Soohyun Nam Liao, Sander Valstar, Kevin Thai, Christine Alvarado, Daniel Zingaro, William G. Griswold, and Leo Porter. Behaviors of higher and lower performing students in CS1. In *Proceedings of the 2019 ACM conference on innovation and technology in computer science education*, ITiCSE '19, pages 196–202, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6895-7. doi: 10.1145/3304221.3319740.

[56] Fang-Ling Lin and Chyi-In Wu. Exploring the network dynamics in a flipped classroom. In *Proceedings of the 2018 IEEE/ACM international conference on advances in social networks analysis and mining*, ASONAM '18, pages 855–860. IEEE Press, 2018. ISBN 978-1-5386-6051-5.

[57] Jennifer Lock, Beaumie Kim, Kim Koh, and Gabrielle Wilcox. Navigating the Tensions of Innovative Assessment and Pedagogy in Higher Education. *Canadian Journal for the Scholarship of Teaching and Learning*, 9(1), April 2018. ISSN 1918-2902.

[58] Kirsti Lonka, Erkki Olkinuora, and Jarkko Mäkinen. Aspects and Prospects of Measuring Studying and Learning in Higher Education. *Educational Psychology Review*, 16(4):301–323, December 2004. ISSN 1040-726X, 1573-336X. doi: 10.1007/s10648-004-0002-1.

[59] Madeleine Lorås. From teaching assistants to learning assistants – lessons learned from learning assistant training at Excited. *Læring om læring*, 4(1), April 2020. ISSN 2535-4108.

[60] Madeleine Lorås, Rune Hjelsvold, Abdullah Bahmani, Shaun S Nykvist, and Andreas Krokan. Online Teaching and Learning - First impressions from students as NTNU transitions to an online only mode of learning. Technical report, NTNU, Trondheim, 2020.

[61] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. Introductory Programming: A Systematic Literature Review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE 2018 Companion, pages 55–106, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-6223-8. doi: 10.1145/3293881.3295779.

[62] Anna Makles. Stata Tip 110: How to Get the Optimal K-Means Cluster Solution. *The Stata Journal*, 12(2):347–351, June 2012. ISSN 1536-867X. doi: 10.1177/1536867X1201200213.

[63] Lauri Malmi, Judy Sheard, Päivi Kinnunen, and Jane Sinclair. Computing education theories: What are they and how are they used? In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, pages 187–197.

[64] Lauri Malmi, Judy Sheard, Päivi Kinnunen, Simon, and Jane Sinclair. Theories and Models of Emotions, Attitudes, and Self-Efficacy in the Context of Programming Education. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*, ICER '20, pages 36–47, New York, NY, USA, August 2020. ACM. ISBN 978-1-4503-7092-9. doi: 10.1145/3372782.3406279.

[65] F. Marton and R. Säljö. On Qualitative Differences in Learning: I—Outcome and Process*. *British Journal of Educational Psychology*, 46(1):4–11, February 1976. ISSN 2044-8279. doi: 10.1111/j.2044-8279.1976.tb02980.x.

[66] Joseph A. Maxwell. Paradigms or Toolkits? Philosophical and Methodological Positions as Heuristics for Mixed Methods Research. *Mid-Western Educational Researcher*, 24(2):27–30, 2011. ISSN 1056-3997.

[67] Robert McCartney, Anna Eckerdal, Jan Erik Mostrom, Kate Sanders, and Carol Zander. Successful Students' Strategies for Getting Unstuck. In *Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ITiCSE '07, pages 156–160, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-610-3. doi: 10.1145/1268784.1268831.

[68] Mitchell J. Nathan and R. Keith Sawyer. Foundations of the Learning Sciences. In R. Keith Sawyer, editor, *The Cambridge Handbook of the Learning Sciences*, pages 21–43. Cambridge University Press, Cambridge, 2 edition, 2014. ISBN 978-1-139-51952-6. doi: 10.1017/CBO9781139519526.004.

[69] National Academies of Sciences, Engineering, and Medicine and others. *Assessing and responding to the growth of computer science undergraduate enrollments*. National Academies Press, 2018.

[70] Philip M. Newton. The Learning Styles Myth is Thriving in Higher Education. *Frontiers in Psychology*, 6, 2015. ISSN 1664-1078. doi: 10.3389/fpsyg.2015.01908.

[71] NOKUT. Studiebarometeret. Norwegian Agency for Quality Assurance in Education, Oslo, Norway, April 2019.

[72] Sarath A. Nonis and Gail I. Hudson. Academic Performance of College Students: Influence of Time Spent Studying and Working. *Journal of Education for Business*, 81(3):151–159, January 2006. ISSN 0883-2323. doi: 10.3200/JOEB.81.3.151-159.

[73] Ola Nordal. NUSSE. Great Norwegian Encyclopedia, Oslo, Norway, February 2018.

[74] The Glossary of Education Reform. Learning Environment. may 2013.

[75] Anthony J Onwuegbuzie and R Burke Johnson. The validity issue in mixed research. *Research in the Schools*, 13(1):48–63, 2006.

[76] Anthony J Onwuegbuzie and Nancy L Leech. On becoming a pragmatic researcher: The importance of combining quantitative and qualitative research methodologies. *International journal of social research methodology*, 8(5):375–387, 2005.

[77] Seymour Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc., New York, NY, USA, 1980. ISBN 978-0-465-04627-0.

[78] Andrew Petersen, Michelle Craig, Jennifer Campbell, and Anya Tafliovich. Revisiting Why Students Drop CS1. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, Koli Calling '16, pages 71–80, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4770-9. doi: 10.1145/2999541.2999552.

[79] Jean Piaget. *Psychology and Epistemology: Towards a Theory of Knowledge*. Penguin Books, reprint edition edition, May 1977. ISBN 978-0-14-004482-9.

[80] Paul R Pintrich et al. *A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ).* ERIC, 1991.

[81] James Prather, Raymond Pettit, Kayla McMurry, Alani Peters, John Homer, and Maxine Cohen. Metacognitive Difficulties Faced by Novice Programmers in Automated Assessment Tools. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*, ICER '18, pages 41–50, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5628-2. doi: 10.1145/3230977.3230981.

[82] James Prather, Brett A. Becker, Michelle Craig, Paul Denny, Dastyni Loksa, and Lauren Margulieux. What Do We Think We Think We Are Doing? Metacognition and Self-Regulation in Programming. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*, ICER '20, pages 2–13, New York, NY, USA, August 2020. ACM. ISBN 978-1-4503-7092-9. doi: 10.1145/3372782.3406263.

[83] Anthony Robins, Janet Rountree, and Nathan Rountree. Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2): 137–172, June 2003. ISSN 0899-3408. doi: 10.1076/csed.13.2.137.14200.

[84] Anthony V. Robins, Lauren E. Margulieux, and Briana B. Morrison. Cognitive Sciences for Computing Education. In Sally A. Fincher and Anthony V. Robins, editors, *The Cambridge Handbook of Computing Education Research*, Cambridge Handbooks in Psychology, pages 231–275. Cambridge University Press, 2019. doi: 10.1017/9781108654555.010.

[85] Colin Robson. *Real World Research*. Wiley-Blackwell, New Jersey, 2 edition, March 2002. ISBN 978-0-631-21305-5.

[86] Matthew Scheidt, Ryan Senkpeil, John Chen, Allison Godwin, and Edward Berger. Sat does not spell success: How non-cognitive factors can explain variance in the gpa of undergraduate engineering and computer science students. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–5, 2018. doi: 10.1109/FIE.2018.8658989.

[87] Dale Schunk. *Learning Theories: An Educational Perspective*. Pearson, New Jersey, 5 edition, 2008. ISBN 978-0-137-07195-1.

[88] Dale H. Schunk. Self-Regulated Learning: The Educational Legacy of Paul R. Pintrich. *Educational Psychologist*, 40(2):85–94, June 2005. ISSN 0046-1520. doi: 10.1207/s15326985ep4002_3.

[89] Dale H. Schunk. Metacognition, Self-Regulation, and Self-Regulated Learning: Research Recommendations. *Educational Psychology Review*, 20(4):463–467, December 2008. ISSN 1573-336X. doi: 10.1007/s10648-008-9086-3.

[90] Corwin Senko, Chris S. Hulleman, and Judith M. Harackiewicz. Achievement Goal Theory at the Crossroads: Old Controversies, Current Challenges, and New Directions. *Educational Psychologist*, 46(1):26–47, January 2011. ISSN 0046-1520. doi: 10.1080/00461520.2011.538646.

[91] J. Sheard, A. Carbone, D. Chinn, and M. Laakso. Study Habits of CS 1 Students: What Do They Say They Do? In *2013 Learning and Teaching in Computing and Engineering*, pages 122–129, March 2013. doi: 10.1109/LaTiCE.2013.46.

[92] Kerry Shephard. Higher Education Pedagogy. In Sally A. Fincher and Anthony V. Robins, editors, *The Cambridge Handbook of Computing Education Research*, Cambridge Handbooks in Psychology, pages 276–291. Cambridge University Press, 2019. doi: 10.1017/9781108654555.011.

[93] George Siemens. Connectivism: A Learning Theory for the Digital Age. *Medical teacher*, 38(10):1064–1069, 2016.

[94] Simon Simon. *Emergence of computing education as a research discipline*. Aalto University, 2015. ISBN 978-952-60-6416-1.

[95] StataCorp. *Stata: Release 16*. StataCorp LLC., College Station, TX, 2019. ISBN 978-1-59718-301-7.

[96] Katrien Struyven, Filip Dochy, Steven Janssens, and Sarah Gielen. On the dynamics of students' approaches to learning: The effects of the teaching/learning environment. *Learning and Instruction*, 16(4):279–294, August 2006. ISSN 0959-4752. doi: 10.1016/j.learninstruc.2006.07.001.

[97] Claudia Szabo, Nickolas Falkner, Andrew Petersen, Heather Bort, Kathryn Cunningham, Peter Donaldson, Arto Hellas, James Robinson, and Judy Sheard. Review and Use of Learning Theories within Computer Science Education Research: Primer for Researchers and Practitioners. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*, ITiCSE-WGR '19, pages 89–109, New York, NY, USA, December 2019. ACM. ISBN 978-1-4503-7567-2. doi: 10.1145/3344429.3372504.

[98] Abbas Tashakkori and Charles Teddlie. *Handbook of Mixed Methods in Social & Behavioral Research*. SAGE, 2003. ISBN 0-7619-2073-0.

[99] Deborah Tatar. The Design Tensions Framework. *Human–Computer Interaction*, 22(4):413–451, November 2007. ISSN 0737-0024. doi: 10.1080/07370020701638814.

[100] Santiago Tejedor, Laura Cervi, Ana Pérez-Escoda, and Fernanda Tusa Jumbo. Digital Literacy and Higher Education during COVID-19 Lockdown: Spain, Italy, and Ecuador. *Publications*, 8(4):48, December 2020. doi: 10.3390/publications8040048.

[101] Neena Thota. Repertory grid: Investigating personal constructs of novice programmers. In *Proceedings of the 11th koli calling international conference on computing education research*, Koli calling '11, pages 23–32, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1052-9. doi: 10.1145/2094131.2094137.

[102] Neena Thota, Anders Berglund, and Tony Clear. Illustration of Paradigm Pluralism in Computing Education Research. In *Proceedings of the Australasian Conference on Computing Education*, volume 123, page 10, Melbourne, Australia, 2012. Australian Computer Society, Inc.

[103] Vincent Tinto. Stages of Student Departure: Reflections on the Longitudinal Character of Student Leaving. *The Journal of Higher Education*, 59(4):438–455, 1988. ISSN 0022-1546. doi: 10.2307/1981920.

[104] Denise Tolhurst. The influence of web-supported independent activities and small group work on students' epistemological beliefs. In *Proceedings of the sixth australasian conference on computing education - volume 30*, ACE '04, pages 311–316, AUS, 2004. Australian Computer Society, Inc.

[105] Tara Tressel, Susanne P. Lajoie, and Melissa C. Duffy. A guide for study terminology: Reviewing a fragmented domain. *Canadian Psychology/Psychologie canadienne*, 60(2):115, 2019. ISSN 1878-7304. doi: 10.1037/cap0000138.

[106] Trochim, Donnelly, and Arora Kanika. *Research Methods: The Essential Knowledge Base*. Cengage Learning, January 2015. ISBN 978-1-305-44518-5.

[107] Paul R Trowler et al. *Academic tribes and territories*. McGraw-Hill Education (UK), 2001.

[108] Karthikeyan Umapathy and Albert D. Ritzhaupt. A Meta-Analysis of Pair-Programming in Computer Programming Courses: Implications for Educational Practice. *ACM Transactions on Computing Education*, 17(4):1–13, September 2017. ISSN 1946-6226. doi: 10.1145/2996201.

[109] Arto Vihavainen, Juha Helminen, and Petri Ihantola. How novices tackle their first lines of code in an IDE: Analysis of programming session traces. In *Proceedings of the 14th koli calling international conference on computing education research*, Koli calling '14, pages 109–116, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3065-7. doi: 10.1145/2674683.2674692.

[110] L. S. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, October 1980. ISBN 978-0-674-07668-6.

[111] Etienne Wenger. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, 1998. ISBN 978-0-521-66363-2.

[112] Salla Willman, Rolf Lindén, Erkki Kaila, Teemu Rajala, Mikko-Jussi Laakso, and Tapio Salakoski. On study habits on an introductory course on programming. *Computer Science Education*, 25(3):276–291, July 2015. ISSN 0899-3408. doi: 10.1080/08993408.2015.1073829.

[113] Robert K. Yin. *Case Study Research: Design and Methods*, volume 5 of *Applied social research methods*. SAGE Publications, California, 4 edition, 2009. ISBN 978-1-4833-2224-7.

[114] Yusuf F. Zakariya, K. Bjørkestøl, H. K. Nilsen, S. Goodchild, and M. Lorås. University students' learning approaches: An adaptation of the revised two-factor study process questionnaire to Norwegian. *Studies in Educational Evaluation*, 64: 100816, March 2020. ISSN 0191-491X. doi: 10.1016/j.stueduc.2019.100816.

[115] Mark Zarb, Bedour Alshaigy, Dennis Bouvier, Richard Glassey, Janet Hughes, and Charles Riedesel. An international investigation into student concerns regarding transition into higher education computing. In *Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education*, ITiCSE 2018 companion, pages 107–129, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-6223-8. doi: 10.1145/3293881.3295780.

[116] Barry J. Zimmerman and Dale H. Schunk. *Self-Regulated Learning and Academic Achievement: Theoretical Perspectives*. Routledge, May 2001. ISBN 978-1-135-65914-1.

 80

Appendices

Appendix A

Interview Guides

Interview guides in English for Study 2 and 5 are included in the following pages. The original Norwegian versions are available upon request.

Study 2 – Interview guide

[Translated from Norwegian to English by Madeleine Lorås (2021). The translation is literal and does not account for nuances or terminology differences in English. The wording of the questions should not be reused without adaption.]

Interview 1

Introduction
- Information about the study
- Information about what it means to participate

Do you have any questions about the study?

<Go through consent form and sign.>

Choosing IT
- Tell me about yourself?
- Where did you go to upper secondary school?
- What courses did you take?
- Why did you/did you not take IT-courses?
- Can you describe your first encounter with IT / informatics / programming?
- Why did you choose NTNU?

The first semester

Academics
- What has it been like to be a student so far?
    - Is it as expected?
- What is your favorite course? Why?
- What is your least favorite course? Why?
- How do you study?
    - What does a typical day/week look like?
- How is your contact with lecturers?
    - What about the teaching assistants?
    - Would you like there to be more?

Socially
- What has it been like socially?
    - Have you made any friends?
    - Joined any organizations?
- What do you like doing when you're not studying?

Do you have any ideas for what could improve the first semester so far?

Anything else you would like to add?

Do you have any questions for me?

Interview 2

Follow up from last time
How are you doing? How are things?

In the last interview we talked about <make some notes on each participant and ask follow-up questions specifically>.

What are your thoughts at this point about studying IT?
- How has it changed?

The second semester
What has it been like to be a student this semester?
- What has been difficult?
- What has been good?
- Have you been to the Study day?

How would you describe your motivation for the study as a whole?
- Motivation for IT job?
- What increases/decreases motivation for you?

Exam periods
How did it go during the exam period?

How do you prepare for exams?
- Different or similar to your normal routines?

Study technique
How do you learn?
- What do you learn most from?
- What activities provide learning for you?
- Who do you learn from?

Can you describe a good learning situation?
- Can you describe the last time you experience mastery?

What is the most important thing for you to learn as much as possible?

Anything else you would like to add?

Do you have any questions for me?

Interview 3

Looking back at the first year

How would you describe your study habits during the first year?

- First semester, second semester, now?

How did you change? Why?

- Structure, study techniques/strategies, motivation

  What advice would you have given yourself at the beginning of the first year now?

Looking at the findings so far

1) The study habits of the students change throughout the year
   - Start: learning-focused: motivated by being new, learning and academic content in focus
   - End: assignment-focused: shortcuts, lower motivation, more work, assignments and exams in focus
2) One subject's focus / priority is affected by the whole, i.e., the other subjects.

What are your thoughts on this?

How does your experience relate to these findings? How are they different?

Taking the Study Process Questionnaire

Do you think your answers would be different a year ago?

Anything else you would like to add?

Study 5 – Interview guide

Introduction

Introduction
- Information about the study
- Information about what it means to participate

Do you have any questions about the study?

<Go through consent form and sign.>

Part 1 – Background information
- Study program
- Class level
- How do you like being a student in this program?
- Where in Norway are you now?
- How would you describe yourself as a student?

Part 2 – Online teaching and learning

Online learning activities:
- How do you experience learning activities in this online world?
    - Do you have any examples of something that works very well / better?
    - Not so good / worse?
- How do you experience:
    - Lectures
    - Exercises / exercises
    - Access to resources
    - Information / communication
    - Individual activities
- Do you actively participate in any of the teaching? How? Why not?
- Do you have all the equipment you need?
- Did you feel prepared for this situation? How so? Why not?

The learning environment:
- Where are you studying now? Can you describe your physical work environment?
- How does this work? Better / worse?
- What are your thoughts/feelings about the home exam?

Study habits:
- How would you describe a regular study day for you now?
    - Do you have any examples of something that works very well / better?
    - Not so good / worse?
- What do you learn most from?
- How is your motivation?

Help and support:

- Where do you go for help if you need it now? Who?
    - Have you taken advantage of this? How? Why not?
- How is this different / similar in relation to "regular" teaching?
- In terms of help and support, how do you experience:
    - Using different forums
    - Learning assistants
    - To contact fellow students
    - Teacher / lecturer

Feedback and assessment:

- Have you received any feedback?
    - How did that work? What technology did you use?
    - Are there any better / worse differences?
    - Is it important to you to get good feedback?
- Have you received any assessments?
    - How did that work? What technology did you use?
    - Are there any better / worse differences?
    - Is it important to you?

Social:

- Do you have any contact with fellow students during this period?
    - How did that work? What technology did you use?
    - Is it important to you?

Do you have anything more to add?

Appendix B

Self Report Questionnaire

The self report questionnaire in English for Study 4 and 5 is included in the following pages. The original Norwegian versions are available upon request.

Self-report questionnaire

[Translated from Norwegian to English by Madeleine Lorås (2021). The translation is literal and does not account for nuances or terminology differences in English. The wording of the questions should not be reused without adaption.]

Pre-questionnaire

The purpose of this research is to investigate and map how IT students' study. Specifically, what skills, knowledge, and strategies students use when studying in an IT study program. IT in this context is a collective term for computer technology, information technology, communication technology and informatics.

All information about you and data collected will be treated confidentially. In material that is written or otherwise presented to others, all persons involved will be anonymized. All data that can be used to identify people will be deleted after the project is completed, no later than August 2020.

None of your answers will be used in any way in your assessment. That is, it will not count on your final grade. Your answers will in no way be used to judge you or your skills. It is important that you answer as honestly as you can.

Questions, comments or other input can be directed to Madeleine Lorås (madeleine.loras@ntnu.no).

1 Gender

Choose one	
Male	
Female	
Prefer not to answer	

2 What previous experience do you have with IT?

By IT we include computer science, information technology, programming, informatics etc.

	Yes	No
I have no previous experience with IT.		
I have learned a little programming and the like at home on my own.		
I have participated in informal computing activities such as Code club, summer schools and Code hour etc.		
I have taken IT1 and/or 2 at upper secondary school.		

3 What was your grade point average from upper secondary school?

With GPA we mean the average of grades received. Not including potential extra points.

4 Norwegian version of the Study Process Questionnaire

See the following for a description of the Norwegian version.

Zakariya, Y. F., Bjørkestøl, K., Nilsen, H. K., Goodchild, S., & Lorås, M. (2020). University students' learning approaches: An adaptation of the revised two-factor study process questionnaire to Norwegian. *Studies in Educational Evaluation*, *64*, 100816. https://doi.org/10.1016/j.stueduc.2019.100816

For an original English version, see:

Biggs, J., Kember, D., & Leung, D. Y. P. (2001). The revised two-factor Study Process Questionnaire: R-SPQ-2F. British Journal of Educational Psychology, 71(1), 133–149. https://doi.org/10.1348/000709901158433

5 To what degree to you agree with the following questions about motivation

	Completely disagree		…		Completely agree
I am motivated to learn programming and IT.					
I am motivated to learn math.					
I am motivated for study efforts in general.					

6 What goal do you have for your grades?

Choose one	
To pass	
Slightly above passing	
On average	
Slightly above average	
Well above average	

7 Consent

Participation in the study is voluntary and it is possible to withdraw at any time by sending an email to Madeleine Lorås: madeleine.loras@ntnu.no

You can also choose to participate in only parts of the study.

None of your answers will in any way be linked to you as a person. When you submit this form, your answers will receive an identifier which means that it is not possible to link your answers to your name.

I consent to...

Choose the ones that apply for you	
... my answers to these questionnaires being used for research.	
... my answers to these questionnaires in connection with assignments being used for research.	
... my course results (grades, score and answer) being retrieved and used for research.	

Weekly reports

1 To what extent did you use the following place to work on this week's assignment?

	To a large extent	To some extent	To little extent	Never
At home				
In the labs at the Science Building				
In other study areas at the Science Building or other on campus areas				
At libraries on campus				
In the cafeteria				
In areas off campus (cafes, public library, etc.)				

2 How much time (in hours) did you spend on the following activities related to this assignment?

	Nothing	1	2	3	4	5	More than 5
Lectures (theory, ordinary, practice lecture)							
Assignment lecture*							
Working with a teaching assistant in the labs							
Working alone							
Working alone, but with other students*							
Collaboration with other students							
Doing the assignment							
At the Study Day Initiative*							

3 When did you work on this assignment?

Answer for the days you worked on the assignment.

	Before 12:00	Between 12:00-17:00	After 17:00	Large parts of the day	Nothing this day
Monday					
Tuesday					
Wednesday					
Thursday					
Friday					
Saturday					
Sunday					

4 To what extent did you use the following techniques to achieve the learning objectives in this assignment?

	To a large extent	To some extent	To little extent	Never
Reading in the syllabus book				
Reading relevant texts on the internet (slides, resources on Blackboard or other websites)				
Taking notes (from the book or the internet)				
Programming examples from lectures				
Programming examples you have found yourself				
Programming / solving the assignments				
Watching videos				
Memorizing				
Drawing diagrams*				

Comment:
Other things you may have spent time on.

5 How much of this did you already know?

Most/everything	A lot	Some	A little	Very little/nothing

6 How much do you feel you have learned in CS1 during this week?

Very much	Much	Medium	Little	Very little

7 How satisfied are you with your own study efforts in CS1 this week?

Very satisfied	Satisfied	Moderately satisfied	Slightly unsatisfied	Very unsatisfied

8 Which program are you enrolled in?
(Choose from list not included here)

9 Compared to CS1, how much have you worked on the following courses:

	Less	Approximately the same	More	A lot more	Not relevant for me
Calculus 1					
Scientific philosophy					
Discrete mathematics					
Math 1					
Web technology					
Linear algebra					
Program specific course					

10 How satisfied are you with your own study efforts in general this week?
In all subjects combined.

Very satisfied	Satisfied	Moderately satisfied	Slightly unsatisfied	Very unsatisfied

* Added in the second semester to account for some differences in courses.

96

Appendix C

Supplementary Material

The dataset for **Paper 1** can be found on this link:
https://doi.org/10.18710/MWLHOA

The dataset from the questionnaires in **Paper 2** can be found on this link:
https://doi.org/10.18710/7TUIJL

The dataset for **Paper 3** can be found on this link:
https://doi.org/10.18710/JQX7NW

The dataset for **Paper 4** can be found on this link:
https://doi.org/10.18710/YLVIAN

An overview of themes, codes and exemplary quotes from **Paper 8**
can be found on this link:
https://doi.org/10.18710/O8FCIK.

Part II

Part II: Collection of Research Papers

Paper 1

First Year Computer Science Education in Norway

Madeleine Lorås, Trond Aalberg, Guttorm Sindre

UDIT 2018

Authors' contributions: Lorås led the paper writing and was the main author. Lorås, Aalberg and Sindre designed and supervised the study. Lorås collected the data, performed the analysis and wrote the paper. Aalberg and Sindre provided general supervision of the research and the paper writing.

 102

FIRST YEAR COMPUTER SCIENCE EDUCATION IN NORWAY

A mapping study of computer science study programs in Norway focusing on the first year

Madeleine Lorås
PhD candidate
Department of Computer Science, NTNU

Guttorm Sindre
Professor
Department of Computer Science, NTNU

Trond Aalberg
Associate professor
Department of Computer Science, NTNU

The need for ICT knowledge in Norway is increasing and the demand for candidates is currently higher than the number of students graduating. It has been identified that the first-year experience is crucial to student motivation and throughput of study programs, therefore it is interesting to look at the state of the art of computer science study programs in Norway. In this paper we present a survey and study of the number of undergraduate computer science programs in Norway and map their characteristics in order to gather an up to date overview of the selection of programs. Through a systematic review of all Norwegian undergraduate programs using data from national databases we have found that there are 12 institutions offering 56 different programs in Norway in 2018. The study showed that the characteristics of these programs vary, that is, the amount of computer science courses during the first year, the number of students, admission requirements, student satisfaction and time commitment. This article presents these findings along with an analysis of what characteristics impact the students' contentment and learning experience.

KEY WORDS: Education, computer science, first year experience

1 INTRODUCTION

Norway will in the near future face a shortage of computer and information science professionals is the conclusion of a report done for the Ministry of Local Government and Modernization in 2017. The need for advanced information and communication technology is increasing, and with the current student enrollment and graduation rates there will be a gap between supply and demand (Ministry of Local Government and Modernization, 2014). These predictions are in line with the situation in other parts of the world, for instance USA. The National Academies of Sciences published a report last year concluding that although the number of bachelor's degrees in computer and information science has increased substantially, there is indeed a gap to be filled as far as industry need (2017). In addition, this report stresses the fact that this massive growth will in the near future demand a number of computer science educators the sector will not be able to fill. Especially in higher education, the fact that "over half of new PhDs are drawn to opportunities in the industry, hiring and retaining CS faculty is currently an acute challenge that limits institutions' abilities to respond to the increasing CS enrollment" (National Academies of Sciences, 2017, p. 5).

When it comes to solving the problem of increasing demand, high computer science student enrollment and the possible shortage of computer science educators, both reports have several recommendations. Firstly, it is important to state that the high student enrollment problem of course can be solved by limiting the number of students in computer science study programs, however, the consequences of doing so should be considered comprehensively, and the benefits and costs weighed for the entire university community (National Academies of Sciences, 2017). Furthermore, the Norwegian report suggests nine concrete actions, where six of them directly concern higher education. Summarized, these actions are concerned with increasing the number of graduates by increasing throughput. In order to do so, the National Academies of Sciences recommends actions to support diversity and to facilitate an improved understanding of national undergraduate enrollment trends. Therefore, this study has aimed to provide an overview of what computer science programs exist in Norway today and how they are prepared to meet these demands. The research inquiry is as follows:

What characterizes the first year of computer science study programs in Norway?

- How are they designed?
- What impacts student contentment and learning experience?

2 DEFINITIONS AND BACKGROUND

2.1 Defining computer science education

In Norway the term information and communication technology, ICT, is used as an umbrella term for all things computing and computer technology. Regardless if the accurateness of this, or personal preference, it is in this case important to have a common understanding of the terms. For the purpose of this paper the term computer science is used consistently, with the understanding that the term includes what we in Norway call ICT: computing, informatics, information and computer technology.

When it comes to computer science education, the various universities and colleges have different ways to further define and divide their departments and study programs. In Norwegian higher education there are two major stakeholders who have an important role in computer science education; The Norwegian Universities and Colleges Admission Service (NUCAS) and the Norwegian Agency for Quality Assurance in Education (NOKUT) (NOKUT, 2018b; Samordna Opptak, 2018a). NUCAS handles all applications and admissions to public undergraduate education in Norway. All students wishing to study at any public university or college in Norway must go through their web portal, which means that NUCAS gathers data on grade point averages and student admission numbers. Furthermore, NOKUT is the organization who accredits the various study programs and is in charge of quality assurance across all higher education institutions in Norway, public as well as private. Part of the work with quality assurance is a national survey of all study programs called Studiebarometeret (NOKUT, 2018b). The survey asks for the students' perceptions of educational quality in their study programs and is sent out to 60 000 students each fall. In addition to the valuable data gathered by NUCAS and NOKUT, the way these organizations categorize the various study programs is important for the purpose of this paper. This will be described in detail in the methodology section.

2.2 The importance of the first year

In order to meet future demands for computer science (hereby referred to as CS) it has been identified that increasing the number of CS graduates is essential. This means decreasing the drop-out rates and increasing throughput. Research by Vincent Tinto on student departure identifies the first-year experience as crucial for retention of students (Tinto, 1975). Tinto discusses student departure as several stages; separation (from a known home environment), transition (into a new social and physical structure) and incorporation (into a community and culture) and argues that the students' first year experience lays important groundwork, even though students may drop out later in their study (Tinto, 1988). An important part of this groundwork is related to what learning strategies and study skills the students develop during this time (Adams, Berzonsky, & Keating, 2006; Blickle Gerhard, 1996). Therefore, this study has chosen to focus on the first year for CS study programs, the admission process and students time commitment.

3 METHODS

The purpose of this study was to survey and categorize all undergraduate computer science study programs in Norway, focusing on the first year. In the following sections the data collection process, inclusion criteria and method of analysis will be described further.

3.1 Data collection

The first step in the data collection process was to make a list of all study programs within the aforementioned CS definition. For this, three approaches were taken. Firstly, a list of all the study programs within the NUCAS database was made. Secondly, this list was compared and reviewed according to the list of study programs from the NOKUT database. These two sources provided a list of 54 study programs. Additionally, a manual search was performed in both databases for the key words "informatics, computer science, computer technology and ICT". This provided two additional study programs to the list, making it a total of 56 study programs. A full list of these study programs along with selected variables can be found in Appendix A.

The next step of the data collection process was to combine the data from the two databases. This was done manually creating a spreadsheet with data on grade point averages (GPAs), admission requirements, student numbers, gender balance and survey data from Studiebarometeret.

The final step of data collection was to survey the first year of these study programs in order to categorize the various courses and their content. The researchers manually looked up each study program's web page and added the various courses to the spreadsheet. For each course the name and number of credits was documented, as well as a category indicating what kind of course it was. These categories are described in Table 1.

Label	Category	Explanation
P	Programming course	Courses about or involving a lot of programming.
D	Computer science course	Courses about topics in computer science not revolved around programming.
M	Mathematics course	Courses in mathematics.
F	Scientific philosophy	Courses in scientific philosophy and/or ethics.
A	Miscellaneous	Other courses. Including, but not limited to, economics, physics, finance and engineering.

Table 1: Overview and explanation of the different course categorizations.

The basis for this categorization was the name of the course and the learning goals listed on the web page. This was done independently by two researchers. The two researchers reached an agreement level of 81%. The researches then discussed the various differences and agreed on the final categorization. Most of the disagreements were related to a systematic difference of opinion. For instance, whether a web development course was to be considered a programming course or a computer science course (the researchers concluded the former). A small number of discrepancies were due to errors in the data gathering process, copy/paste errors, which were easily corrected in this process.

In addition to this general survey, all the programming courses were further investigated to categorize what programming language was used. This assessment was based on the course website information about content and learning coals, as well as any available syllabuses. This process also revealed some discrepancies, where a course which was given category P in reality was a D. However, some descriptions did not reveal what language was used, still it was clear that it was a programming course. These instances were given the value missing (.).

3.2 Inclusion criteria

Following the methodology of a general systematic review there is a need to identify some defined inclusion criteria (Booth, Sutton, & Papaioannou, 2016; Kitchenham, 2004). In order to exclude non-CS study programs, the researchers used the pre-defined categories "information technology and informatics" and "information and computer technology" in the NUCAS and NOKUT, respectively, to find study programs. In addition, all included study programs had to have 15 credits or more in programming or computer courses during the first year, with at least 7,5 credits per semester. Since the focus of this study was the first year, only bachelors programs and 5-year integrated masters programs were included. Additionally, study programs that were online, flexible or not full time were excluded because they are not comparable to on campus programs in this regard. Finally, in order to use data from Studiebarometeret, the study program had to have a sufficient amount of responses (defined by NOKUT). Although, six study programs did not have useable data in Studiebarometeret, they were still included in analysis which did not involve this data. In summary, the inclusion criteria and number of study programs was as follows:

- ICT study program (N= 86)
- Full time, Not online or flexible (N=58)
- Bachelor program or 5-year integrated master's program (N= 56)
- 15 credits or more in programming or computer courses during the first year, with at least 7.5 credits per semester (N=56)
- *Has usable data from Studiebarometeret (N=50)*

3.3 Method of analysis

The way the data was analyzed can be divided up into a descriptive and exploratory analysis. The descriptive analysis aimed to answer the research question concerning characteristics and design of CS study programs. Therefore, the analysis was focused on describing and summarizing the data, which in this case involved creating sorted lists identifying top and bottom study programs according to the different variables, as well as calculating averages. Furthermore, the exploratory results focused on identifying possible correlations between variables, and thus exploring what impacts student contentment and learning experience. Correlation in this study is defined as a "statistical relationship between two variables", and for calculating this Microsoft Excel was used to calculate the Pearson Product-Moment Correlation Coefficient (Ringdal, 2012, p. 321).

4 RESULTS

The results of this study are both descriptive and exploratory. The descriptive results summarize and characterizes the various study programs, while the analytical/exploratory results try to identify some important correlations and relationships.

4.1 Descriptive results

The descriptive results summarize some important data about the various study programs. These results have been further divided up into four categories: the first-year composition, the student body, admission criteria and time commitment.

4.1.1 First year composition

The first-year composition category describes the academic content of the first year according to the variables *amount of CS-courses* and *programming language used*. This data gives a general overview of how much and what kind of CS each study program has included. The amount of CS courses in the first year varies from 100% to 25%. This variable is calculated by adding the number of credits categorized as programming courses (P) to computer science courses (D). The study programs with less CS, fills up the year with mathematics courses (N=33), miscellaneous courses (N=28) and in some cases a scientific philosophy course (N=14). Figure 1 gives the full summary of each study program and the categorization of courses.

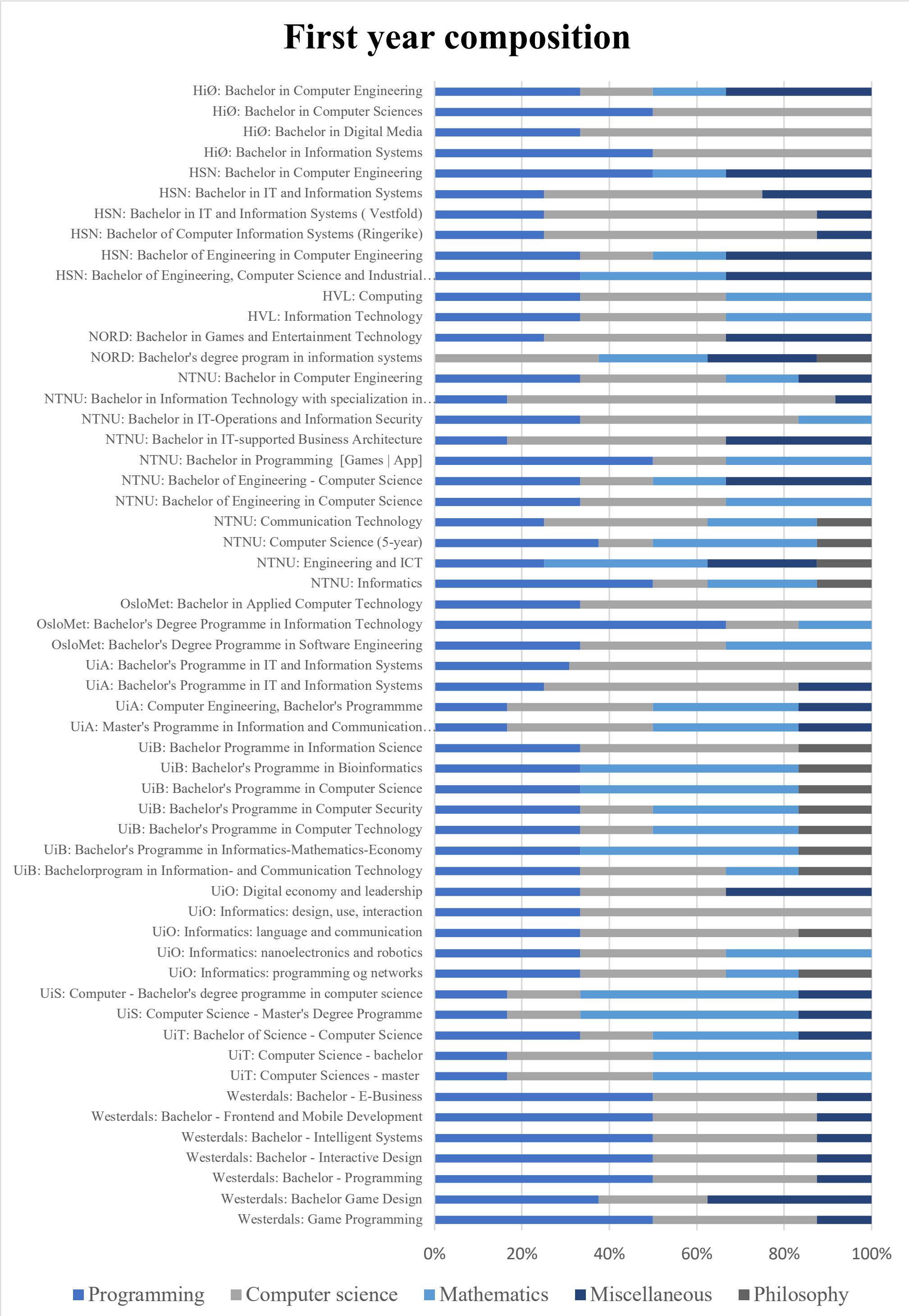


Figure 1: The composition of the first year of CS study programs in Norway, in alphabetical order.

When it comes to what programming language is used in the first year, this too varies. However, Java is by far the most popular programming language (N=48), followed by web-based languages such as HTML, CSS and JavaScript (N=25) and Python (N=18). A full summary can be found in Table 2.

Fall semester		**Spring semester**		**Total**
Arduino	0	Arduino	2	2
C	1	C	0	1
C#	2	C#	2	4
C++	3	C++	6	9
HTML, CSS	9	HTML, CSS	6	15
HTML, CSS, JavaScript	2	HTML, CSS, JavaScript	1	3
Java	15	Java	33	48
JavaScript	4	JavaScript	0	4
MATLAB	1	MATLAB	0	1
PHP, JavaScript	0	PHP, JavaScript	3	3
Python	9	Python	2	11
Missing (no data)	8	Missing (no data)	10	18

Table 2: Overview of programming languages used in the first year.

4.1.2 Student body

The student body category describes the composition of students according to the variables *number of students*, *gender balance*, and overall *satisfaction* with the study program. These numbers give a very general overview of the student population and their contentment.

The number of students in each study program and the corresponding gender balance is data gathered by Studiebarometeret via Database for Statistics on Higher Education. Overall, there are 8452 students enrolled in the included study programs. The number of students in each program vary from a total of 24 to 692, while the majority of programs have between 100 and 250 students. NTNU is the institution with the most CS study programs and also the most students in total with 2737 (32% of all CS students). Furthermore, the NTNU study programs Computer Science (engineering, 5-year) and Informatics (bachelor, 3-year) has the most students (N=692 and N=481 respectively), however they are not entirely comparable considering they are a different number of years. Nevertheless, the fact is that NTNU has more students than any of the other institutions as is evident in Table 3 below.

Institution	**Number of students**	**% of all CS students**	**CS programs**
NTNU	2737	32 %	11
UiO	811	11 %	5
OsloMet	642	9 %	3
UiA	772	11 %	4
UiB	686	10 %	7
HSN	717	10 %	6
HiØ	451	6 %	4
HVL	290	4 %	2
NORD	132	2 %	2
UiT	354	5 %	3

UiS	284	4 %	2
Westerdals	576	8 %	7

Table 3: Number of students at each institution. Note that one of UiOs programs started this fall, and therefore has 0 students in this statistic.

Gender balance in CS study programs is a much debated topic, and one that has gotten a lot of attention over the last decade. This study found that there are 1393 female students and 6910 male students enrolled in CS-programs in Norway. That gives a percentage of 17% in total, while the average percentage is 16%. The ten study programs with the highest female percentage is listed in Table 4.

Program	Students	Female	Male	% females
UiB: Bachelor's Programme in Bioinformatics	24	15	9	63 %
Westerdals: Bachelor - Interactive Design	89	48	41	54 %
UiO: Informatics: design, use, interaction	302	128	174	42 %
NTNU: Communication Technology	229	90	139	39 %
NTNU: Engineering and ICT	257	87	170	34 %
UiO: Informatics: language and communication	65	22	43	34 %
UiB: Bachelor's Programme in Informatics-Mathematics-Economy	24	7	17	29 %
HiØ: Bachelor in Digital Media	53	14	39	26 %
NTNU: Bachelor in IT-supported Business Architecture	183	45	138	25 %
UiS: Computer Science - Master's Degree Programme	70	15	55	21 %

Table 4: Top 10 study programs according to gender balance.

In the national survey, Studiebarometeret, students are asked a number of questions about their experience as a student in the various study programs on a five-point Likert scale. The questions are grouped by different categories, hence creating an index. The categories are teaching, learning environment, organization, influence, inspiration, engagement, relevance, exams and expectations. In this study an average of these indexes was used to create a variable for overall satisfaction, which can be considered an indicator of student contentment. The satisfaction among students in CS programs vary from 2,13 to 4,13, with an overall average of 3,67. The scale is from 1-5, where 5 is the most satisfied. Table 5 below shows the top ten study programs according to satisfaction.

Program	Satisfaction
UiT: Computer Sciences - master	4,13
Westerdals: Bachelor - E-Business	4,08
HiØ: Bachelor in Computer Sciences	4,00
Westerdals: Bachelor - Intelligent Systems	4,00
Westerdals: Bachelor - Programming	3,99
HiØ: Bachelor in Digital Media	3,97
NTNU: Bachelor of Engineering in Computer Science	3,96
UiO: Informatics: programming and networks	3,95
UiO: Informatics: nanoelectronics and robotics	3,94
UiA: Master's Programme in Information and Communication Technology	3,93

Table 5: Top 10 study programs according to satisfaction.

4.1.3 Admission criteria

The admission criteria category describes the characteristics of the students enrolling in a study program according to the variables *grade point average (GPA)* and *admission requirements*. These variables can be used to indicate the popularity of a program, as well as the quality of the students enrolling.

As described in section 2.1, all students wishing to enroll in a public institution have to apply via NUCAS. In these cases, the only deciding variable for admission is the students' GPA from upper secondary school. GPA in NUCAS consists of both the students actual grade average and possibly some extra points given for certain subjects or accomplishments. For example, student can receive four extra points for taking science courses in upper secondary school, or they might get extra points for military service.

For enrollment in a private institution, local guidelines apply. In this study the only private institution is Westerdals Oslo ACT, and according to their admissions office they generally admit all qualified candidates. The remaining 55 study programs uses GPA to distinguish candidates, where the students with the highest grades, including possible extra points, will be admitted. In some cases, when the number of candidates is equal to or lower than the number of places in the program, all qualified applicants may be enrolled (these have been given the value 30). The NUCAS database publishes enrollment data for each year, including all the study programs and their corresponding lowest admitted GPA (Samordna Opptak, 2018b). Table 6 lists the top ten study programs in 2016 and 2017.

Program	**GPA 2016**	**GPA 2017**
UiO: Digital economy and leadership	-	62,1
NTNU: Computer Science (5-year)	57,0	58,5
NTNU: Communication Technology	56,7	57,0
NTNU: Engineering and ICT	55,9	56,2
NTNU: Bachelor in Computer Engineering	53,1	55,6
NTNU: Informatics	51,5	53,2
UiB: Bachelor's Programme in Computer Science	48,9	53,1
UiO: Informatics: nanoelectronics and robotics	52,1	53,1
UiO: Informatics: programming and networks	51,0	53,1
UiO: Informatics: design, use, interaction	50,9	53,0

Table 6: Top 10 study programs according to GPA 17 in the regular admission[1]. Digital economy and leadership was created in 2017 and therefore has no data for 2016.

In addition to GPA, some study programs will also have an admission requirement. Some study programs require students to take a certain amount of math and science courses in order to qualify for admission. Table 7 summarizes the results and explains the various requirements found in CS study programs.

Requirement	**Explanation**	**Number of CS programs**
MATRS	Math for natural sciences level 1 OR Math for social sciences level 1 + 2	19
GENS	General admission, no special requirements	14
HING	Math for natural sciences level 1 OR Math for social sciences level 1 + 2	13

[1] There is also a quota of first time applicants, which is also often used, but in this case, it is the regular admission. That means all qualified applicants compete.

	Physics level 1	
REALFA	Math for natural sciences level 1 Math for natural sciences level 2 OR other science course level 1	4
ING4R2	Math for natural sciences level 1 OR Math for social sciences level 1 + 2 Math for natural sciences level 2 with grade minimum of 4 Physics level 1	3
SIVING	Math for natural sciences level 1 OR Math for social sciences level 1 + 2 Math for natural sciences level 2 Physics level 1	3

Table 7: Summary and explanation of the various requirements

4.1.4 Time commitment

The time commitment category describes the time students spend studying according to the variables *organized teaching activities* and *self-study*. These numbers give an overview of the students' time commitment which is an interesting possible indicator of education quality.

The total amount of time students in CS study programs spend studying varies from 20 hours a week to 52, while the average is 35 hours which is the national average for all students in Norway (NOKUT, 2018a). This total time commitment variable is calculated from student reported time spent in organized education (lectures, labs, etc.) and time spent studying independently (reading, doing assignments, alone and in groups etc.). Table 8 shows the top 10 study programs according to time commitment.

Program	**Organized education**	**Self-study**	**Time commitment**
UiT: Computer Sciences - master	10	42	52
UiS: Computer Science - Master's Degree Programme	13	31	44
UiA: Bachelor's Programme in IT and Information Systems	18	26	44
UiA: Bachelor's Programme in IT and Information Systems	18	26	44
HSN: Bachelor of Engineering in Computer Engineering	19	25	44
UiT: Computer Science - bachelor	5	38	44
HiØ: Bachelor in Computer Sciences	17	26	43
Westerdals: Bachelor - Programming	19	22	41
UiO: Informatics: nanoelectronics and robotics	17	23	40
Westerdals: Bachelor - E-Business	22	18	40

Table 8: Top 10 study programs according to time commitment

4.2 Exploratory results

In addition to these descriptive results the researchers were interested in investigating possible correlations between these variables. Especially, what had the most impact on time commitment and overall satisfaction. Therefore, a correlation analysis was done comparing the various variables described above with time commitment and satisfaction. The results of this correlation analysis are shown in Table 9.

Variable	Correlation	
	Time commitment	**Satisfaction**

Number of students	0,19	0,12
Number of females	0,09	0,07
Number of males	0,21	0,13
GPA	0,24	0,04
Has math requirement	0,09	-0,23
Has natural science level 2 math requirement	0,26	-0,05
Has math and science requirement	0,14	-0,11
Amount CS in first year	-0,02	0,11

Table 9: Results of correlation analysis.

5 DISCUSSION

In the following section the results presented above will be discussed further following the same categorization. This discussion includes both descriptive and exploratory results, as well as reliability and validity considerations.

5.1 First year composition

As showed in Figure 1 the design of the first year CS study programs vary considerably. Notably, there seems to be no correlation between the amount of CS courses and time commitment or satisfaction. One might assume that students pursuing a degree in computer science would be more satisfied with a study program with a high CS content, however, these results indicate otherwise. On the other side, all included study programs have CS courses in both semesters. Subsequently, the categorization process might not reflect the full content of these courses. The quality of the course websites varied considerably, and it is possible they were not up to date.

When it comes to programming languages taught in the first year it is not surprising that Java is the most popular programming language. However, it is interesting that Java is most common in the second semester. Additionally, that web-based languages are equally popular in the first semester. The debate about what programming language is the best to start out with is ongoing, and this study does not aim to settle this debate. Nevertheless, these findings can provide an interesting base for further research on the topic.

5.2 Student body

The results on the topic of number of students and number of CS study programs vary considerably, therefore it is impossible to draw any conclusions as to what characterizes CS study programs in Norway accordingly. However, the numbers do reflect the changes the higher education reform implemented over the last four years (Kunnskapsdepartementet, 2015). Several institutions have merged which has changed the dynamics in Norwegian higher education. In the case of CS education, NTNU and HSN (now USN) has been the most impacted, as is evident from the number of study programs.

On the topic of gender balance the graphs have been pointing slightly upwards over the last couple of years, at least according to numbers from NTNU (DBH, 2018). However, a total average of 17% female students is not high enough. Especially considering that recruiting more female candidates is the best source to increasing CS enrollment. An interesting observation from the results of this study is that two of the three top study programs all include design of some sort, while the two bottom study programs are both related to game programming (Westerdals: Game Programming, 1,9%, and NTNU: Bachelor in Programming [Games | App], 5,2 %). These results seem to confirm that certain stereotypes and possible misconceptions are indeed present in the student population. However, more research into this topic is needed before any conclusions can be made.

When it comes to student satisfaction the overall average of 3,68 indicates that students in CS study programs in Norway are generally very content. While contentment is a subjective interpretation of the students' experience, this is still a variable that can be used in study quality assurance work. Considering the results from the correlation analysis it is interesting to determine that not one factor, out of these variables, seem to have any considerable impact on student satisfaction. The highest correlation is the math requirement with -0.23, which is difficult to interpret and needs more research. The math requirements may result in enrolled students that are more dedicated or more hard working, which in turn may lead to increased student satisfaction. On the other side, the unit of analysis in this case was the study program, and it might be more interesting to investigate at the individual student. Therefore, the researchers plan on continuing the work with this by examining individual student data from Studiebarometeret.

5.3 Admission criteria

The GPA variable along with the number of applicants is often used as a measure of popularity and prestige by the institutions. In addition, this number can give some indication of the quality of students enrolled. The study programs with higher GPA are enrolling students who performed well in upper secondary school, which would seem to indicate "good students" in higher education. However, the correlation analysis done in this study only produces a value of 0,24 between time commitment and GPA, which can indicate otherwise. That is, if one considers time commitment as an indicator for quality of the student. In this case, the high performing students in upper secondary school can be spending less time studying in higher education because they have a good knowledge base to build on. However, the correlation indicates that GPA has a positive impact on time commitment. Looking at GPA in higher education would perhaps be a better indicator, unfortunately these numbers were not available in the data used in this study. On the other side, grades in higher education are more difficult to compare considering there are no national exams or such, as there is in upper secondary education.

Considering the admission requirements for CS study program they can be divided up into various levels of math requirements. Only 14 study programs do not require any math, which additionally do not have any requirements at all. Consecutively there was 23 study programs with no math courses in the first year, however they might include math later in the program. The remainder of study programs require some level of math, and in some cases also some type of science course. In this regard it is striking that no study programs have CS as a prerequisite, however, four study programs do have it as a possibility (REALFA). When it comes to the impact of these requirements on time commitment and satisfaction, the only notable correlation is the math for natural sciences level 2 requirement on time commitment which is 0,26.

5.4 Time commitment

Time commitment in CS study programs on average is within the norm for Norwegian students, however compared to a traditional work-week in Norway it is a bit low. Notably, for a large majority of study programs students spend more time on self-study than in organized teaching activities (N=43). The findings of this study do not reflect the reason for this, or what kind of activities the students are doing, but considering the number of students in CS study programs is increasing this might be an increasing number in the future. Nonetheless, it is important to consider that these numbers are an average of all student responses, and there are likely individual differences here. Furthermore, it is important to consider that these numbers are self-reported by the students themselves. Therefore, they may not be entirely accurate. Some students may be understating their time commitment; however, some may overstate.

For the purpose of educational research, time commitment can be an interesting variable to use as an indicator for the quality of a student or a study program. Compared to GPA, which is an obvious alternative, time commitment can be more relevant for comparison between institutions and countries. Additionally, time commitment has in some cases been found to be a good predictor for academic performance, however, there are also studies suggesting the contrary (Nonis & Hudson, 2006; Plant, Ericsson, Hill, & Asberg, 2005; Schuman, Walsh, Olson, & Etheridge, 1985). Nonetheless, time commitment is an interesting variable to further investigate, and the authors of this paper plan on doing more research on the topic in the future.

5.5 Exploratory results

The correlation analysis of the different variables for the most part resulted in few significant results. However, the lack of correlation is also interesting because they can contradict common assumptions.

In this case, the lack of correlation between student satisfaction and number of students (both genders) is interesting because smaller classes of student are commonly assumed to create a better class environment.

6 CONCLUSIONS

This study has through a systematic review of Norwegian CS study programs attempted to identify some characteristics and important factors that impact student contentment and learning experiences. The study has found that Norwegian CS study programs vary in number of students, admission requirements, student satisfaction and time commitment. Concurrently, the gender unbalance is a consistent across all programs, and we found that there are similarities as to how the first year is designed. Further research is needed to deepen the understanding of what affects the students' contentment and time commitment. For example, additional research using individual data should be conducted. This research should focus on gender unbalance, factors impacting student satisfaction and further exploration of time commitment as an indicator for study quality. Additionally, there are variables not included in this study that could also be interesting to investigate, such as degree of completion and performance in the job marked.

7 REFERENCES

Adams, G. R., Berzonsky, M. D., & Keating, L. (2006). Psychosocial Resources in First-Year University Students: The Role of Identity Processes and Social Relationships. *Journal of Youth and Adolescence*, *35*(1), 78–88. https://doi.org/10.1007/s10964-005-9019-0

Blickle Gerhard. (1996). Personality traits, learning stratigies, and performance. *European Journal of Personality*, *10*(5), 337–352. https://doi.org/10.1002/(SICI)1099-0984(199612)10:5<337::AID-PER258>3.0.CO;2-7

Booth, A., Sutton, A., & Papaioannou, D. (2016). *Systematic Approaches to a Successful Literature Review*. SAGE.

DBH. (2018, April 11). Registrerte studenter - IE-fakultetet - kvinner. Retrieved April 11, 2018, from http://dbh.nsd.uib.no/statistikk/rapport.action?visningId=125&visKode=false&columns=arstall&index=1&formel=232&hier=insttype!9!instkode!9!fakkode!9!ufakkode!9!progkode&sti=&param=arstall%3D2018!8!2017!8!2016!8!2015!8!2014!8!2013!8!2012!8!2011!9!semester%3D3!9!dep_id%3D1!9!kategori%3DS

Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews* (Joint Technical Report). Keele, UK: Keele University.

Kunnskapsdepartementet. (2015). *Meld. St. 18 (2014-2015)* (Stortingsmelding). Retrieved from https://www.regjeringen.no/no/dokumenter/meld.-st.-18-2014-2015/id2402377/

Ministry of Local Government and Modernization. (2014). *Dimensjonering av avansert IKT-kompetanse* (Report). Retrieved from https://www.regjeringen.no/no/dokumenter/Dimensjonering-av-avansert-IKT-kompetanse/id762445/

National Academies of Sciences, E. (2017). *Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments*. https://doi.org/10.17226/24926

NOKUT. (2018a). *Studiebarometeret 2017: Tidsbruk på faglige aktiviteter og betalt arbeid*.

NOKUT. (2018b, February). Studiebarometeret. Retrieved April 6, 2018, from http://www.studiebarometeret.no/en/

Nonis, S. A., & Hudson, G. I. (2006). Academic Performance of College Students: Influence of Time Spent Studying and Working. *Journal of Education for Business*, *81*(3), 151–159. https://doi.org/10.3200/JOEB.81.3.151-159

Plant, E. A., Ericsson, K. A., Hill, L., & Asberg, K. (2005). Why study time does not predict grade point average across college students: Implications of deliberate practice for academic performance. *Contemporary Educational Psychology*, *30*(1), 96–116. https://doi.org/10.1016/j.cedpsych.2004.06.001

Ringdal, K. (2012). *Enhet og mangfold (Samfunnsvitenskapelig forskning og kvantitativ metode)* (3rd ed.). Bergen: Fagbokforlaget. Retrieved from http://www.fagbokforlaget.no/Enhet-og-mangfold3/I9788245013283

Samordna Opptak. (2018a, March 19). Samordna opptak. Retrieved March 19, 2018, from http://www.samordnaopptak.no/info/index.html

Samordna Opptak. (2018b, April 11). Poenggrenser - Samordna opptak. Retrieved April 11, 2018, from http://www.samordnaopptak.no/info/opptak/poenggrenser/index.html

Schuman, H., Walsh, E., Olson, C., & Etheridge, B. (1985). Effort and Reward: The Assumption That College Grades are Affected by Quantity of Study. *Social Forces*, *63*(4), 945–966. https://doi.org/10.2307/2578600

Tinto, V. (1975). Dropout from Higher Education: A Theoretical Synthesis of Recent Research. *Review of Educational Research*, *45*(1), 89–125. https://doi.org/10.3102/00346543045001089

Tinto, V. (1988). Stages of Student Departure: Reflections on the Longitudinal Character of Student Leaving. *The Journal of Higher Education*, *59*(4), 438–455. https://doi.org/10.1080/00221546.1988.11780199

APPENDIX A

Program	CS	NoS	%F	S	T	GPA
Institution abbreviation: English name of study program	*Amount CS in first year*	*Number of students*	*Amount female students*	*Satisfaction*	*Time commitment*	*Admission GPA for 2017**
HiØ: Bachelor in Computer Engineering	50 %	125	7 %	3,68	39	30
HiØ: Bachelor in Computer Sciences	100 %	144	11 %	4,01	43	30
HiØ: Bachelor in Digital Media	100 %	53	26 %	3,98	33	30
HiØ: Bachelor in Information Systems	100 %	129	12 %	3,70	37	42,8
HSN: Bachelor in Computer Engineering	50 %	129	12 %	3,70	33	30
HSN: Bachelor in IT and Information Systems	75 %	125	10 %	2,84	32	38,8
HSN: Bachelor in IT and Information Systems (Vestfold)	88 %	121	16 %	3,79	25	43,7
HSN: Bachelor of Computer Information Systems (Ringerike)	88 %	117	12 %	3,32	32	37
HSN: Bachelor of Engineering in Computer Engineering	50 %	96	7 %	3,85	44	30
HSN: Bachelor of Engineering, Computer Science and Industrial Automation	33 %	129	7 %	3,63	40	30
HVL: Computing	67 %	194	12 %	3,49	39	48,8
HVL: Information Technology	67 %	96	14 %	3,73	32	46,8
NORD: Bachelor in Games and Entertainment Technology	67 %	88	15 %	0,00	.	41,9
NORD: Bachelor's degree program in information systems	38 %	44	11 %	2,52	32	30
NTNU: Bachelor in Computer Engineering	67 %	232	13 %	3,68	36	55,6
NTNU: Bachelor in Information Technology with specialization in Network administration.	92 %	132	10 %	3,78	30	50,2
NTNU: Bachelor in IT-Operations and Information Security	83 %	147	12 %	3,54	38	44,2
NTNU: Bachelor in IT-supported Business Architecture	67 %	183	25 %	3,37	35	49
NTNU: Bachelor in Programming [Games | App]	67 %	96	5 %	3,74	37	45,3
NTNU: Bachelor of Engineering - Computer Science	50 %	148	5 %	3,41	24	43,3

Program	CS	NoS	%F	S	T	GPA
Institution abbreviation: English name of study program	*Amount CS in first year*	*Number of students*	*Amount female students*	*Satisfaction*	*Time commitment*	*Admission GPA for 2017**
NTNU: Bachelor of Engineering in Computer Science	67 %	140	6 %	3,96	35	47,1
NTNU: Communication Technology	63 %	229	39 %	3,78	38	57
NTNU: Computer Science (5-year)	50 %	692	20 %	3,74	36	58,5
NTNU: Engineering and ICT	25 %	257	34 %	3,77	36	56,2
NTNU: Informatics	63 %	481	16 %	3,73	32	53,2
OsloMet: Bachelor in Applied Computer Technology	100 %	218	20 %	3,71	28	51,1
OsloMet: Bachelor's Degree Programme in Information Technology	83 %	153	16 %	3,67	33	49,9
OsloMet: Bachelor's Degree Programme in Software Engineering	67 %	271	15 %	3,40	30	49
UiA: Bachelor's Programme in IT and Information Systems	83 %	216	11 %	3,85	44	47
UiA: Bachelor's Programme in IT and Information Systems	100 %	216	11 %	3,85	44	47
UiA: Computer Engineering, Bachelor's Programmme	50 %	261	10 %	3,81	32	30
UiA: Master's Programme in Information and Communication Technology	50 %	79	10 %	3,93	38	30
UiB: Bachelor Programme in Information Science	83 %	252	16 %	3,48	23	46
UiB: Bachelor's Programme in Bioinformatics	33 %	24	63 %	.	.	46,6
UiB: Bachelor's Programme in Computer Science	33 %	61	13 %	3,87	29	53,1
UiB: Bachelor's Programme in Computer Security	50 %	79	9 %	3,08	25	45,7
UiB: Bachelor's Programme in Computer Technology	50 %	175	9 %	3,87	35	52,2
UiB: Bachelor's Programme in Informatics-Mathematics-Economy	33 %	24	29 %	.	.	46
UiB: Bachelorprogram in Information- and Communication Technology	67 %	71	21 %	2,62	20	47,8
UiO: Digital economy and leadership	67 %	.	0 %	0,00	.	62,1
UiO: Informatics: design, use, interaction	100 %	302	42 %	3,83	31	53
UiO: Informatics: language and communication	83 %	65	34 %	3,83	32	51,1
UiO: Informatics: nanoelectronics and robotics	67 %	103	17 %	3,94	40	53,1
UiO: Informatics: programming and networks	67 %	341	16 %	3,95	34	53,1
UiS: Computer - Bachelor's degree programme in computer science	33 %	214	10 %	3,78	39	43,7
UiS: Computer Science - Master's Degree Programme	33 %	70	21 %	3,85	44	30
UiT: Bachelor of Science - Computer Science	50 %	134	15 %	3,61	37	30
UiT: Computer Science - bachelor	50 %	118	9 %	3,56	44	45,1
UiT: Computer Sciences - master	50 %	102	9 %	4,14	52	44,7
Westerdals: Bachelor - E-Business	88 %	110	15 %	4,09	40	30
Westerdals: Bachelor - Frontend and Mobile Development	88 %	53	8 %	.	.	30
Westerdals: Bachelor - Intelligent Systems	88 %	109	8 %	4,00	37	30
Westerdals: Bachelor - Interactive Design	88 %	89	54 %	3,78	33	30
Westerdals: Bachelor - Programming	88 %	106	8 %	3,99	41	30
Westerdals: Bachelor Game Design	63 %	56	16 %	3,53	33	30
Westerdals: Game Programming	88 %	53	2 %	.	.	30

Programs are listed in alphabetical order. Value of . indicates that there was no available data for that variable.

** Value of 30 indicates that all applicants were admitted.*

Paper 2

First Year Computing Study Behavior: Effects of Educational Design

Madeleine Lorås, Trond Aalberg

FIE 2020

Authors' contributions: Lorås led the research design, data collection, analysis, and was the main author. Aalberg provided general supervision of the research and the paper writing.

 118

First Year Computing Study Behavior: Effects of Educational Design

Madeleine Lorås
Department of Computer Science
Norwegian University og Science and Technology
Trondheim, Norway
madeleine.loras@ntnu.no

Trond Aalberg
Department of Computer Science
Norwegian University og Science and Technology
Trondheim, Norway
trond.aalberg@ntnu.no

***Abstract*— This full research paper presents a study exploring first year computing students' study behavior and the effects of educational design. Some research has indicated that the relationship between students' study behavior and their academic performance is as strong as the relationship to more common predictors such as past performance and test scores. However, knowledge about students' study behavior, how behavior develops and is influenced by program and course design, and consequently, the effect various design parameters have on learning is limited. This paper presents a model describing computing students' study behavior and how these are affected by the educational design. Through a mixed-method approach, a population of computing students was followed through their first year. Results from in-depth interviews with students throughout their first year found that the educational structure and organization of a study program conditions the students' study behavior. In order to further investigate these tendencies, two surveys (N=215) were conducted within the whole first-year student population at the beginning and end of the year. A significant difference found was in the use of surface and deep strategies at the beginning and end for the first year, indicating that students shift from deep to surface learning during the year. Even if students initially seek a deep content-driven approach to learning, the structure of the education and other organizational factors may be the cause of a more surface and task-focused approach towards the end of the first year. Students' study behavior is constrained by the educational design, which furthermore may lead to different learning outcomes than desired. Researching and developing learning goals, course content, lectures and assignments is one way to improve computing education; however, this research suggests that taking a comprehensive and integrated approach to educational design might also lead to improvements.**

Keywords—Study behavior, Study habits, Computing education, Engineering education, Educational design

I. Introduction

Designing education that is suitable for all students and fulfills every learning goal is a challenging task. Within computing education (CE), the enrollment numbers into higher education are increasing; however, there is a demand for even more computing students to graduate [1], [2]. More students accepted into a program also means a more diverse group of learners, and in recent years most higher education institutions have emphasized throughput as the main metric when measuring institutional performance. Together, this creates a demanding reality where educators are required to continuously develop the quality of education with increasing student numbers, as well as improve the throughput of graduates. Unfortunately, educators and higher education institutions have a limited room for action, as teaching and organizational resources are not increasing at the same pace.

This paper describes a study looking into computing students' study behavior. Students' study skills, habits and strategies are highly important for academic performance and throughput, which is significantly influenced by program and course design. This paper contributes a new perspective that can help solve major challenges in computing and engineering higher education.

When seeking to understand the academic success and failure of students in higher education, there are many stakeholders and various factors to consider. Previous research has indicated that there is a strong relationship between academic performance and study behavior [3], [4]. In their meta-analysis from 2008, Credé and Kuncel found that study skills and habits exhibit a strong relationship to performance, even as strong as more common predictors such as prior academic performance and admission test results [3]. In other words, the way students study is central to their learning.

Therefore, the work presented in this paper aims to increase the knowledge about computing study behavior and the interaction with educational design. Additionally, the first year of higher education is said to be formative for the student and crucial for retention [5]. Hence, the research inquiry is as follows:

- What characterizes computing students' study behavior during the first year?
- How is this behavior impacted by the educational design of the study program?

II. Study Behavior

How students' study and learn can be summarized as *study behavior* and has, over the years, been the focus of many research studies, although the terms and definitions described are often inconsistent. A review by Tressel, Lajoie and Duffy from 2019 addresses this fragmented domain and proposes a hierarchal study terminology based on research from the last decades [4]. They define study behavior as "any actions students make when preparing for, or taking part in, study-based activities." This definition is broad on purpose and is the base level of all study terms. Furthermore, the study process, skills, habits, strategies and tactics are terms placed hierarchically under behavior as described in Tab. I.

There are many ways to further view these terms, and for the purpose of this paper, it is useful to differentiate between internal and explicit study behavior. The internal study behaviors are the processes and strategies on the cognitive level and inherently influences the explicit behavior. Skills, habits and tactics are the specific intentions and actions the student takes when studying. This relation is illustrated in Fig. 1.

TABLE I DEFINITION OF STUDY TERMINOLOGY.
Based on Tressel et al. (p. 121)

Term	Definition
Behavior	Any actions students make when preparing for, or taking part in, study-based activities.
Process	The cognitive level of engagement with study tasks.
Skills	The students' level of ability to maintain and succeed in study tasks.
Habits	The consistency of study behavior, including the study environment. When, where and how much students study.
Strategies	The intentional behavior where a learner chooses how to study from a variety of study tactics while considering the demands of the task.
Tactics	The individual learning tools students use. E.g., Notetaking, highlighting, self-testing, etc.

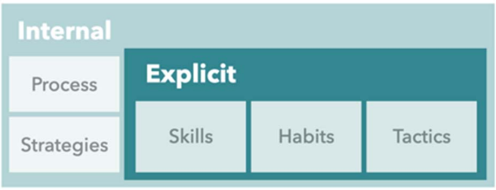


Fig. 1: Internal and explicit study behavior

When it comes to the internal aspect of study behavior the students approaches to learning (SAL) framework is an important theory developed by Marton and Säljö in 1976 and further developed by Biggs [6], [7]. According to SAL theory, students learning and studying process can be categorized into deep and surface approaches. The deep approach is an internally driven motivation and commitment to learning, where the intention to extract meaning produces active learning. Whereas the surface approach is externally driven, which concerns just coping with various tasks and is considered a much more restricted learning process. Most recently Biggs described this difference as the surface approach referring to "activities of an inappropriately low cognitive level, which yields fragmented outcomes that do not convey the meaning of the encounter" and the deep approach as "activities that are appropriate to handling the task so that an appropriate outcome is achieved" [8, p. 42]. Considering the explicit study behavior skills, habits and tactics, Credé and Kuncels work have been influential [3]. Their meta-analysis of study skill constructs is based on the study skills, habits and attitudes framework (SSHA). This framework also includes study attitudes, which refers to the students' mindset and motivation towards higher education and studying. Tressel et al. argue that attitudes are important to assess but should be placed under the broader umbrella of study skills. The remaining constructs, skills and habits, are related to the when, where and how students' study, and is similarly defined in Tressel et al.'s review.

III. Educational Design

In general, the design of a study program and the first year varies across universities; however, there are some commonalities. Regardless of organization, higher education can be viewed as three levels: program, course and student level. The program is designed with overall learning outcomes and goals for the students. A program consists of courses, which have more specific learning outcomes, learning activities, teaching staff and assessment methods. Lastly, there is the student level, which involves the students' study behavior and interaction with the other levels.

Furthermore, each level will have certain design parameters that constitute the educational design as a whole. As described further in Tab II, these parameters pose questions about certain design aspects educators must consider. For instance, how many courses there are in a semester, the use of assignments and assessment in a course, and if the course open to all students or reserved for one study program (open or closed enrollment). These parameters will affect the individual students and their behaviors, as well as the classes of students as a group.

TABLE II HIGHER EDUCATIONAL DESIGN AND PARAMETERS

Level	Description	Parameters
Program	Admission Program design Social, academic and physical learning environment	Prerequisites, enrollment structure Number of semesters Weight of a course (number of credits) Enrollment and admission regime Parallel vs. modular courses Campus layout
Course	Course structure Learning activities Educators Assessment	Open or closed enrollment Pedagogical design Number of lectures Number of assignments and/or projects Individual or group-based activities Type of assessment and exams Number of students
Student	Study behavior Demographics and background	The internal and explicit study behavior of the student, and the interaction with program and course design.

IV. Computing Education

When investigating the students' study behavior, it is important to discuss the context, which, in this case, is computing study programs in Norway. For the purpose of this paper, we consistently use the term computing, with the understanding that the term includes what in Norway is often categorized as ICT: computer science, informatics, information and computer technology.

On the program level, not much directly relevant research has been done in terms of educational design. However, one can argue that the research on pedagogy is interesting in this regard. In Ben-Ari's influential discussion of constructivism in computing education (CE), the author argues that the theory is highly applicable to CE, yet not satisfactory implemented [9]. Furthermore, research investigating constructive alignment is also relevant to the program level. Biggs defines constructive alignment as formulating learning goals and synchronizing this with constructivist-based learning and teaching activities and assessment tasks likely to lead to said learning goals [10]. On the course level, there are multiple empirical studies on everything from content and curriculum to use of technology and assessment, both in computing and STEM in general.

A. Study Behavior in Computing Education

The 2018 ITiCSE working group on introductory programming reported that research on student behaviors had seen an increase in focus on gathering and analyzing behavior

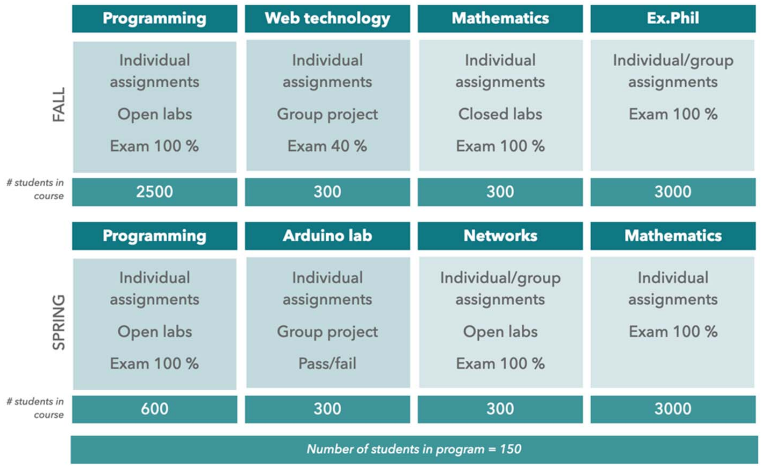


Fig. 2: Typical design of a first year computing education program in Norway.

data in order to learn about how students study and learn [11]. Furthermore, they report that predicting success, performance, identifying difficulties, encouraging change, designing interventions, and tools for these purposes to be the main value of such research to educators.

Tendencies very similar to the findings of Tressel and colleagues were observed in previous research on study behavior in CE [4]. There seem to be various perspectives and definitions being used, as well as many different research methods. Common for many studies is the data-driven approach [12],[13], meaning that behaviors and habits are defined around the data available, as compared to theoretical frameworks. As far as methodology, surveys and interviews are widely used. More recent studies have used log-file and submission data as well [12]–[14].

Many studies are focused on introductory-level courses [13], [15]–[17]. One common underlying motivation for these studies is to learn more about how computing students study and predicting performance. Previous programming experience and lecture attendance have been found to have a positive effect on exam performance while using the internet, non-lecturer instructors, working with others, and the use of tutorials and model solutions did not [15]. Furthermore, they found that classroom experience is no longer the central aspect of a student's learning behavior. Instead of lectures and teachers, students relied more on online resources and working independently [16]. More recent studies have compared behaviors of higher and lower performing students in an introductory computing course [13]. Among other factors, the results show that high performing students were better at soliciting help, seek out extra resources and take extensive course notes. In contrast, lower-performing students were more inclined to memorizing code, getting answers from others without understanding them and not continuing work on assignments post-deadline.

V. Methodology

This paper presents a mixed methods study aiming to explore computing students' study behavior and the impact of the educational design. Therefore, the study was set up with an exploratory sequential design [18]. Firstly, a qualitative interview study was done with a sample of students throughout their first year. Based on the findings from these interviews, a second quantitative survey study was done. After describing the context and participants of this design, the rest of the paper will be structured sequentially. First, the analysis and results from the interviews will be described and discussed, then the survey.

A. Context and Participants

Computing education (CE) at the university level in Norway is generally structured into two semesters. The fall semester lasts from August to December and the spring semester from January to mid-June. The semesters are structured into courses, usually three or four will run in parallel. Assessment is often based on a final exam, although more focus has been put on alternative and diverse assessment plans in recent years. As an example of a computing program in Norway, the structure and content of a typical computing program have been summarized in Fig 2.

The participants in this study all attended a program with a similar design. For the survey phase students from 11 different programs participated, and for the interview phase students from one of these programs were selected. Common for all these study programs is that all first year courses are mandatory and between 50-75% if the courses are in programming or computing of some sort. Generally, these courses are structured with weekly or biweekly assignments the students must complete, alone or in groups. The tasks do as a rule not count towards the final grade and are not considered forms of assessments. Instead, they are considered required work, which gives the students the qualification to take the final exam, which decides the grade. Furthermore, the number of students enrolled yearly into computing programs included in this study varies from 30-150, and the percentage of female students between 10-30% [19]. The students in these computing programs, often take courses with other computing and engineering students, increasing the total number of students in each course. For example, Fig. 2 depicts a program with 150 first year students, who in this instance take an introductory programming course with 2350 students from other programs.

Out of this student group, six students were recruited to participate in the interviews, all from the bachelor's in computing program exemplified in Fig. 2. These students agreed to meet the researcher through their whole program, or possibly follow up if they chose to switch programs or drop out. The students were recruited at a voluntary weekly study day. All attendees were invited, ten people signed up, and six were chosen on the basis of diversity and background. Out of the six interview participants, two were female, and one had a minority background. Additionally, two of the students had completed some other higher education study program before starting this one, two had done a gap year, and the remaining two started university straight from upper secondary school. Lastly, only two of the students had previous formal training in computing. When presenting the results, these details will not be linked to the various statements in order to preserve the participants' anonymity.

VI. Phase 1: Interviews

Interviews are considered a good method for gaining insight into people's attitudes, perceptions and experiences [20], [21]. As this study focused on exploring computing student's study behavior, it was essential to understand their experiences. Therefore, doing semi-structured, in-depth interviews were chosen as an approach. Three rounds of interviews were performed, one late in the first semester, one in the middle of the second, and a retrospective interview early in their third semester. This means each student was interviewed three times during their three semesters, making the total number of interviews conducted. Each interview lasted between 30-50 minutes, making the total interview time over 10 hours.

The participants consented to record the interviews, which were subsequently transcribed before analysis. The interviews were exploratory in nature but focused on certain topics. In the first interview, the focus was on previous knowledge, motivation and experiences with being a student so far. The second interview emphasized on study behavior and learning experiences, while the third was overall self-evaluation of the first year as a whole. All rounds of interviews were guided by an interview protocol; however, the researcher heavily followed the student's line of conversation. Additionally, the researcher used certain probes to make the participants comfortable and assured [21]. The researcher performing the interview had completed the study program in question and used this knowledge and experience to encourage the students to elaborate by sharing certain experiences.

A. Interview Analysis and Results

The interview transcriptions were analyzed with a grounded theory approach. The aim of grounded theory analysis is to reduce the data and extract theoretical ideas, explanations and understanding [21], [22]. In this case, the data was analyzed by coding in three phases, as described by Corbin and Strauss: open, axial and selective coding [22]. In open coding, all phrases and statements found interesting were initially coded, creating 36 very broad codes (e.g., study structure, study habits, learning environment, motivation, positive/negative learning experiences). In the next step, each code was inspected more closely and a set of 105 more nuanced codes emerged (e.g., factors of prioritizing work, strategies for getting unstuck, the social group as supportive, collaboration is motivating). In axial coding, these initial codes were printed and cut out, and then laid out on a big table using a constant comparative method [21]. By comparing all codes to each other, some overall categories and hierarchy emerged from the data. In the selective coding process, the research questions guided the process of identifying central themes or trends emerging from the data.

As far as the internal study behavior goes, the interview results showed how students prioritize, how they structure their study week and what underlies their study process. An example of how students talked about prioritizing is this student who described time and challenge:

> *Mostly I work on what deadline is coming up first. Either that or I work on the course, I understand the least.*

Furthermore, the students talked about how they studied, that is how they structured their independent work. It was common for all the students that the various aspects of the course design impacted their behavior. This quote describes how the student structured his/her work based on assignments:

> *It's much easier to study when I have to, rather than when I should. I have liked that about this semester. Having an assignment to do each week. It kind of forces you to study and having a study routine.*

Following these students through their first year, the learning activities provided in each course seemed to be a driving factor for the students' study behavior. As exemplified in these quotes, deadlines and assignments were fundamental to the structuring of students' study day. They also mentioned lectures and available support and resources in relation to finishing assignments. This student reflects on the benefit of morning lectures in this way:

> *Because then you get up in the morning and get to campus. And when you are there you're there, studying and working, when you're on campus anyway. So that is really just an advantage.*

When it comes to getting help, the students use a broad range of available learning recourses. Some students use the teaching assistans to get help on assignments, while others use their friends. An example of how the social and academic environment is important, is this quote:

> *I almost learn more thom my friends here. Because they just explain things easier.*

Additionally, the interview results indicated some interesting trends as to how their study behavior develops over the first year. The students all described decreased motivation and, in their own words, "worse" study habits in the second semester. They talk about taking shortcuts, impacts of social life and the increased workload as negative aspects of the second semester. They also express a motivation to change their habits and improve their study process. An example of this is a student's response when asked how the second semester as compared to the first:

> *There was something about being new. You were just so on all the time. But this semester, it's not the same.*

The final result of the coding process was the development of a model shown in Fig. 3, illustrating how the students' described their study behavior (priorities, strategies, habits, skills and motivation) and how they are constrained by the educational design, as well as how this might affect the learning outcomes.

B. *Model of Student Behavior and Educational Design*

The interview results indicated that the educational design of the first year on a program level had an impact on students' study behavior. The various aspects of a course, as well as the alignment between courses, seemed to outweigh the internal motivation or drive to learn when it came to structuring study behavior. Based on these findings, we propose a model of computing study behavior and educational design. This model illustrated in Fig. 3 describes how these elements interact and their possible impact on learning. On the one side there is the input the students bring with them, that is their behavior, here described further by prioritization, strategies, habits, skills and motivation. With this input, the educational structure and organization provide the conditions for the students' study behavior, i.e., acting as limits and constraints. The students will adapt their study behavior to fit these boundaries. Lastly, there is the outcome here described as what knowledge and skills learned.

The model describes the student perspective on and experience with the educational design. Considering the educational design parameters presented earlier, it seems like there are certain aspects students do not identify. Based on the interview results, students focus on the course design parameters, and in particular assignments and assessment. When it comes to the program level aspects, except for the social and academic learning environment, none of the parameters were mentioned by the students. Lastly, on the student level students describe their behavior and the interaction with course design parameters more often than the program parameters.

On another level, there is the educator's role. The educators have made design choices based on the parameters described in Tab. II, which will interact with the students' input, as will the planned and implemented teaching and learning activities. These will lead to the learning of skills and knowledge, which may or may not fulfill the actual planned and desired outcome. The interesting and important role of this model is how the students' input, interact with the educational design and whether or not this leads to the desired outcome. The planned and implemented teaching and learning activities may fit their learning goals; however, this model suggests that the students' priorities, strategies, habits, skills and attitudes may lead to different outcomes. In other words, if the educators' plan is based on students taking a deep approach in one course, but the students are limited by the educational design and chose a surface approach, do they learn the skills and knowledge they were supposed to?

Most educators would agree that deep learning, where the student understands the content and really learn the skills of the course, is the desired outcome [23]–[25]. However, these results have indicated that in this case, the structure and organization, together with the students' priorities and strategies, may not facilitate this. Additionally, these results indicate that the students' development over the first year is not desirable, which further suggests that there is something about the structure and organization of the education that influences them. The way the students use different words to describe their study process at the beginning and end of the year is striking. During the first interview, the students would consistently focus on the content of the courses and how interesting the various programming features were. In the second interview, on the other hand, the language used by the students was much more task-oriented. The students would consistently talk about assignments and exams instead of programming and computing constructs. This shift from a content-driven study behavior to a task-oriented one lead us

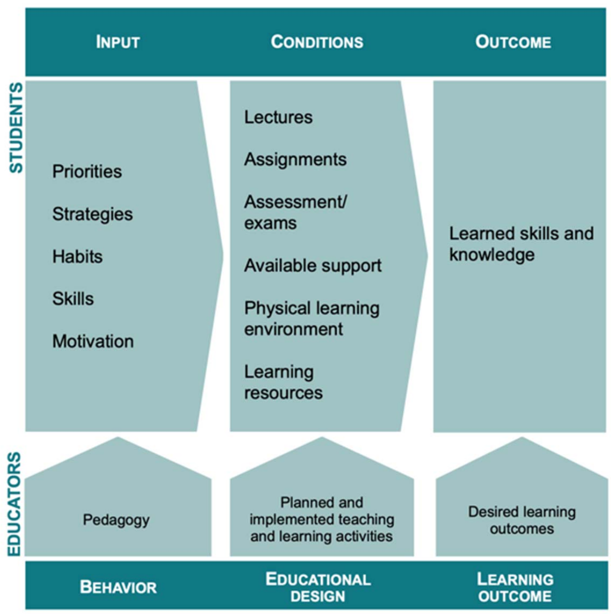


Fig. 3. Model of student behavior and educational design

to consider the possibility that students came into university with a deep approach to learning but were shifted to a surface approach in interaction with the educational organization and structure. Previous research on the SAL framework and the development over the first year has indicated that the assessment structure influences the students into a surface approach [26]. The interview results, on the other hand, suggests that incoming students were also affected by other educational factors. And that the development over the whole first year fosters this shift. To further investigate this, Phase 2 of this research was initiated.

VII. Phase 2: Survey

In addition to categorizing the characteristics of computing students' study behavior, the interviews indicated a change in study approach throughout the first year from deep to surface. The way the students changed the language when describing their study behavior from content-focused to assessment focused, indicated a switch from deep to surface strategy. This founded the motivation for the survey study. Furthermore, the survey was intended to test the hypothesis: *Computing students have a different study strategy at the end of the first year than they had in the beginning.*

In order to test this, we used the Biggs revised two-factor Study Process Questionnaire (SPQ), which can indicate whether a student has a deep or a surface approach to learning [27]. This questionnaire is commonly used to investigate students' internal study behavior, that is, the process and strategies [3], [4]. The SPQ was translated into Norwegian and sent out to the students at the beginning and end of the first year during the academic year 2018/19. All first-year students in computing programs at NTNU were invited to participate in an online questionnaire about expectations to university studies. The first iteration of this survey was sent out within two weeks of the first semester, and the second at the end of the year. Because of privacy issues, the survey did not include identifiers, so it was not possible to track the students on an individual level. However, the survey provides an overview of the student population since it was the same group of students who participated in both surveys.

A. Survey Analysis and Results

The first iteration of the survey was sent out to first year students in all NTNU computing study programs, a total of 695 students, and 215 students responded with consent. That leaves a respondent rate of 30% for the first iteration. For the second iteration, the study process questionnaire was part of a larger survey sent out to all students (in all years). Out of all the students, the number of students who responded that they were in the first year was only 96, although almost half of the respondents unfortunately did not answer this question. Therefore, the respondent rate for first year students in iteration two was 13%, while the overall respondent rate for the survey in total was 20%. For both iterations, the number of female respondents was around 30%.

The deep and surface scores were calculated following the revised two-factor method described in Biggs et al. [27]. When analyzing these results, the first step was to see if there seemed to be a difference from the beginning to the end of the semester. A Kernel density plot for respectively fall 2018 and spring 2019 was drawn using the statistical software Stata MP [28]. As seen in Fig. 4 there seems to be a visible shift. The surface approach scores seem to be the same for the fall and spring semester, whereas the deep approach scores have shifted towards the lower end of the scale.

In order to further test if the observed shift is an actual difference in study strategy, thus testing the hypothesis, the *two sample t-test* was used to evaluate the mean difference between the fall and spring scores [29]. Accordingly, the original hypothesis needed further specification:

Computing students have a different study strategy at the end of the first year than they had in the beginning.

- H1: There is a significant difference between the surface scores for the fall and spring semesters.
- H2: There is a significant difference between the deep scores for the fall and spring semesters.

B. Difference in Surface Approach

The students at the beginning of the year had a slightly higher surface score (*M= 23.5, SD=4.49*) than the end of the year (*M=22.5, SD=5.12*). The mean difference was, however, not significant within a 95% confidence interval, *t(243)=1.60, p=0.111, d=1.00*. When testing the assumptions for t-tests, it became clear that there were outliers in the data. The normality and homogeneity of variance, on the other hand, were within acceptable ranges [29]. After removing the outliers, the mean difference was significant, *t(241)=2.06, p=0.041, d=1.25.*

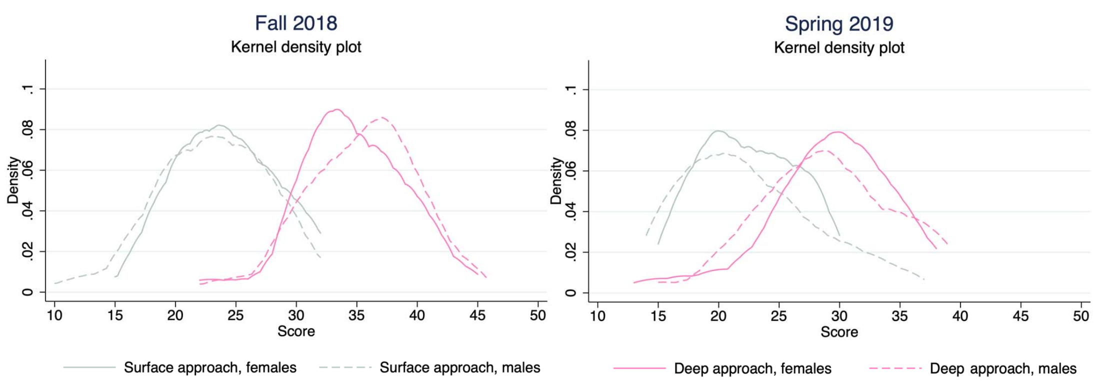


Fig. 4: Kernel density plot of deep and surface scores at the beginning (fall 2018) and end (spring 2019) of the year, divided by gender.

C. Difference in Deep Approach

The students at the beginning of the year had a considerably higher deep score (*M= 35.2, SD=4.53*) than the end of the year (*M=29.0, SD=5.47*). The mean difference was significant, *t(242)=9.54, p>0.001, d=6.16*. When testing the assumptions for t-tests, it became clear that there also were outliers in this data. The normality and homogeneity of variance, on the other hand, were again within acceptable ranges. After removing the outliers, the adjusted mean difference was still significant, *t(241)=9.16, p>0.001, d=5.68*.

D. From Deep to Surface

These results indicate that there is indeed a shift in the students' study approach at the beginning and end of the first year. However, analysis of both surface and deep scores indicate a lower score at the end of the year, which is difficult to interpret. As far as the hypothesis' goes, both H1 and H2 are confirmed. Firstly, there is a slight but significant difference in surface scores from the beginning to the end of the first year. Lastly, the change in deep scores was also significant, but considerably higher, by a factor of five.

VIII. Discussion

The research goal of this study was to characterize computing students' study behavior in the first year and investigate the impact of educational design. The model of study behavior and educational design presented in this paper characterizes computing students' study behavior in the context of educational organization and structure. Furthermore, the model highlights the aspects of educational design, which typically are developed and changed by different stakeholders. For example, the fact that there are certain aspects of the design, we as course teachers can and cannot change. Students and their input into this model are aspects we cannot change; however, the design parameters which frame the students' learning are changeable. And these are aspects that were found to highly affect and influence the students' study behavior and learning outcome.

Following the framework presented in Section II, the model includes most of the mentioned dimensions [4], [7]. Considering the internal study behavior, the model addresses prioritization and strategies, which are important constructs in the students' study process and strategy. The survey results confirm that students change their internal study behavior throughout the first year. Students start the first year with a deep approach where their study behavior is content-driven and end the year with a surface and task-focused behavior. Although, this change might be due to general study fatigue during the first year, there also seems to be reason to believe that the learning activites and program design are influential.

The explicit behaviors, habits and skills, thereunder motivation, are also evident. When asked about how they plan and implement their study week, they all based their independent study time on some organizational elements, such as lectures, assignments, collaboration, or teaching assistant availability, which is in line with previous research [15], [16]. It is evident that the students are influenced and constrained by the educational design of the courses. On the program level, it is interesting to see how the students manage their computing-courses relative to their other courses. They all discuss prioritizing they study activities based on computing relevance.

As far as educational design is concerned, the results indicate that the students' study behavior is influenced by the structure and organization of the education. In other words, educational design can be viewed as an independent variable when investigating the students' study process and behavior. On the other hand, factors such as previous experience, employability concerns, expectations and social learning environments might be influential factors as well.

A. Implications and Future Work

This study has found grounds to pursue the inclusion of educational design parameters in future research and practice. As previous research has shown, there are limitations in how much insight can be gained about how students' study when only considering specific activities. In order to fully understand these processes, there is a need to broaden the theoretical discussion to include study program design elements. The current study argues that design parameters should be viewed in a holistic manner, both in theory and practice. Some concrete examples extracted from the data are listed below:

- Courses should coordinate the use of assignments and projects so that the students keep a content-driven focus throughout the program. Four weekly assignments in parallel seem to foster a task-focused approach, leading the students to surface learning.
- The use of individual and group-based activities should be balanced throughout the program, both for social and academic reasons.
- The use of formal formative assessment should be increased in a manner that keeps students in a content-driven mindset.
- The access to help and support on a program level should be increased. This should include both course-specific topics and general study support in order to scaffold first year students' study behavior over time.
- The number of students should be considered in relation to the use of open or closed courses and labs. Students report that the sense of belonging is affected by the closeness to their peers, and educators should therefore support classes as a whole. Especially in larger institutions.

Based on the results presented in the current study, we have implemented some adjustments based on these parameters in our own study programs. The Informatics Study Day initiative has shown promising results [30].

B. Generalizability and Limitations

This study examined a specific institution with one student population. Other universities with different student groups will most certainly have different inputs, conditions and, consequently, different outcomes. Nevertheless, the model presented here can be used by all educators to design better and more aligned programs and courses. Lastly, the research methodology used in this study has some limitations. The study program examined, and the students who participated were from one institution and a relatively small non-random sample. The model will need to be further validated and expanded with research on other populations. Furthermore, the constraints of qualitative research are apparent in the sense of bias, however rigorous and systematic the data gathering, and analysis was performed. The survey and interview data provided source triangulation, and during analysis, the researcher used well established and validated techniques such as thematic coding [21].

IX. Conclusion

In this paper, the theoretical perspectives on computing students' study behavior in the first year of higher education have been explored. Through analyzing in-depth student interviews, a clear link was confirmed between study behavior and educational design. Computing students' priorities, strategies, habits, skills and motivation are constrained by the educational design, which may lead to different learning outcomes than desired. Furthermore, this study found that there is a significant shift between the beginning and end of the first year when it comes to internal study behavior. The students initially have a deep, content-driven approach to studying; however, they develop a surface and task-focused approach towards the end.

Researching and developing learning goals, course content, lectures and assignments is one way to improve computing education; however, this research suggests that taking a comprehensive and integrated approach to educational design might also lead to improvements. It is important to consider what kind of learners computing students become, as well as making sure they have the required content knowledge. The model presented in this paper outlines clearly where the room for action is for educators, and the design parameters provide a concrete starting point for educational change. Developing an educational design of the first year, which aligns the curriculum, courses and teaching in such a way that students become expert learners through effective study behavior may prove useful to later courses and employers.

Acknowledgements

The work presented in this paper was conducted at the Excited Centre for Excellence in IT Education. Excited receives public funding through DIKU, Norwegian Agency for International Cooperation and Quality Enhancement in Higher Education. We would like to especially acknowledge professor emeritus Tor Stålhane for valuable feedback.

References

[1] Ministry of Local Government and Modernization, "Dimensjonering av avansert IKT-kompetanse," Report, Jun. 2014. Accessed: Sep. 25, 2017. [Online].

[2] E. National Academies of Sciences, *Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments*. Washington DC: The National Academies Press, 2017.

[3] M. Credé and N. R. Kuncel, "Study Habits, Skills, and Attitudes: The Third Pillar Supporting Collegiate Academic Performance," *Perspect. Psychol. Sci.*, vol. 3, no. 6, pp. 425–453, Nov. 2008, doi: 10.1111/j.1745-6924.2008.00089.x.

[4] T. Tressel, S. P. Lajoie, and M. C. Duffy, "A guide for study terminology: Reviewing a fragmented domain.," *Can. Psychol. Can.*, vol. 60, no. 2, p. 115, 2019, doi: 10.1037/cap0000138.

[5] V. Tinto, "Dropout from Higher Education: A Theoretical Synthesis of Recent Research," *Rev. Educ. Res.*, vol. 45, no. 1, pp. 89–125, Mar. 1975, doi: 10.3102/00346543045001089.

[6] J. Biggs, "Individual differences in study processes and the Quality of Learning Outcomes," *High. Educ.*, vol. 8, no. 4, pp. 381–394, Jul. 1979, doi: 10.1007/BF01680526.

[7] F. Marton and R. Säljö, "On Qualitative Differences in Learning: I—Outcome and Process*," *Br. J. Educ. Psychol.*, vol. 46, no. 1, pp. 4–11, Feb. 1976, doi: 10.1111/j.2044-8279.1976.tb02980.x.

[8] J. Biggs, "What the student does: teaching for enhanced learning," *High. Educ. Res. Dev.*, vol. 31, no. 1, pp. 39–55, Feb. 2012, doi: 10.1080/07294360.2012.642839.

[9] M. Ben-Ari, "Constructivism in Computer Science Education," in *Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education*, New York, NY, USA, 1998, pp. 257–261, doi: 10.1145/273133.274308.

[10] J. Biggs, "Enhancing teaching through constructive alignment," *High. Educ.*, vol. 32, no. 3, pp. 347–364, Oct. 1996, doi: 10.1007/BF00138871.

[11] A. Luxton-Reilly *et al.*, "A Review of Introductory Programming Research 2003–2017," in *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, New York, NY, USA, 2018, pp. 342–343, doi: 10.1145/3197091.3205841.

[12] S. C. Goldstein, H. Zhang, M. Sakr, H. An, and C. Dashti, "Understanding How Work Habits Influence Student Performance," in *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, New York, NY, USA, 2019, pp. 154–160, doi: 10.1145/3304221.3319757.

[13] S. N. Liao *et al.*, "Behaviors of Higher and Lower Performing Students in CS1," in *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, New York, NY, USA, 2019, pp. 196–202, doi: 10.1145/3304221.3319740.

[14] J. B. Fenwick Jr., C. Norris, F. E. Barry, J. Rountree, C. J. Spicer, and S. D. Cheek, "Another Look at the Behaviors of Novice Programmers," in *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, New York, NY, USA, 2009, pp. 296–300, doi: 10.1145/1508865.1508973.

[15] D. Chinn, J. Sheard, A. Carbone, and M.-J. Laakso, "Study Habits of CS1 Students: What Do They Do Outside the Classroom?," in *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103*, Darlinghurst, Australia, Australia, 2010, pp. 53–62.

[16] J. Sheard, A. Carbone, D. Chinn, and M. Laakso, "Study Habits of CS 1 Students: What Do They Say They Do?," in *2013 Learning and Teaching in Computing and Engineering*, Mar. 2013, pp. 122–129, doi: 10.1109/LaTiCE.2013.46.

[17] S. Willman, R. Lindén, E. Kaila, T. Rajala, M.-J. Laakso, and T. Salakoski, “On study habits on an introductory course on programming,” *Comput. Sci. Educ.*, vol. 25, no. 3, pp. 276–291, Jul. 2015, doi: 10.1080/08993408.2015.1073829.
[18] J. W. Creswell and V. L. P. Clark, *Designing and Conducting Mixed Methods Research*, 3rd ed. SAGE, 2018.
[19] M. Lorås, G. Sindre, and T. Aalberg, “First year computer science education in Norway,” in *Proceedings from the annual NOKOBIT conference 2018*, 2018, vol. 26.
[20] L. Cohen, L. Manion, and K. Morrison, *Research Methods in Education*, 5th ed. London, UK: Routledge, 2002.
[21] C. Robson, *Real World Research*, 2nd ed. New Jersey: Wiley-Blackwell, 2002.
[22] J. Corbin and A. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 4th ed. Los Angeles: SAGE Publications, Inc, 2014.
[23] M. Barattucci, “Approach to Study as an Indicator of the Quality of Teaching and of Learning Environment: the contribution of John Biggs,” *J. E-Learn. Knowl. Soc.*, vol. 13, no. 2, May 2017.
[24] N. Entwistle, “Promoting deep learning through teaching and assessment: conceptual frameworks and educational contexts.,” p. 12, 2003.
[25] W. W. Lake and W. E. Boyd, “Is the university system in Australia producing deep thinkers?,” *Aust. Univ. Rev.*, vol. 57, no. 2, p. 54, 2015.
[26] J. Malm, M. Alveteg, and T. Roxå, “Are We Promoting a Surface Approach to Learning During the First Year of Engineering Educations?,” presented at the 3:e Utvecklingskonferensen för Sveriges ingenjörsutbildningar, 2011.
[27] J. Biggs, D. Kember, and D. Y. P. Leung, “The revised two-factor Study Process Questionnaire: R-SPQ-2F,” *Br. J. Educ. Psychol.*, vol. 71, no. 1, pp. 133–149, 2001, doi: 10.1348/000709901158433.
[28] StataCorp, *Stata: Release 16*. College Station, TX: StataCorp LLC., 2019.
[29] F. J. Gravetter and L. B. Wallnau, *Statistics for the Behavioral Sciences - Standalone Book*, 9th ed. Australia ; United States: Cengage Learning, 2013.
[30] M. Lorås and T. Aalberg, “Informatics Study Day: Helping CS Students Become Better Students,” in *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, New York, NY, USA, 2018, pp. 380–380, doi: 10.1145/3197091.3205826.

 128

Paper 3

Study Behavior in Computing Education - a Systematic Literature Review

Madeleine Lorås, Trond Aalberg, Hallvard Trætteberg, Guttorm Sindre

TOCE 2021

Authors' contributions: Lorås led the research design, data collection, analysis, and was the main author. Lorås, Aalberg, Trætteberg and Sindre contributed in the data extraction process. Aalberg and Sindre provided general supervision of the research and the paper writing.

130

Study Behavior in Computing Education—A Systematic Literature Review

MADELEINE LORÅS, GUTTORM SINDRE, HALLVARD TRÆTTEBERG, and TROND AALBERG, Norwegian University of Science and Technology, Norway

As the field of computing education grows and matures, it has become essential to unite computing education and higher education research. Educational research has highlighted that how students study is crucial to their learning progress, and study behaviors have been found to play an important role in students' academic success. This article presents the main results of a systematic literature review intended to determine what we know about the study behaviors of computing students and the role of educational design in shaping them. A taxonomy of study behaviors was developed and used to clarify and classify the definitions of study behavior, process, strategies, habits, and tactics as well as to identify their relationship to the educational context. The literature search resulted in 107 included papers, which were analyzed according to defined criteria and variables. The review of study behavior terminology found that the same terms are used to describe substantially different study behaviors, and the lack of standard terminology makes it difficult to compare findings from different papers. Furthermore, it was more common for papers to use study behaviors to explain other aspects of students rather than exploring and understanding them. Additionally, the results revealed a tendency to focus on specific educational contexts, predominantly introductory programming courses. Although computing education as a field is well equipped to expand the knowledge about both study behaviors and their connection to the educational context, the lack of common terminology and theories limits the impact. The taxonomy of study behaviors in computing education proposed in this article can contribute to contextualizing the research in such a way that researchers and educators across institutional borders can compare and utilize results. Last, the article outlines some areas for future research and recommendations for practice.

CCS Concepts: • **Social and professional topics → Computing education**; **Computer science education;**

Additional Key Words and Phrases: Computing education, study behavior, study process, study strategies, study habits, study tactics, educational context

ACM Reference format:
Madeleine Lorås, Guttorm Sindre, Hallvard Trætteberg, and Trond Aalberg. 2021. Study Behavior in Computing Education—A Systematic Literature Review. *ACM Trans. Comput. Educ.* 22, 1, Article 9 (October 2021), 40 pages.
https://doi.org/10.1145/3469129

The work in this article was conducted at Excited Centre for Excellence, publicly funded through DIKU.

Authors' address: M. Låras, G. Sindre, H. Tratteberg, and T. Aalberg, Norwegian University of Science and Technology, Trondheim, Norway; emails: madeleine.loras@ntnu.no, guttorm.sindre@ntnu.no, hal@ntnu.no, trond.aalberg@ntnu.no.

1946-6226/2021/10-ART9 $15.00
https://doi.org/10.1145/3469129

1 INTRODUCTION

Study behaviors have been found to be crucial to students' academic success [30]. Within computing education, we know that students exhibit many different behaviors when studying and learning computing concepts [12, 109, 164] and that differences between effective and ineffective students can often be explained by such behaviors [134]. Research on study behaviors in computing education has seen an increase in focus over recent years [100]. Specifically, researchers have focused on gathering and analyzing behavior data to identify difficulties, design interventions, encourage change, and predict success and performance. However, this previous work on computing students study behavior is fragmented. For example, many different terms are used to describe the same behaviors [130, 152]. There is also still a need for further research focused both on the behaviors and definitions in use and on the role of the educational context in computing education. Therefore, this article explores how the computing education research community has approached computing students study behavior.[1] More specifically, the research questions are as follows:

- RQ1: How are study behaviors defined in computing education research?
- RQ2: In what ways are study behaviors included in computing education research?
- RQ3: What is known about the role of educational context in shaping study behaviors in computing education?

To answer these questions, we performed an extensive systematic literature review of study behavior in computing education. To do so, we developed a taxonomy of study behaviors by combining research in higher education, psychology, and learning sciences. This work takes a broad perspective on study behaviors, including everything from cognitive levels of engagement to concrete tools students use, making the contribution of this article different than other reviews. Previous reviews within computing education have looked at specific aspects of students' behaviors, such as metacognition [130] or the role of behaviors in predicting performance [71]. This review reveals that the variety of terminology and infrequent use of theoretical definitions limit the value of the research when it comes to generalizing and transferring knowledge between educational contexts. Based on the results of this literature review, the taxonomy was updated to include the study behavior terms identified in computing education trough the analysis. This extended taxonomy provides a tool for classifying the behaviors present in computing education literature, and other researchers and educators can use it as a tool in the future.

The rest of this article is organized as follows: In Section 2, we present the taxonomy and definitions on which the analysis is based. Section 3 presents the methodology used for the literature review by describing how papers were selected and analyzed. In Sections 4, 5, and 6, we present the findings to the three research questions, respectively. Section 7 provides a discussion of these findings and their implications and outlines opportunities for future research. Finally, Section 8 summarizes and concludes the article.

2 THEORETICAL PERSPECTIVES ON STUDY BEHAVIOR

Study behavior has, over the years, been the focus of many research papers, although the terms and definitions described are often inconsistent [100, 152]. Tressel, Lajoie, and Duffys review from 2019 addresses this fragmented domain and proposes a hierarchical terminology based on research from recent decades [152]. They define study behavior as "any actions students make when preparing for, or taking part in, study-based activities" [152, p. 121]. This definition is intentionally broad

[1] To limit the confusion between the terms *study behavior* and *research study*, any references to study or studies in this article refers to aspects of study behaviors. Any references to research studies will use different terminology.

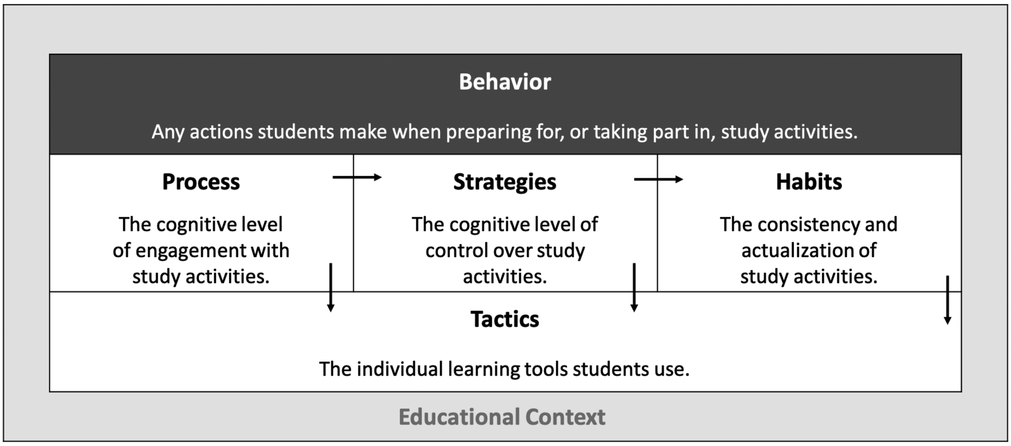


Fig. 1. Study behavior taxonomy: Definition and hierarchy of study behaviors.

and encapsulates all study terms. Based on this hierarchy, Figure 1 offers an outline of a taxonomy of study behaviors, which is the foundation of the analysis in this article.

A *taxonomy* is a system for naming and organizing things into groups that share similar qualities [35]. Self-regulation and metacognition, for example, share similar qualities and are therefore placed under study strategies [130]. Furthermore, the taxonomy as a tool serves two main purposes: It classifies the different constructs of study behavior and illustrates how they are related. The taxonomy above is based on the idea that any *study term* (i.e., self-regulation, time engagement, approach to learning) is placed in only one *behavior construct* (i.e., process, strategies, habits). Last, because the educational context construct is related to all behaviors, it is represented as such by being the background in Figure 1. Process and strategies primarily relate to the cognitive level. Habits and tactics primarily relate to the concrete what students do and use. These boundaries are not definite, and there are cognitive and concrete elements to all four constructs. Together, these four behavior constructs compose a more general construct of behavior, and study terms can be placed within such a construct. In addition to Tressel et al. [152], we draw from other research and theories within general education and computing education to further define the different terms.

The taxonomy as depicted in Figure 1 should be read from top left to bottom, with each row representing one level. The upper levels are grounded in the lower levels, and the behaviors on the same level inform each other, as illustrated by arrows. Thus, study process, strategies, and habits are closely connected and affect each other, and they act as drivers of the choice and use of tactics. For example, the case of a student working on a programming assignment illustrates how the levels of the taxonomy interact: First, the students level of engagement sets the foundation for this work. If the student takes a calculated approach, then the strategies she chooses will be guided by time management and self-regulation abilities with the goal of meeting a deadline, not necessarily understanding the concept. Furthermore, the habits in this case may be aimed toward limiting the total time engagement and perhaps not going to all lectures. Last, the tactics the student employs are guided by all these constructs, aiming for the deadline with strategic decisions, such as engaging in trial and error, high compilation frequency, and using the internet to quickly debug problems. In this hypothetical scenario, the student also navigates through the educational context, for example, attending organized teaching activities if needed, leaning on the social environment or utilizing labs or other study spaces. It is important to note here that this taxonomy does not state how a student studies, neither does it model ideal behaviors. This hypothetical is merely an example of how the constructs in the taxonomy work together. In the following subsections, we define terms and explore theories related to the behavior constructs process, strategies, habits, and tactics as well as the educational context.

2.1 Process

Study process in this article is defined as cognitive engagement with study activities, that is, students' internal approaches to studying and learning. It has been established that information processing consists of different levels of depth in cognitive processing [29]. There are two main directions within the many theoretical frameworks necessary to understand the study process described in behavior literature: student approaches to learning and learning styles.

The **student approaches to learning (SAL)** framework is a theory developed by Marton and in Säljö 1976 [106] and further developed by Biggs [14] and Entwistle and Ramsden [43]. According to SAL theory, students learning and studying process can be categorized into two categories: deep cognitive processing and surface cognitive processing. The deep approach is an internally driven motivation and commitment to learning, where the intention to extract meaning produces active learning. In contrast, the surface approach is externally driven and concerns coping with various tasks; it is considered a much more restricted learning process. More recently, Biggs described this difference as follows: The surface approach refers to activities of an inappropriately low cognitive level, which yields fragmented outcomes that do not convey the meaning of the encounter, and the deep approach refers to activities that are appropriate to handling the task so an appropriate outcome is achieved [15, p. 42]. Biggs and colleagues developed a questionnaire to measure whether students use a deep and surface approach [16], and it is commonly used to evaluate teaching initiatives and student learning approaches. The revised two-factor Study Process Questionnaire has been adapted and validated across countries and cultures (e.g., Reference [53]).

In addition to SAL theory, the notion of learning styles came from experiential learning theory and was first introduced by Kolb in the 1980s [87]. Experiential learning refers to the generalized differences in learning orientation based on the degree to which people emphasize the four modes of learning process [88, p. 76]. Many different frameworks for learning styles have been developed since then, but a common theme is describing learner characteristics in different dimensions [26]. On the topic of learning styles, it is important to address a substantial critique voiced over the years: the lack of empirical justification when matching instructional methods to the supposed learning styles of individual students [116]. Several reviews have found that there is inadequate evidence to justify incorporating learning style assessments into educational practices (see, for example, Pashler et al. [123] and Coffield [26]). Furthermore, learning styles have been criticized for the potentially harmful practice of diagnosing students [115] as well as for the commercial profits being made from the sale of tools and software [26]. Even though learning styles still seem to be in use in the educational system, many researchers view the framework as debunked [115].

In this review, we make a distinction between learning styles and the SAL framework; however, the latter has also received some skepticism [64, 133]. Whereas learning styles are criticized for the lack of empirical evidence, SAL theory has been miscited and misunderstood in many research papers [133]. Moreover, the perspective of SAL as a model, rather than a theory, has caused deep and surface approaches to learning to result in deep and surface learners [104, 133]. Defenders of SAL theory acknowledge limitations to how SAL should be used and emphasize room for further development and contextualization of the theory [26, 104]. Indeed, SAL theory does not aim to characterize a learner and is dependent on the context [104]. A student may adopt a deep approach in one context and a surface approach in another, depending on the characteristics of the context and the learners interpretation thereof [44]. We therefore argue that there is reason to distinguish between learning styles and student approaches to learning (keeping in mind that learning styles are criticized in the literature and that SAL theory should be viewed with an appropriate level of academic skepticism).

2.2 Strategies

Besides process, it is also important to understand strategies relating to studying. *Study strategies* are in this article defined as one's level of cognitive control over study activities. In this definition, we have combined some theoretical perspectives to clarify the terminology. First, this definition includes what Tressel et al. [152] define as skills and strategies because that definition is more in line with other definitions, such as Credé and Kuncel [30] and Prather et al. [130]. Second, differentiating between a skill and a strategy in practice was challenging and unnecessarily confusing. For example, the term self-regulation, which is the process of executing cognitive control during a task [131], could be considered both a skill and a strategy. To avoid the same terms being categorized into two behavior constructs, we combined the concepts of skills and strategies and used the term strategies to refer to both because the word skills has a very solid establishment within competency frameworks [52].

Within this definition of strategies fall the study terms metacognition and self-regulation, time management, motivation, and affective constructs. First, it is important to define and differentiate metacognition and self-regulation. Prather et al. [130] did a systematic review of metacognition and self-regulation in programming education in 2020, clarifying terms and measurements. They define metacognition as knowledge about one's own cognitive control, whereas self-regulation is the process of executing cognitive control [130, p. 3]. In other words, the difference lies in knowledge versus execution. It has also been pointed out that the environment plays an important role in self-regulation, whereas metacognition is focused on the mind of the individual [37]. Together, they constitute cognitive control, and they are closely connected [82]. Within cognitive control, time management is important and is an indicative measurement of self-regulation [165].

Last, there are the affective constructs [92], also referred to as non-cognitive factors [136]. Affective constructs are terms related to emotions, attitudes, feelings, and beliefs [152]. Examples of affective constructs common in the literature on study behaviors are epistemological beliefs [72], personality [129], confidence, attitudes [68], self-efficacy, and grit [38].

2.3 Habits

In addition to strategies, habits also play an important role in how study behaviors affect the success of computing students. Study habits is one of the most loosely defined terms in the literature [30, 152]. Tressel et al. [152] argue that *study habits* should be defined by the consistency of study behaviors, regularity in the use of study strategies, and the study environment. This definition means that study habits are informed by the study process and strategies but are related to explicit behaviors. In this article, study habits are defined as the consistency and actualization of study activities, which means that the interaction with the environment has been removed (see Section 2.5 for more).

An important aspect of study habits in our definition is that it is related to the activities students partake in when studying. Whereas process and strategies are related to purely cognitive processes, habits, and tactics are concrete. In a way, process and strategies can be seen as aspects of why and habits and tactics as what. Nevertheless, research on study habits commonly includes the ability to manage time [174]. We propose to differentiate time management and time engagement based on this distinction between why and what. In Credé and Kuncel [30]'s definition of study habits, they are related to the frequency of study sessions or time engagement, whereas time management is related to the planning and intention of time spent studying [165]. Therefore, time engagement is a study term within the construct of habits, and time management is a term within the construct of strategies.

2.4 Tactics

Last, tactics are defined as the individual learning tools a student uses during their studying [152, p. 120]. Examples of study tactics are note-taking, self-testing and viewing videos. Within computing education, there are many specific tactics, such as debugging and use of **integrated development environment tools (IDEs)** [154]. The use of tactics is informed by the study process, strategies, and habits. Research on tactics has revealed that students success is related to the awareness of using certain tactics and the breadth of tactics used [57]. Like habits, tactics are aspects of what students actually do; however, the choice and use of specific tactics are connected to cognitive levels of engagement and control. When differentiating between habits and tactics, one can consider their origin and consistency. Habits are consistent routines that students have acquired, while tactics are concrete elements informed by the habits as well as by process and strategy. Furthermore, tactics are often discipline-specific and include tools unique for computing, such as debugging and pair programming.

2.5 The Educational Context

Students study behaviors happen in close relation to the educational context, here defined as the organized teaching and learning activities, learning environment, and curriculum [17]. Tressel et al. [152] consider students' interaction with the learning environment to be part of a student's study habits; however, we find it more logical to view the educational context as a factor affecting all study behaviors. The educational context involves physical, cultural, and social aspects and is inherently linked to cognitive and concrete aspects of study behaviors [9, 36]. Bandura's theory of reciprocal determinism states that a person's behavior influences and is influenced by personal factors and the social environment [9]. Teaching activities are the organized events involving an educator, such as lectures, seminars, and assessments. Learning activities reference the organized activities students are expected to do independently, such as assignments, projects, quizzes, and general studying. The learning environment includes diverse physical locations, social contexts, and cultures in which students learn, including their interactions with teaching and learning activities and content and curriculum. How a student studies is influenced not just by the educational context but also by the student's perceptions of the learning environment [97]. Thus, a student's ability to navigate within the educational context is a central aspect of study behavior, linked to process [17], strategies [37], and habits [152].

In this section, we outlined the theoretical perspectives and definitions underpinning this literature review. The taxonomy in Figure 1 outlines the constructs and terms within research on study behavior from general educational domains. After presenting the methodology in Section 3, we will present the results of how study behaviors are defined and used within the computing education context.

3 METHODOLOGY

A **systematic literature review (SLR)** must follow well-defined protocols, guidelines, and academic norms. The current research is positioned at the intersection between higher education research and computing education. Within the computing and computing education fields, it is common to follow Kitchenham's procedures for performing systematic reviews, made to "introduce the concept of rigorous reviews of current empirical evidence to the software engineering community" [86, p. 1]. Within higher education research, there are several similar procedural guidelines. Bearman et al. [10] reviewed the use of systematic literature reviews in the field and outlined several common types. The current SLR is based on the Kitchenham procedure, which largely overlaps with what Bearman et al. refer to as the "Campbell-Cochrane systematic review." Common for both is the transparent and systematic nature of the search procedure, data extraction, and assessment, which is described for the current SLR in this section [10, 86].

3.1 Systematic Review Planning

To the authors' knowledge, no previous work has produced a systematic and comprehensive review of the existing published work on study behaviors in computing education. Thus, this article systematizes and summarizes the empirical work in the field and provides researchers and educators with insights for moving forward.

3.1.1 Search Strings. As described above, the various uses of terminology in the domain of study behaviors make it difficult to synthesize and compare results of various studies. In addition, the identification of relevant literature also becomes difficult in this regard. For this systematic review, we kept the definition of study behavior as broad as possible to identify these discrepancies and to resolve them. Therefore, the search terms used for study behavior include all terms in the hierarchy of Tressel et al. [152], namely, *study behavior, process, skills, habits, strategies,* and *tactics*.

To limit the search to computing education, we again ran into a definition problem, since computing education is denoted by a variety of terms throughout the world. In response to this problem, we chose to include the terms used in the 2005 Joint Task Force Computing Curricula [50], including the following: *computer science, computer engineering, information systems, information technology, software engineering,* and *computing*. Last, we limited the search to include education, specifically higher education. By using the search terms AND and OR, we created the following search string (in italics):

- Study behavior: *"(("study behavior" OR "study process" OR "study skills" OR "study habits" OR "study strategies" OR "study tactics" OR learning behavior OR studying)*
- Computing: *AND ("computer science" OR engineering OR programming OR cs OR CS OR computing OR ICT)*
- Education: *AND (education OR "higher education"))"*

The search terms were prototyped in a trial search [86, 100], confirming that the search string was reliable. We also learned that inclusion decisions based only on abstracts were not going to be possible, so the review process was adjusted to include full-text reviews.

3.1.2 Search Strategy and Selection Criteria. To find papers relevant for the review, we decided to include peer-reviewed empirical papers written in English that addressed study behaviors within higher computing education. Initially, the authors considered four inclusion criteria and four exclusion criteria to select papers for further analysis, as shown in Table 1. Next, we continued the selection process according to the set of seven quality criteria shown in Table 2. These quality criteria were informed by the Critical Appraisal Skills Programme [39, 83], which specifies the rigor, credibility, and relevance that need to be considered when evaluating the quality of papers.

The search for literature was done in several databases, using the "search within anything" function. First, we searched in the IEEE and ACM digital libraries, because they cover many of the most relevant conferences and journals in computing education research. In addition, we searched the more general libraries of Scopus, Web of Science, and Engineering Village to cover more literature. Table 3 shows that the initial search from these databases yielded 1,701 results, including duplicates. Searches in the Springer, ERIC, Elsevier, and SAGE databases were also performed; however, the results from these were either too large (n > 10, 000) or too broad (top listed papers were on

Table 1. Inclusion/Exclusion Criteria

Inclusion criteria	Exclusion criteria
The research was done within computing education or with a majority of computing students.	The paper is not a research study or peer-reviewed paper (e.g., extended abstracts, posters, reviews, blogs).
The research was done in higher education.	The paper is not written in English.
The research includes aspects of study behavior.	The paper is not accessible via university subscriptions.
The research is empirical.	The paper is under four pages.

Table 2. Quality Criteria

1. Does the paper address the research problem?
2. Is there a clear statement of the aims of the research?
3. Was the research design appropriate to determine the aims of the research?
4. Does the paper clearly determine the research methods (subjects, instruments, data collection, data analysis)?
5. Was the data analysis sufficiently rigorous?
6. Is there a clear statement of findings?
7. Is the paper of value for research or practice?

Table 3. Search Results by Source

Database	Initial extraction
ACM Digital Library	644
Engineering Village	589
IEEE Xplore	107
Scopus	217
Web of Science	145
Total	1,701

irrelevant topics). Upon inspection, there seemed to be a significant overlap in relevant papers between these databases and the ones included in this review (ACM, IEEE, EV, Scopus, and WoS).

3.2 Systematic Review Execution

The whole process of searching for, including, and excluding papers is illustrated in Figure 2. The first step was gathering papers from the various databases, as listed in Table 3. The next step involved removing duplicates and non-relevant item types, such as posters, books, and patents. With the remaining 1,301 papers, a read-through of titles and publication names was done to remove obviously irrelevant papers (step 3). In this phase, papers in unrelated fields, such as medicine and agriculture, were removed as well as blogs and posters that had, for some reason, survived step 2. Because of the broad search terms, a substantial number of titles were removed in this phase, resulting in 904 papers for abstract review. Next, a read-through of abstracts—and full text if needed—was done using the inclusion criteria presented in Table 1 (step 4). We evaluated papers in the following way:

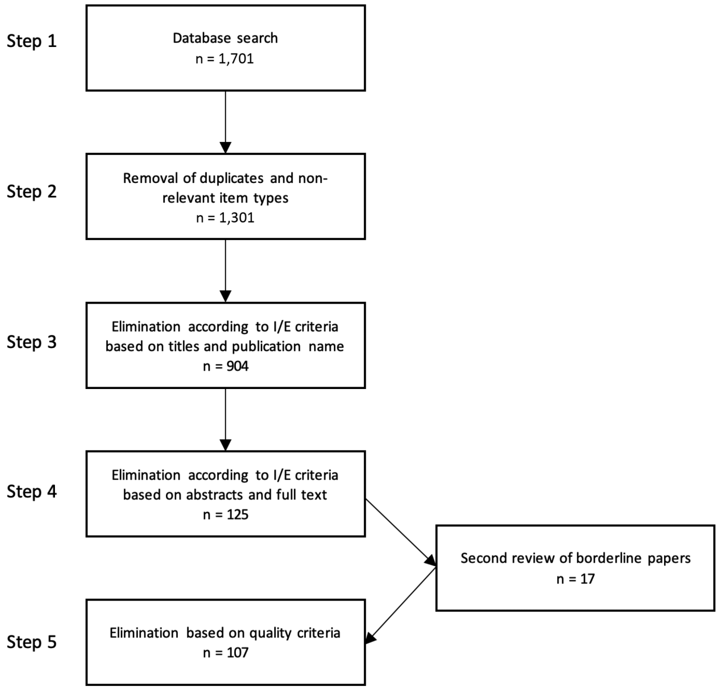


Fig. 2. Overview of search and selection process.

- Does the abstract reveal that the paper should be excluded? For example, this step excluded papers relating to the K-13 level, a different field (mathematics, physics), and papers not focused on behavior.
- If the abstract was inconclusive, then the full text was investigated. For example, the paper was excluded if it was about behavior but made no reference to context, field, or level of education.
- If the abstract was conclusive, then a full-text check was still performed to ensure the page number and language. For example, some papers were about study behavior in computing education at the university level; however, it was initially not clear what type of publication it was.

During abstract review, 723 papers were excluded, 125 were included, and 56 were labeled as borderline. A second review was performed on the borderline papers, which resulted in 17 new inclusions. Until this point, the first author had performed the search and selection process alone, but for steps 4 and 5, we had a second author review the papers. One author also did a second review of all borderline papers. To evaluate the quality criteria, all papers were reviewed by the first author as well as one of the other authors. Finally, we ended with 107 papers for data extraction and analysis, as listed in Table 11 (Appendix A).

Most of the papers included were published in peer-reviewed conferences (74%). ACM and IEEE channels were most common; however, there were also some learning technology and general education venues present. There has been a rise in the number of publications on these topics since the first paper in 1994, with a significant jump in the mid 2010s and with 63% of the included papers being published after 2015. The papers originate from all parts of the world; however, many papers referenced research done on the American continent (n = 49), and most of these were from the US or Canada. Otherwise, 27 papers originated from Europe, 12 from Asia, 12 from Australia or New Zealand, and 2 from Africa. Last, there were 5 multinational papers, ranging from two countries included to 10.

3.2.1 Data Extraction and Analysis. Data extraction was done by coding each paper according to nine variables [86]. The results of this coding process were then further analyzed to answer the research questions. Table 4 describes these variables and how they address the research questions. Some variables were accompanied by predefined categories, and some were based on noting or copying excerpts from the texts. For these open categories, we made sure only to extract data that was stated in the paper. For example, when extracting data on teaching implications, we only noted the actual implications mentioned in the paper, not what our opinion on potential implications was. A full overview of extracted data can be found at https://doi.org/10.18710/JQX7NW.

The first author coded all the papers, while the remaining authors coded a set each, providing double coverage of all the papers. The authors paired up to review their data extraction, identify differences, and agree on the final version. In instances where there was disagreement between these two authors, a consensus was reached by discussion. Certain factual fields were checked against the paper, while more subjective fields, such as study behavior, were merged in a way to include the most details.

The analysis was performed using non-statistical methods following the nature of the variables. Where needed, we categorized and counted the extracted data. For example, behaviors were categorized and grouped following the taxonomy presented in Figure 1. In the following sections, we detail this analysis, summarize the results, and describe the findings for each research question.

4 DEFINING STUDY BEHAVIORS IN COMPUTING EDUCATION (RQ1)

This section describes the results relating to the first research question: How are study behaviors defined in computing education research? When extracting the study behavior aspects of the selected papers, we placed the study terms used in the papers into behavioral constructs following the taxonomy presented in Figure 1. When analyzing the data further, we combined the research goals, data collection methods, description of behavior, main results, and implications to determine what behavioral constructs were discussed and how they were defined.

4.1 The Study Behaviors Identified

After extracting the various study terms from the selected papers, we mapped them into constructs according to the taxonomy. For example, papers that referenced deep and surface approaches to learning were placed under "process." Many papers, though, used terminology that was inconsistent with the definitions presented in Section 2. For example, "study habits" was used to describe many behavioral constructs that would be placed under tactics or skills according to our taxonomy. In one source, the term "learning habit" is used to describe time spent on assignments and the number of submissions, posts, and videos watched in an online learning system [66]. In this case, one could argue that study time, or time engagement in a study activity, should be categorized as a habit; however, the use of videos and posts would be considered a study tactic. In Hedin and Kann [70], the focus was on study skills, listed as preparing before lectures, smart note-taking, repetition, planning the upcoming week, maintaining a study diary, reading the course literature in three steps, and not procrastinating. However, most of these constructs are tactics, except for planning and procrastination, which are terms under "strategies." In other words, a central finding is that terminology use is inconsistent. The same terms are used to describe different aspects of studying. In the following sections, we review the findings for each of the taxonomy constructs.

Table 4. Description of Data Extraction Variables and the Connection to Research Questions (RQs)

Variable	**Description**	**Categories**	**RQs**
Research/educational goal	What was the goal of the research? In what way (if any) is the research related to performance and/or learning outcome?	Write down	1/2
Research questions	What were the research questions/hypothesis?	Research questions Hypotheses Lessons learned	2
Data collection	Type of data source/collection methods	Survey Questionnaire Validated questionnaire Log-data Submission data Interviews Focus groups Exam results/grades Other: write down	1
Behavior	What aspects of study behaviors were reported on, and how are they measured?	Write down	1/2
Main results	What were the main results?	Write down	1
Teaching implications	What were the teaching implications (if any)?	Write down	3
Sample population	What level was the research done in?	Introductory level Undergraduate level Graduate level All levels Other: write down	3
Educational context	What was the education context for this research?	Campus Online Blended Mixed (students from both)	3
Pedagogical context	What was the pedagogical context for this research	Traditional Peer Instruction Flipped MOOC Other: write down	3

4.1.1 Process. While the term "study process" refers to the level of cognitive engagement in study activities, it is also commonly used to describe the different stages and events in studying [11, 141]. In total, 24 of the papers included aspects relating to the study process, referencing the **student approaches to learning (SAL)** framework or the learning styles framework, as listed in Table 5. Within the SAL framework, deep, surface, strategic, and achieving dimensions are in use, but the Biggs's Study Process Questionnaire (with only the deep/surface dimensions) is the most common. Within learning styles, we found examples of Felder's dimensions (active/reflective,

Table 5. Overview of Papers Referencing the Study Process

Process	Papers	Count
Student approaches to learning	[5, 24, 49, 56, 62, 74, 93, 96, 99, 101, 108, 114, 117, 118, 126, 144, 155, 168, 172, 173]	20
Learning styles	[22, 23, 34, 112]	4

Table 6. Overview of Papers Referencing Study Strategies

Strategies	Papers	Count
Affective constructs	[24, 34, 54, 62, 63, 65, 68, 85, 91, 103, 122, 124, 136, 143, 151]	15
Time management	[2, 6–8, 40, 45, 51, 59, 70, 81, 95, 105, 161, 166]	14
Strategies	[1, 28, 42, 48, 63, 77, 90, 95, 146, 149, 159, 160]	12
Self-regulation	[5, 25, 46, 61, 78–81, 122, 169]	10
Motivation	[1, 61, 65, 68, 125, 166, 172, 173]	8
Metacognition	[28, 32, 48, 69, 75, 81, 125, 151]	8
Programming Strategies	[33]	1

sensing/intuitive, visual/verbal, sequential/global) [22, 23] and Kolb's learning cycle (concrete experience, reflective observation, abstract conceptualization, active experimentation) [22, 23, 34, 112]. In reference to the substantial criticism of learning styles described above, it is important to note that the four papers referencing learning styles were published between 1999 and 2009, indicating that learning styles are no longer a part of computing education research literature.

4.1.2 Strategies. In total, 68 references were made to study strategies in the selected papers, as further specified in Table 6. Some papers referenced several aspects of strategies and therefore appear more than once. Furthermore, some papers only referenced strategies in a general way—for example, describing the application of tactics [160] or cognitive routines [48]. Several papers used the term "strategy" but were referring to the study process [172, 173]. One paper talked about programming strategies, referring to specific planning strategies related to programming problems, such as "finding an average through several sub-algorithmic plans such as a triangular swap" [33].

A large number of the referenced strategies were related to metacognition and self-regulation, but as Prahter et al. [130] established, it can be challenging to distinguish between these terms. To differentiate and specify the terminology landscape, we chose to keep the underlying terms visible in Table 6. The seven papers that referenced metacognition generally used the term to describe monitoring [69] or reflecting [28] on one's own study strategies, or those papers used the umbrella term "metacognitive factors" [32]. Within self-regulation, we found the terms "organization," "direction," and "time management." Within time management, two papers referenced pacing study activities as a specific management aspect [155, 161]. Furthermore, three papers explored the starting time of assignments as tasks, both discussing starting early [2, 45] or late [59]. Start and finish times, which are closely linked to procrastination, were the focus of seven papers [8, 40, 51, 70, 81, 95, 105].

Last, we grouped personality, epistemological beliefs, attitudes, motivation, grit, and confidence into affective constructs [70, 152], also referred to as non-cognitive factors [136, 143]. There seems to be slight disagreement regarding whether these terms are aspects of metacognition or whether they should be viewed independently. For example, motivation and epistemological beliefs can be

Table 7. Overview of Papers Referencing Study Habits

Habits	Papers	Count
Time engagement	[11, 24, 27, 41, 42, 46, 51, 58, 65, 66, 73, 75, 76, 84, 89, 94, 108, 121, 122, 125, 127, 139, 141, 142, 154, 156–158, 162, 164, 171]	31
Habits	[4, 20, 24, 27, 31, 42, 45, 46, 66, 70, 80, 81, 91, 120, 138–140, 167]	18
Attendance	[1, 19, 24, 31, 89, 108, 121, 169]	8
Programming habits	[2, 154]	2
Life	[158]	1
Social networks	[59]	1

found under self-regulation and metacognition in Prather et al. [130]. However, for the purpose of this mapping, there seems to be an agreement in the definitions that these are all aspects of cognitive control. Affective constructs were often one of several aspects being researched or used to explain differences in performance. For example, Haungs et al. [68] describe a course development where motivation and confidence were two of several variables investigated to improve success and retention. A different example is Tolhurst [151], who specifically investigated the effects of a course revision on epistemological beliefs.

4.1.3 Habits. An overview of study habits identified in the included papers can be viewed in Table 7. In the review of the included papers, it was challenging at times to classify the reported behaviors as habits, since the authors often referred to what we have defined as strategies. We, therefore, made a distinction between intention and action when determining if a reported behavior should be considered a strategy or habit. Whereas strategies refer to cognitive control (i.e., planning, monitoring, and intention), habits depict what students actually do. In the article by Foo and Ng [49, p.2], study habits are defined as "the behaviors associated with studying (excluding methods used to learn or utilize academic material) such as time management and anxiety reduction," a definition that is more in line with the cognitive perspective of study strategies. An illustrative example of this distinction is the difference between time management (strategy) and time engagement (habit). *Time management* refers to the planning and intention of studying, often relating to when students study. *Time engagement* [89], however, refers to when the students did study and how much—for example, how much time students spent on an activity [24, 41, 46, 108, 139, 157], time spent in a system [142], time spent coding [70], or time spent before or after a class [171]. Similarly, attendance is a study term concerned with what a student has actually done and was the focus of eight papers [1, 19, 24, 31, 89, 108, 121, 169]. Some papers also focused on change in habits over time [46, 70] or the effect of an intervention such as an academic enhancement program [42], or supplemental instruction [45, 81].

A common theme in the papers on study habits was the discussion of good and bad behaviors. In some papers, habits were referenced as "good" or "bad" [2, 20, 27]. However, some papers also referenced "habits leading to success" [24] or "harmful habits" [8]. Not all papers were systematic in describing what good and bad habits are, which is arguably a relative concept. Some papers define bad habits by looking at how they relate to performance [27, 41, 121] or predict success [4, 45].

Two papers referenced programming habits specifically, with one relating to how novice programmers write code [154] and the other focusing on time spent programming [2]. In that latter paper, Allevato and Edwards [2] used time spent programming, among other variables, when evaluating the effects of extra credit on procrastination behavior. Only one paper specifically mentions social aspects of study habits and views participation in social networks as a habit [59].

Table 8. Overview of Papers Referencing Study TactiTcs

Tactics	Papers	Count
Techniques	[7, 8, 48, 51, 54, 56, 70, 76, 99, 113, 120, 139, 160, 162]	14
Resources	[47, 48, 54, 66, 99, 103, 120, 127, 156, 167]	10
Social	[19, 63, 66, 73, 96, 141, 142, 157, 162]	9
Trying	[7, 8, 20, 42, 48, 58, 141, 142, 162]	9
Preparations	[70, 111, 160, 167, 171]	5
Coding	[45, 47, 59, 154, 170]	6
Help	[69, 95, 99, 114, 160]	5

4.1.4 Tactics. In total, there were 57 references to tactics in the included papers, with several papers mentioning more than one tactic. When distinguishing a habit from a tactic, we considered the origin and consistency of the behavior. For example, attendance is considered a habit but taking notes a tactic. We further grouped the various tactics into seven categories, as illustrated in Table 8. For the previous constructs, the categorization was based on theoretical concepts, but for tactics, we found it more useful to create new groups. First, we made a distinction here between using various *resources*, such as videos [48, 66, 103, 127, 167], books [99, 167], and hints [47], and *techniques*, such as memorization [6, 139] and note-taking [70, 160]. Furthermore, the category of *trying* includes tactics related to attempting assignments [48, 142], solving many problems [7, 42], and retaking quizzes [20, 162], often tracked with log-file data. In contrast to most of the other behavior levels *social* interactions [19, 66, 73, 96, 141, 142, 157] and collaboration [63, 162] are two frequently mentioned tactics. The *help* category includes asking questions [99, 160] and help seeking behavior [69, 95], and the *preparation* category refers to preparing for lectures [167], tests [160], and classes [70, 111, 171]. Last, the *coding* category relates to specific tactics used when programming, such as using auto-complete [154], compilation frequency [45, 47, 154], debugging, and use of version control systems [170]. Vihavainen et al. [154] for example, looked at how novices tackle their first lines of code in an IDE and found that students tend toward three tactics: writing code from left to right, using auto-complete, and copying and pasting.

4.2 Theoretical Frameworks Used

In addition to categorizing the behavior terms and mapping them into the proposed taxonomy, it is also interesting to note where the definitions in the selected papers came from. Fewer than a third of the papers were grounded in established theoretical frameworks (n = 32). The most common framework used was Biggs's study process (n = 11). Some papers also relied on a validated questionnaire used in defining behaviors; however, the framework behind the questionnaire was not necessarily explored beyond the results (n = 14). In total, 15 papers reported their results by using a validated questionnaire within the learning and behavior domain. In addition, a substantial number of papers proposed their own definitions for what qualifies as a study behavior (n = 30) or based their definition on the data (n = 24). For example, based on log data from a MOOC platform, Sheshadri et al. [142] looked at study habits via time engagement, defined as "study sessions as consecutive sequences of study actions that occur between breaks for food or sleep." Similarly, one paper defined study habits as time spent in the system, number of submissions, and number of posts and videos watched [66]. In general, time management and engagement were often used as indicators of strategies and habits. Last, seven papers did not reference any definitions. For

Table 9. Overview of How Papers Used Study Behaviors

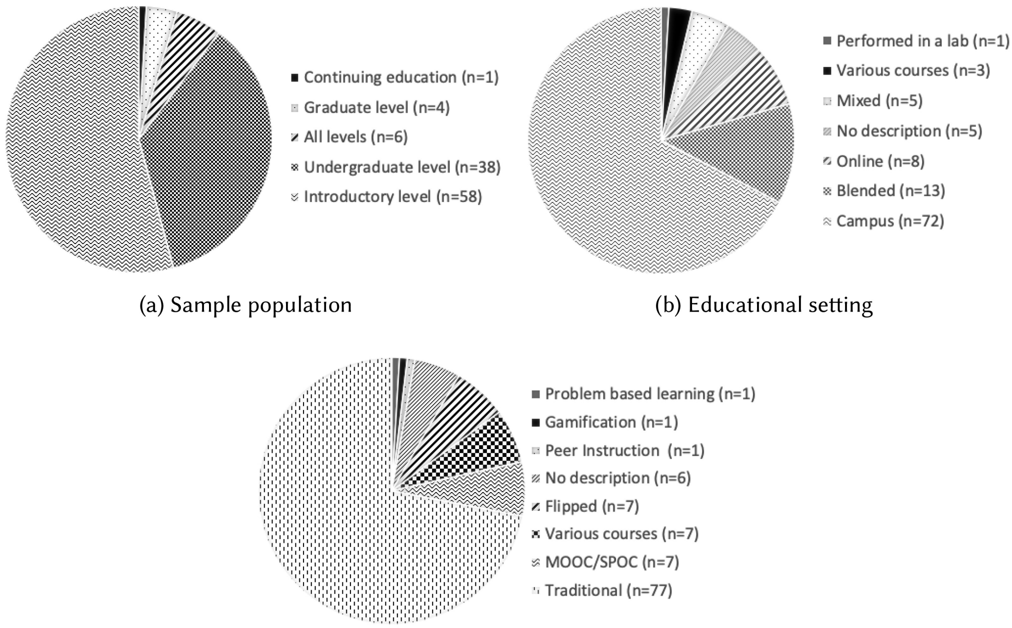
Decrease/reduce	Papers	Count
Dropout	[1, 5, 11, 74, 89, 125, 169]	7
Failure rates	[45, 80, 160]	3
Procrastination	[40, 62]	2
Bad behavior	[2]	1
Improve/enhance/increase		
Learning	[28, 73, 76, 78, 79, 81, 90, 93, 95, 96, 105, 114, 127, 139, 140, 155, 164, 166, 167, 170]	19
Study behavior	[8, 42, 49, 56, 121, 122, 146, 151]	8
Performance	[6, 7, 31, 32, 41, 162, 172, 173]	8
Retention	[19, 70, 85, 138, 159, 161]	6
Engagement	[48, 61, 84, 118]	4
Experience	[4]	1
Online learning	[79]	1
Programming skills	[33, 63, 111]	3
Learn about/understand/identify		
Study behaviors	[20, 22–25, 46, 59, 99, 117, 120, 126, 141, 149, 168, 171]	12
Learning	[22, 23, 126]	3
Online learning	[51, 101]	2
Programming learning	[65]	1
Predict		
Performance	[34, 58, 68, 75, 94, 136, 142–144, 157, 158]	11
Identifying students at risk	[47, 69, 77, 154]	4
Various		
Improving a tool/system	[27, 54, 91, 108, 113, 156]	6
Culture/gender diversity	[103, 147]	2
Transition to university	[66, 112]	2
Supporting teachers	[124]	1

example, Carpenter and McCusker [20] mention retaking quizzes as a way to reinforce good habits but do not elaborate further.

5 THE ROLE OF STUDY BEHAVIORS IN COMPUTING EDUCATION (RQ2)

This section describes the results relating to the second research question: In what ways are study behaviors included in computing education research? In this analysis, we used the variables of research/educational goal, data collection, and study behaviors. By investigating the goal of the various papers, we found why study behaviors were used as well as how they were used. Inspecting the research/educational goal, we found that most papers had one of four goals: (1) decrease or reduce undesired outcomes; (2) improve, enhance, or increase desired results; (3) learn more about, understand, or identify something; or (4) predict behaviors or events. These goals are illustrated in Table 9.

A majority of the selected papers used different study behavior constructs to explain other aspects of education, such as performance, drop-out, or prediction (n = 72). For example Benda et al.



(a) Sample population

(b) Educational setting

(c) Pedagogical context

Fig. 3. Summary of educational context parameters.

[11] investigated why online computing students drop out, using time engagement as an explanatory variable. Similarly, Chinn et al. [24] focused on identifying study habits that lead to success. Several papers related behavior to performance, such as Höök and Eckerdal [76], who investigated habits, and Hedin and Kann [70], who looked at strategies and tactics. Common for most of these examples, and most of the explanatory papers in general, is that the behaviors were one variable of many in the analysis or discussion sections [4, 46, 80, 138, 139].

A minority of the included papers viewed study behavior as the dependent variable, where the goal was to explore these behaviors (n = 35). In these exploratory papers, it was common to investigate how various interventions affected certain behaviors, to model behaviors for use in online learning tools or the study process across student groups [49]. Sheard et al. [140] adopted a holistic focus on study habits by exploring where, when, how, and with whom computing students studied, but such an approach was less common. Regarding the inclusion of behaviors in the analysis, it can be concluded that using behaviors as an explanatory factor is more prevalent than exploring behaviors. Furthermore, there is a focus on improving learning by decreasing or increasing various behaviors; however, the definition of "better" is somewhat unclear.

6 EDUCATIONAL CONTEXT AND STUDY BEHAVIORS (RQ3)

This section describes the results of the third research question: What is known about the role of educational context in shaping study behaviors? For this analysis, we used sample population, educational context variables, and pedagogical context variables. When considering the educational context, it is valuable first to examine the sample population, and in this case, the population's level of education. A majority of the papers used students at the introductory level (n = 58) – that is, first-year courses (CS0, 1, and 2). A somewhat typical example is the paper by Gomes et al. [61], who investigated connections between study strategies and performance in an introductory programming course. A different example is the multi-national investigation of Simon et al. [144] into cognitive, behavioral, and attitudinal factors that influence entry-level student's success in learning programming. Furthermore, Figure 3(a) shows that 38 papers included students at the undergraduate level; 4, students at the graduate level; and 6, students at various levels. Last, 1 paper presented research done on students enrolled in continuing education [11].

When it comes to the educational setting in which the research was performed, the majority of papers described a campus-based environment (n = 72). Some were done in a blended environment (n = 13) or fully online (n = 8). Only one paper presented research performed in a laboratory [127], indicating that behaviors are mainly researched in a natural setting. Additionally, some papers described mixed environments, where some students attended on campus simultaneously with students online (n = 5). For example, Petersen et al. [125] investigated reasons for dropping out of a multi-campus CS1 course with students at different campuses and online. As many as seven papers did not describe the setting in which the research was performed, and two studies were done in multiple courses where the educational setting was not described. For example, Halde et al. [65] used machine learning to investigate the impact of study strategies and habits on performance for students across the computing department. These findings are summarized in Figure 3(b).

Figure 3(c) summarizes the pedagogical context for the included papers. The predominant pedagogical context identified in the selected papers was a traditional design (n = 77), meaning that lectures and labs were primary components. While the exact learning design of these courses may have had significant variation in how lectures and labs were conducted and whether labs counted toward the grade, all pedagogical contexts with a heavy focus on lectures and labs were coded as "traditional" unless the paper described alternative pedagogical approaches that positioned it in another category. For example, Manley and Urness [103] compared the use of video lectures to in-person lectures in a course with quizzes and lab exercises. Some papers described MOOC and SPOC contexts (n = 7), and some described the program level or included several courses, making the pedagogy difficult to describe (n = 7). Flipped classroom designs were the focus of seven papers, such as Lin and Wu [96], who explored social interactions in a flipped classroom setting. In seven papers, the pedagogical context was not described; however, that omission was often because the focus on the paper was on specific tools [156], techniques [144], or teacher perspective [124]. Last, a few papers examined specific pedagogical contexts, such as Ma's [101] investigation of students" approaches to learning in problem-based learning.

The findings on the relationship between study behaviors and educational context in the included papers are somewhat ambiguous. The learning activities and interventions proved difficult to categorize, because the various papers had different goals and focuses. The main observation is that most papers examined general study behaviors, sometimes with a specific intervention, but often without one. It can be concluded, however, that introductory-level education is most prevalent, as is traditional pedagogy in campus-based environments. Concurrently, it was observed that there are discrepancies in the level of detail in the descriptions of educational and pedagogical contexts, making it hard to make further inferences. The next step is to further solidify the connection between various behaviors and the specific educational design parameters. Table 10 lists some proposed relations between the study behavior terms and educational design parameters, including references to example papers found in the current review.

7 DISCUSSION

In this section, we discuss the results, identify contributions, and present some observations and recommendations that follow from our review. We take this opportunity to summarize the important findings for each of the research questions and discuss the relation between them, building on the theoretical perspectives in Section 2 and extending the taxonomy of study behaviors.

Table 10. Potential Link between Study Behaviors and Educational Design Parameters

Behavior	Educational design parameters	Potential impact factors and examples
Process	Program, semester, and course design	The study process is hard to influence; however, research has found that approaches to learning do develop over time, suggesting that the educational design parameters have an effect [98]. The number of courses per semester, parallel versus modular approaches, weight and alignment between courses are some aspects to consider [125, 151].
	Learning activities and assessment	The holistic design of each year, the combination of courses and teaching and learning activities play a role [84].
Strategies	Learning outcome goals	Study strategies are also challenging to influence through educational design. However, including learning goals directed toward developing metacognitive skills in addition to content knowledge might support students in this regard.
	Specific training	Offering courses and training targeted toward the development of study strategies is one potential impact factor (e.g., programs integrating courses and academic-enhancement programs [32, 42, 70, 81]).
Habits	Scheduling of organized activities	The scheduling of organized activities can provide useful scaffolding for the development of study habits [80, 84].
	Mandatoryness/participation	The implementation of mandatory participation is a tool educators can especially use to influence habits. However, one should be mindful of the holistic design and ensure variation and balance [164].
Tactics	Learning activities and assignments	When designing learning activities and assessment, one can consider what tactics students might need to master to broaden their studying toolkit [111, 162].
	IDE and technology choices	Similarly, regarding choosing IDEs and technologies for use in computing courses, there is room for broadening the students' abilities (e.g., use of version control systems, web-based platforms, and professional IDEs [154]).

7.1 Defining Study Behaviors in Computing Education (RQ1)

The investigation into how study behaviors are defined in computing education revealed two main findings. First, the review found that the same terms are used to describe substantially different study behaviors and that the lack of standard terminology makes it difficult to compare findings from different papers. This finding is in line with research from other disciplines on the fragmented domain of study behavior definitions and terminology [152]. Educators and researchers should be mindful of this lack of unity and provide clear definitions in future research papers [130]. Second, these definitions are mainly based on data or self-described characterizations. Of all the papers, 75% did not define their terminology clearly, or they used self-defined terms where more established definitions were already available. The use of and development of domain-specific theories and models is an area where computing education research can grow. The work by Prather et al. [130] is a good example of a systematic contribution to bridging the gap between theories on cognitive control and programming education. This review found that the use of theoretical frameworks was often limited to the inclusion of a questionnaire or used as an explanatory element in the computing education field.

In support of this future work, we expanded the taxonomy in Figure 1 to include the study terms identified in the reviewed papers. This extended taxonomy is depicted in Figure 4. In the following, we further discuss the definitions and grouping of the included study behavior terms with regard to the perspectives in Section 2.

- **Process:** For the process behavior construct, two study terms were identified: SAL framework and learning styles. We included learning styles in the taxonomy, because it does not aim to model or moderate anything; however, we urge researchers and educators to be aware of the substantial critique of learning styles [115]. With regard to the SAL framework, we found that deep/surface approaches to learning was a commonly used variable; however, SAL theory is not often discussed. Questions for further exploration include what deep and surface approaches to learning mean in computing education and what insights they can give computing educators about the quality of learning [26, 104].
- **Strategies:** Strategies were defined using many different study behavior terms, and in the extended taxonomy, we include metacognition, self-regulation, time management, and affective constructs. Time management was the most referenced concrete aspect, perhaps because it is somewhat easily measurable. Affective constructs and motivation were also common terms, indicating that many papers attempted to include more personal aspects. We also revealed attempts to differentiate general strategies and programming strategies, which could be an avenue to pursue further.
- **Habits:** Within habits, we include time engagement, attendance, social networks, and balancing student life. The two latter terms were only referenced in one paper; however, social aspects [128] and balancing life [119] are important aspects of studying. Additionally, we found that the habits construct was the most loosely defined study behavior construct, often referring to strategies, specifically time management. Similar to strategies, we found specific mentions of programming habits. Further research could explore the notion that computing requires specific study strategies and habits.
- **Tactics:** For tactics, we include the seven groups of individual learning tools identified in this review. We believe there are more tools but hope the categorization may be valid nevertheless. It is within this construct that we identified the most computing-specific terms, grouped under "coding." However, the tools of social connections and "trying" are also linked to many pedagogical approaches in computing education, such as project and team-based learning [18] and pair programming [163].

To summarize, the contribution of the findings related to the first research question is the extended taxonomy of study behaviors in computing education (Figure 4). The taxonomy is based on theoretical definitions but takes into account many data-driven approaches. Similar to the review

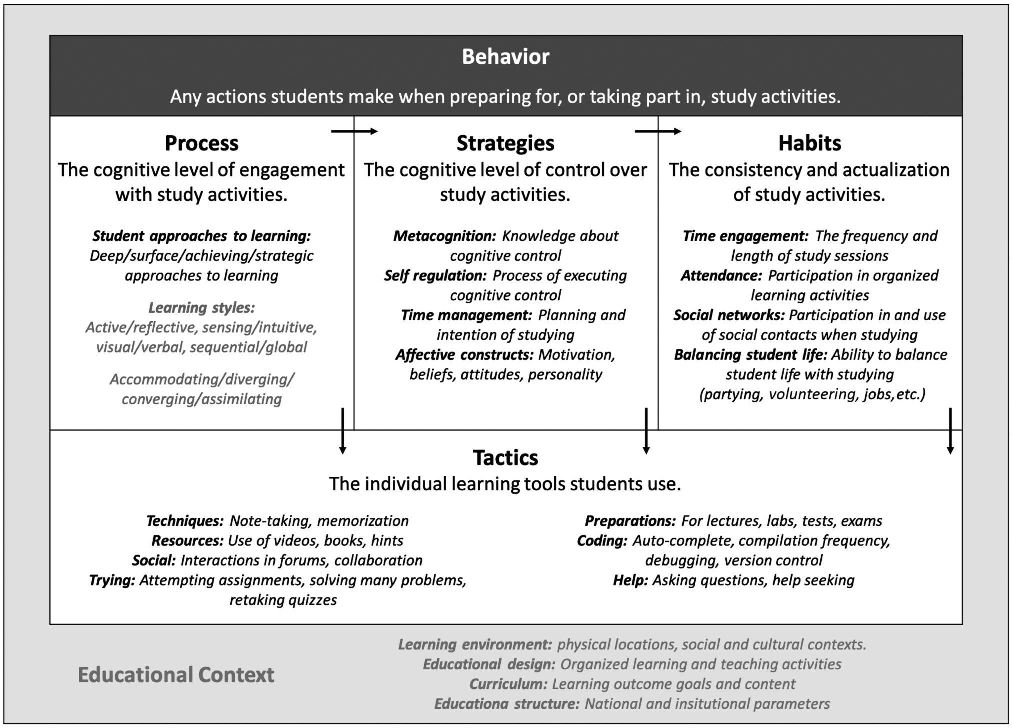


Fig. 4. Extended taxonomy of study behaviors for computing education.

by Szabo et al. [148] of learning theories in computing education, this article provides a synthesized overview and associated exemplars to improve the understanding of study behaviors, including how they relate to the educational context. Future research and practice can use this framework to identify terms when designing research projects or educational innovations, and it can serve as a tool for understanding and interpreting published research. In the discussion of the remaining research questions, these connections are explored in more detail.

7.2 The Role of Study Behaviors in Computing Education (RQ2)

The investigation into what ways study behaviors were included found that most papers used study behaviors to explain other student-related aspects, such as academic performance, engagement and dropout. Consequently, a minority of the included papers explored study behaviors. Considering the prevalence of inconsistent terminology, it is challenging to infer any trends or conclusions as to the role of specific study-behavior constructs. Reviewing Table 9, it is apparent that there were substantially more efforts published aiming to improve, enhance, or increase positive aspects of studying than to decrease or reduce negative aspects. Furthermore, it was not uncommon to read about "good" and "bad" study strategies, habits, and tactics in papers about different educational designs and innovations. To our knowledge, there is no established consensus distilling good, successful study behaviors. As several researchers have pointed out, we must be wary of developing "folk conclusions," whereby certain hypotheses are widely accepted as truths despite lacking empirical verification [67, 132]. Although all educators may have an idea of what good and bad behaviors are, such a vague and coarse categorization is not helpful for research. Furthermore, determining whether a behavior is good, successful, positive, or improved relative to a previous behavior depends on one's perspective. Indeed, the assessment of a behavior depends on whether the goal is for students to perform well on tests, learn the content, have a positive experience or hand in assignments on time. This review provides examples of all such perspectives as well as contradictory results. For instance, procrastination behavior is generally seen as a "bad strategy" [40, 62]. However, Goldstein et al. [60] found that procrastination does not necessarily decrease performance; it is the consistency of behaviors that matter. A student who usually starts assignments late may perform at the same level as one who usually starts early, but when an early starter starts late, the performance declines.

We believe it will be challenging to conclusively define good or bad computing study behaviors, even with more research on the topic. Perhaps a better approach for researchers and educators going forward is to focus on how knowledge about how students do study can help educators support students in developing effective study behaviors. All such discussion should of course keep in mind that the student is a complex being and that there is major individual variation between students.

In addition, while this review provides many examples of using study behaviors to explain the "quantity of learning," another perspective to explore is the quality of learning [44]. To do this, the computing education research community needs to place additional focus on exploring study behaviors. Only one-third of the reviewed papers aimed to identify study behavior, and only a few included perspectives across courses. Supporting students in their ability to learn how to learn is a potential next step for the computing education community. A holistic approach to student learning—considering more aspects of study behavior and educational context together—can be one important step. An additional avenue to pursue is including aspects of study behavior as indicators of success, broadening the perspective of academic success beyond grades and test results.

Another important finding in this review is that study time is a common variable to evaluate study behavior. Together, time management and engagement were by far the most common study terms, used in 44 of the included papers. However, study time is a debated metric. Some papers report that time spent studying can predict performance when seen in relation to other variables, such as previous experience and the learning environment [107, 119, 128, 137, 145]. These papers all emphasize the context, and that study time alone does not seem to be a good indicator of performance. Similar concerns have been raised about the quality of study time data as well as defining what it means to study computing specifically [135]. Moreover, the unresolved question of what study time data can tell us is supported by the current review—namely, that most behaviors are defined based on data and not on theoretical or established definitions. For example, it is interesting to consider what we can learn from timestamp data. As established, there is a theoretical difference between time management and time engagement, where the former is an aspect of cognitive control and the latter an actualization of said control, and timestamp data alone provides limited insight into the cognitive perspective.

The contribution of this section is an overview of where the focus in the field has been. Study behaviors in computing education have mainly played a supporting role in the investigation of academic performance [71, 100]. There are opportunities to improve our understanding of student learning by expanding the role of study behaviors in research and practice. In this work, the taxonomy can play an important role in setting the boundaries for coherently defining study behaviors across the community. Accordingly, we emphasize that this taxonomy is not a model for students' behaviors, only a road-map to understand them.

7.3 Educational Context and Study Behaviors (RQ3)

Regarding the third research question, which explores the relation between educational context and study behaviors, the main finding is related to what was present in the reviewed literature and what was missing. We found that research on study behaviors in computing education has been overwhelmingly focused at the undergraduate level, with a specific focus on introductory programming courses. The research has also mainly looked at study behavior under "traditional" teaching approaches in on-campus settings—that is, courses with a typical weekly progression of lectures and assignments. The prevalence of traditional designs is not surprising, considering this has been the dominant approach to educational delivery [100], but with increasing variation in approaches and settings, there is a need for more research on the alternatives. For example, it would be interesting to see more research on graduate-level study behavior and comparative approaches investigating whether there is a difference between levels and why. As most of the reviewed papers only provide snapshots of students' behavior in one particular course, it would be particularly interesting to see longitudinal approaches following groups of students throughout their studies to investigate how their study behaviors and awareness thereof develop with increased study experience. Latitudinal approaches (comparing behavior in several courses taken by the same group of students) could also be of interest to see the extent to which they adopt different behaviors in different courses and why. In these broader research approaches, we could also further explore the role of informal learning [13] and social interactions [9], two areas that have been largely overlooked in the research. Such research might also help illuminate some of the relationships between educational contexts and study behavior, which are currently unclear.

An important factor not present in most of the published work is the institutional structure, social context, and cultural context surrounding education. One concrete example is the age of the students and their level of independence. In Nordic countries, students enter higher education at the age of 19, while in the US, they may be 17 years old. When discussing study behavior, there is a large difference between 19 and 17, and when further considering the difference in the level of independence for these students, this divide increases. These social factors play an important role for students in their learning [150]; however, such factors are not present in discussions on computing students' study behavior. In the detailed taxonomy in Figure 4, we include balancing student life, where, for example, the presence of part-time jobs is a factor. Only one paper in the review included such an aspect of student life outside of academics [158]. To be able to account for such differences, there is a need to adopt a standard for including and describing educational design parameters at an established level of detail. These are variables outside of educators and researchers' control; however, we argue that they should be a factor considered when interpreting results or designing interventions.

This third research question makes the valuable contribution of revealing the importance of educational context. In the taxonomy, this emphasis is illustrated by adding educational context as an encompassing construct with specific terms. Altering the educational context can change the quality of student learning [110], and some concrete examples of the relation between different educational contexts and the study behavior constructs are summarized in Table 10. The educational contexts present in the published works range from very large classes to small student groups in online, blended, and on-campus settings. Many of the included papers, though, lacked descriptive detail about the educational context.

7.4 Implications

For educators, the value of this review lies mainly in the collection and mapping of research on study behaviors in computing education. The fragmented domain limits our ability to draw conclusions or make recommendations for educators to best support effective study behaviors. We have found some examples of how explicitly teaching students about study behavior, such as time management and planning, results in increased performance and experience [32, 42, 70, 81]. Furthermore, there does seem to be room for action when it comes to designing and structuring courses and programs to support students as they learn to learn, as explored in Table 10. Finally, we hope that the extended taxonomy presented in Figure 4 can serve as a guide for educators seeking to understand how students do (or do not) study.

For researchers, we outline the domain of study behaviors and identified some areas for future research. In addition, we propose a taxonomy for study behavior constructs and terms in computing education, which can be used to inform future research in the field. As with any proposed theoretical or conceptual frameworks, we expect it will need further development and validation [102]. Based on this review, we would like to summarize some recommendations for future researchers:

- Provide clear definitions of the study behavior constructs being researched.
- Avoid turning to self-defined definitions where theoretical frameworks are available.
- Refrain from making assumptions about what behaviors are good or bad.
- Be specific when describing the educational context of the research.

In addition, there are some gaps in the research and possible future research questions:

- Exploring the computing discipline specifically: Are computing students different than others? Do computing topics imply or rely on specific study behaviors? Are the methods and variables used to research study behaviors in computing appropriate and accurate?
- Expanding the perspectives on educational and pedagogical contexts: What is happening outside and between courses? How are students developing their study behaviors throughout their studies? Are there educational designs or teaching approaches that can support students' study behaviors?
- Exploring the roles of informal and social learning in computing: What are students doing outside of organized, formal learning? What social behaviors are important for learning computing?

Although the measurement of study behaviors in computing education research was not the main focus of this systematic review, we cannot avoid addressing how behaviors are being researched and the link to theory. Considering the prevalence of self-defined, data-driven definitions and the reliance on questionnaires found in the reviewed works, it seems to be the data points that drive research on study behaviors rather than theory. This approach has implications for future practice and research, and it is important to raise the question of whether we measure what we think we are. As Prather et al. [130, p. 11] point out, "self-report measurements of cognitive control, such as the MSLQ, often measure what students think they do, rather than what they actually do." The limitations of self-reported measurements are one thing, but we must also consider how researchers interpret data from other sources. Computing education has the benefit of access to much log-file data on students; however, we must be careful in what we can infer from such data [130]. There have been some interesting developments on how log-file data can be used to identify cognitive processes, and calls have been made for further development into identifying effective indicators across disciplines [153] and for dealing with the invisible activities that happen in breaks of data [94]. There are also some developments in the field of learning analytics and multi-modal data in connection to cognitive processing that will be interesting for computing education to follow [21, 55]. With the abundance of data available to computing education researchers through compilers, version control systems, and IDEs, computing education research is well situated to be a part of this development.

7.5 Limitations

The main limitations of this review are biases in the selection, search, and data extraction. The choice to limit the review to English publications may have led to the omission some papers and may be a partial explanation for why North America and Australia/New Zealand were found to have larger output relative to population. Some researchers in various countries may treat computing education research as a side topic, whereas their main research is primarily technical computer science. While focusing on English language venues for their main research, they might, to some extent, present the education-related works in lower-prestige national venues. However, if we were to include papers in the few non-English languages that we also understand, then it would likely have led to more bias, not less, and would have reduced the review's repeatability for researchers of different language backgrounds. The choice to focus on English is commonly made for systematic reviews. Furthermore, there might be educators and researchers out there who do not publish their investigations of study behaviors. Hence, this systematic literature review is limited to what peer reviewers have accepted, not what practitioners attempt to research.

The authors attempted to ensure an unbiased review process by developing a research protocol in advance with predefined research questions. The search string was developed using the research questions and considering a possible lack of standardization in keywords, as they can be discipline- and language-specific. Furthermore, we performed a search in relevant conferences and journal databases for the computing education discipline. In the data extraction and analysis phase, steps were taken to ensure that at least two authors independently examined the data. Based on the finding that theoretical definitions were lacking, we reflected on the implications of our quality criteria. The use of theory was not required for inclusion, which is not uncommon for literature reviews of this sort. Although the quality of the included research papers can be questioned, we do not believe that this possibility substantially diminishes the contributions of the article. Finally, the selected methodology is an in-depth investigation of a relatively narrow area, using specific and pointed research questions that entail certain limitations [10].

8 CONCLUSION

This review of study behaviors in computing education research aimed to investigate how the computing education research community has approached computing students' study behavior. In total, we analyzed 107 peer-reviewed articles from 1994 to 2019. We explored how study behaviors are defined and included in the research as well as the role of the educational context within the computing education field. The results indicate that what computing students do both in and outside the classroom when learning computing topics is of increasing interest to researchers and educators. We also found that that the terminology used to define study behaviors is challenging to navigate. Many different theories and data analysis approaches are in use, providing an excellent foundation to further strengthen the relationship between computing education and higher education disciplines [3]. However, there currently is a need to create common ground between higher education theories and definitions and computing education research. Simultaneously, educational context plays an under-communicated role in existing research, and context needs to be included in future works in a systematic way. The nature of the computing education discipline can facilitate great progress in gaining and utilizing knowledge about how students study. Nevertheless, when researching study behavior, the field of computing education can benefit from not "reinventing the wheel" for every new experiment and dataset, and the taxonomy of study behaviors in computing education presented in this article can provide a good starting point. We intend for this article to serve as a resource for the computing education research community to help practitioners find relevant work on study behaviors and to help researchers make clear contributions to the literature.

APPENDIX

A COMPLETE LIST OF INCLUDED PAPERS

Table 11. List of Selected Papers in Alphabetical Order by Author

Title	Year	Author(s)	Paper ID
A Study of Pair Programming Enjoyment and Attendance using Study Motivation and Strategy Metrics.	2018	Aarne et al. [1]	1
The effects of extra credit opportunities on student procrastination.	2013	Allevato and Edwards [2]	2
Gender Differences in Students' Behaviors in CS Classes throughout the CS Major	2017	Alvarado et al. [4]	3
Assessment of self-regulated attitudes and behaviors of introductory programming students	2012	Ambrosio et al. [5]	4
Altering Study Habits with Email Reminders.	2013	Au et al. [6]	5
Prior Knowledge Dwarfs Hard Work in Achieving Academic Performance	2017	Au et al. [7]	6
Harmful Study Habits in Online Learning Environments with Automatic Assessment	2015	Auvinen [8]	7
When Life and Learning Do Not Fit: Challenges of Workload and Communication	2012	Benda et al. [11]	8
Promoting Students' Social Interactions Results in an Improvement	2018	Cabo and Satyanarayana [19]	9
Retaking object-oriented programming quizzes for study habit insights and improvements	2019	Carpenter and McCusker [20]	10
Using learning style data in an introductory computer science course	1999	Chamillard and Karolick [22]	11
Learning styles across the curriculum	2005	Chamillard and Sward [23]	12
Study habits of CS1 students: what do they do outside the classroom?	2010	Chinn et al. [24]	13
Finding traces of self-regulated learning in activity streams.	2018	Cicchinelli et al. [25]	14
Facilitating Course Assessment with a Competitive Programming Platform	2019	Coore and Fokum [27]	15
Introducing and Evaluating Exam Wrappers in CS2	2016	Craig et al. [28]	16
Predicting Success in University First Year Computing Science Courses: The Role of Student Participation in Reflective Learning Activities and in I-clicker Activities	2015	Cukierman [31]	17

(Continued)

Table 11. Continued

Title	Year	Author(s)	Paper ID
The Academic Enhancement Program: Assessing Programs Designed to Support Student Success	2019	Cukierman et al. [32]	18
Teaching and assessing programming strategies explicitly	2009	de Raadt et al. [33]	19
Predictors of academic achievement of student ICT teachers with different learning styles	2009	Deryakulu et al. [34]	20
Comparing effective and ineffective behaviors of student programmers	2009	Edwards et al. [41]	21
Examining Classroom Interventions to Reduce Procrastination	2015	Edwards et al. [40]	22
The academic enhancement program in introductory CS: A workshop framework description and evaluation	2011	Egan et al. [42]	23
Can Interaction Patterns with Supplemental Study Tools Predict Outcomes in CS1?	2016	Estey and Coady [45]	24
Study Habits, Exam Performance, and Confidence: How Do Workflow Practices and Self-Efficacy Ratings Align?	2017	Estey and Coady [46]	25
Automatically Classifying Students in Need of Support by Detecting Changes in Programming Behaviour	2017	Estey et al. [47]	26
From Study Tactics to Learning Strategies: An Analytical Method for Extracting Interpretable Representations	2019	Fincham et al. [48]	27
Improving Study Methods of Computer Engineering Undergraduates in Singapore	1996	Foo and Ng [49]	28
Exploring students learning behavior with an interactive etextbook in computer science courses	2014	Fouh et al. [51]	29
The professor on your PC: a virtual CS1 course	2009	Gal-Ezer et al. [54]	30
Learning and the Reflective Journal in Computer Science	2002	George [56]	31
Student Behaviour in Unsupervised Online Quizzes: A Closer Look	2018	Gholami and Zhang [58]	32
How Widely Can Prediction Models Be Generalized? Performance Prediction in Blended Courses	2019	Gitinabard et al. [59]	33
A study on students' behaviours and attitudes towards learning to program	2012	Gomes et al. [61]	34

(Continued)

Table 11. Continued

Title	Year	Author(s)	Paper ID
The learning context: Influence on learning to program	2009	Govender [62]	36
Insights on supporting learning during computing science and engineering students' transition to university: A design-oriented, mixed methods exploration of instructor and student perspectives	2017	Guloy et al. [63]	37
Psychology assisted prediction of academic performance using machine learning	2016	Halde et al. [65]	38
Supporting quality teaching using educational data mining based on OpenEdX platform	2017	Han et al. [66]	40
Improving first-year success and retention through interest-based CS0 courses	2012	Haungs et al. [68]	41
Metacognitive calibration when learning to program	2017	Hauswirth and Adamoli [69]	42
Improving Study Skills by Combining a Study Skill Module and Repeated Reflection Seminars	2019	Hedin and Kann [70]	43
On the Bimodality in an Introductory Programming Course: An Analysis of Student Performance Factors	2015	Höök and Eckerdal [76]	45
Stereotype Modeling for Problem-Solving Performance Predictions in MOOCs and Traditional Courses	2017	Hosseini et al. [73]	46
ASSISTing CS1 students to learn: learning approaches and object-oriented programming	2006	Hughes and Peiris [74]	47
How Can Learning Analytics Improve a Course?	2017	Hui and Farvolden [75]	48
Study strategies of online learners	2011	Iscioglu [77]	49
Teaching programming by emphasizing self-direction: How did students react to the active role required of them?	2013	Isomöttönen and Tirronen [78]	50
Flipping and Blending—An Action Research Project on Improving a Functional Programming Course	2016	Isomöttönen and Tirronen [79]	51
Issues with a course that emphasizes self-direction	2013	Isomöttönen et al. [80]	52
Effects of a Program Integrating Course for Students of Computer Science and Engineering	2016	Kann and Högfeldt [81]	53

(Continued)

Table 11. Continued

Title	Year	Author(s)	Paper ID
CS minors in a CS1 course	2008	Kinnunen and Malmi [84]	54
Through the eyes of instructors: a phenomenographic investigation of student success.	2007	Kinnunen et al. [85]	55
Penetrating the black box of time-on-task estimation	2015	Kovanović et al. [89]	56
Examining communities of inquiry in Massive Open Online Courses: The role of study strategies	2019	Kovanović et al. [90]	57
An expert system for the prediction of student performance in an initial computer science course	2017	Kuehn et al. [91]	58
The Effectiveness of Video Quizzes in a Flipped Class	2015	Lacher and Lewis [93]	59
Pauses and spacing in learning to program	2016	Leppänen et al. [94]	60
Behaviors of Higher and Lower Performing Students in CS1	2019	Liao et al. [95]	62
Exploring the Network Dynamics in a Flipped Classroom	2018	Lin and Wu [96]	63
Cross-cultural education: learning methodology and behaviour analysis for Asian students in IT field of Australian universities	2010	Lu et al. [99]	64
Problem-based learning with database systems	1994	Ma [101]	65
Video-based instruction for introductory computer programming	2014	Manley and Urness [103]	66
The Effects of Procrastination Interventions on Programming Project Success	2015	Martin et al. [105]	67
Game elements in a software engineering study group: A case study	2017	Matsubara and da Silva [108]	35
When Practice Doesn't Make Perfect: Effects of Task Goals on Learning Computing Concepts	2011	Miller and Settle [111]	71
Making connections: First year transition for computer science and software engineering students	2005	Moffat et al. [112]	72
Modeling Students Self-studies Behaviors	2015	Mota et al. [113]	73
Social Help-seeking Strategies in a Programming MOOC	2018	Nelimarkka and Hellas [114]	74

(Continued)

Table 11. Continued

Title	Year	Author(s)	Paper ID
Undergraduate students in a computer engineering course: a perspective of their learning approaches and motivation factors	1997	Ng and Ng [117]	75
Examining the mediating role of learning engagement, learning process and learning experience on the learning outcomes through localized real case studies	2014	Nkhoma et al. [118]	76
Investigating Students' Achievements in Computing Science Using Human Metric	2014	Okike [120]	77
Illustrating performance indicators and course characteristics to support students' self-regulated learning in CS1	2015	Ott et al. [121]	78
Does the introduction of an overall study strategy empower students to use appropriate study strategies?	2017	Oysaed et al. [122]	79
What's the Problem? Teachers' Experience of Student Learning Successes and Failures	2007	Pears et al. [124]	80
Revisiting why students drop CS1	2016	Petersen et al. [125]	81
Approaches to studying in first-year engineering: comparison between inventory scores and students' descriptions of their approaches through interviews	2018	Pettersson et al. [126]	82
Anchoring interactive points of interest on web-based instructional video: effects on students' interaction behavior and perceived experience	2019	Pimentel et al. [127]	83
SAT Does Not Spell Success: How Non-Cognitive Factors Can Explain Variance in the GPA of Undergraduate Engineering and Computer Science Students	2019	Scheidt et al. [136]	87
Evaluating a Linked-courses Learning Community for Development Majors	2015	Settle et al. [138]	89
Ludwig: an online programming tutoring and assessment system	2005	Shaffer [139]	90
Study Habits of CS 1 Students: What Do They Say They Do?	2013	Sheard et al. [140]	91
On multi-device use: Using technological modality profiles to explain differences in students' learning	2019	Sher et al. [141]	92
Predicting Student Performance Based on Online Study Habits: A Study of Blended Courses.	2018	Sheshadri et al. [142]	93

(Continued)

Table 11. Continued

Title	Year	Author(s)	Paper ID
An Exploration of Grit in a CS1 Context	2018	Sigurdson and Petersen [143]	94
Predictors of success in a first programming course	2006	Simon et al. [144]	95
Analyzing self-reflection by Computer Science students to identify bad study habits: Self-reflection performed by students of programming courses on the study habits and skills acquired through b-learning supported by an automatic judge	2010	Sustelo and Guerreiro [146]	96
Gender neutrality improved completion rate for all	2016	Svedin and Bälter [147]	97
Repertory grid: investigating personal constructs of novice programmers	2011	Thota [149]	98
The influence of Web-supported independent activities and small group work on students' epistemological beliefs	2004	Tolhurst [151]	99
How novices tackle their first lines of code in an IDE: Analysis of programming session traces	2014	Vihavainen et al. [154]	101
Approaches of Learning and Computational Thinking in Students that get into the Computer Sciences Career	2018	Villalba-Condori et al. [155]	102
The use of lecture videos, eBooks, and clickers in computer courses	2014	Vinaja [156]	103
Pedagogical Intervention Practices: Improving Learning Engagement Based on Early Prediction	2019	Wan et al. [157]	104
SmartGPA: how smartphones can assess and predict academic performance of college students	2015	Wang et al. [158]	105
The combined effect of self-efficacy and academic integration on higher education students studying IT majors in Taiwan	2010	Weng et al. [159]	107
Teaching OO concepts—A new approach	2004	Westin and Nordstrom [160]	108
Implementation of alternative pacing in an introductory programming sequence	2003	Whittington et al. [161]	109
Using online self-assessment in introductory programming classes	2006	Williams et al. [162]	110
On study habits on an introductory course on programming	2015	Willman et al. [164]	111

(Continued)

Table 11. Continued

Title	Year	Author(s)	Paper ID
A Spaced, Interleaved Retrieval Practice Tool that is Motivating and Effective	2019	YeckehZaare et al. [166]	112
Effects of YouTube videos as pre-lecture preparation	2019	Yim et al. [167]	113
Investigation of the Relationship between Learning Process and Learning Outcomes in E-learning Environments	2015	Yurdugul and Menzi Cetin [168]	114
Finding competence characteristics among first semester students in computer science	2015	Zehetmeier et al. [169]	115
DataLab: Introducing Software Engineering Thinking into Data Science Education at Scale	2017	Zhang et al. [170]	116
The Effects of ICT Use on Chinese College Students' Study Behavior in B-learning Evaluating B-learning Effectiveness via Causal Model	2019	Zhao et al. [171]	117
Impact of Student Achievement Goals on CS1 Outcomes	2016	Zingaro and Porter [173]	118
Achievement Goals in CS1: Replication and Extension	2018	Zingaro et al. [172]	119

Paper ID refers to the key in the results from the coding process at this link [removed for anonymous review but will be made available in an online format upon publication].

ACKNOWLEDGMENTS

We would like to especially acknowledge Tore Sletten Langeland for valuable input on the taxonomy. In addition, we are extremely grateful to the associate editor and the reviewers for their constructive comments and useful insights, which significantly improved the article.

REFERENCES

[1] Onni Aarne, Petrus Peltola, Juho Leinonen, and Arto Hellas. 2018. A study of pair programming enjoyment and attendance using study motivation and strategy metrics. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE'18)*. Association for Computing Machinery, New York, NY, 759–764. DOI: https://doi.org/10.1145/3159450.3159493

[2] A. Allevato and S. H. Edwards. 2013. The effects of extra credit opportunities on student procrastination. In *Proceedings of the Frontiers in Education Conference (FIE'13)*. 1831–1836. DOI: https://doi.org/10.1109/FIE.2013.6685154

[3] Vicki L. Almstrum, Orit Hazzan, Mark Guzdial, and Marian Petre. 2005. Challenges to computer science education research. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education (SIGCSE'05)*. ACM, New York, NY, 191–192. DOI: https://doi.org/10.1145/1047344.1047415

[4] Christine Alvarado, Yingjun Cao, and Mia Minnes. 2017. Gender differences in students' behaviors in CS classes throughout the CS major. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE'17)*. Association for Computing Machinery, New York, NY, 27–32. DOI: https://doi.org/10.1145/3017680.3017771

[5] A. P. Ambrosio, L. Almeida, A. Franco, S. Martins, and F. Georges. 2012. Assessment of self-regulated attitudes and behaviors of introductory programming students. DOI: https://doi.org/10.1109/FIE.2012.6462314

[6] Oliver Au, Raymond So, and Hiu-Wing Go. 2013. Altering study habits with email reminders *(Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics))*, Vol. 8038 LNCS, 190–198. DOI: https://doi.org/10.1007/978-3-642-39750-9_18

[7] Oliver Au, Raymond So, and Lap Kei Lee. 2017. Prior knowledge dwarfs hard work in achieving academic performance. In *Blended Learning. New Challenges and Innovative Practices (Lecture Notes in Computer Science)*, Simon K. S. Cheung, Lam-for Kwok, Will W. K. Ma, Lap-Kei Lee, and Harrison Yang (Eds.). Springer International Publishing, Cham, 139–147. DOI: https://doi.org/10.1007/978-3-319-59360-9_13

[8] T. Auvinen. 2015. Harmful study habits in online learning environments with automatic assessment. In *Proceedings of the International Conference on Learning and Teaching in Computing and Engineering*. 50–57. DOI: https://doi.org/10.1109/LaTiCE.2015.31

[9] Albert Bandura. 1978. The self system in reciprocal determinism. *Amer. Psychol.* 33, 4 (1978), 344–358. DOI: https://doi.org/10.1037/0003-066X.33.4.344

[10] Margaret Bearman, Calvin D. Smith, Angela Carbone, Susan Slade, Chi Baik, Marnie Hughes-Warrington, and David L. Neumann. 2012. Systematic review methodology in higher education. *High. Educ. Res. Devel.* 31, 5 (Oct. 2012), 625–640. DOI: https://doi.org/10.1080/07294360.2012.702735

[11] Klara Benda, Amy Bruckman, and Mark Guzdial. 2012. When life and learning do not fit: Challenges of workload and communication in introductory computer science online. *ACM Trans. Comput. Educ.* 12, 4 (Nov. 2012). DOI: https://doi.org/10.1145/2382564.2382567

[12] Anders Berglund and Mattias Wiggberg. 2006. Students learn CS in different ways: Insights from an empirical study. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITICSE'06).* Association for Computing Machinery, New York, NY, 265–269. DOI: https://doi.org/10.1145/1140124.1140195

[13] Naomi Berman. 2020. A critical examination of informal learning spaces. *High. Educ. Res. Devel.* 39, 1 (Jan. 2020), 127–140. DOI: https://doi.org/10.1080/07294360.2019.1670147

[14] John Biggs. 1979. Individual differences in study processes and the quality of learning outcomes. *High. Educ.* 8, 4 (1979), 381–394.

[15] John Biggs. 2012. What the student does: Teaching for enhanced learning. *High. Educ. Res. Devel.* 31, 1 (Feb. 2012), 39–55. DOI: https://doi.org/10.1080/07294360.2012.642839

[16] John Biggs, David Kember, and Doris Y. P. Leung. 2001. The revised two-factor study process questionnaire: R-SPQ-2F. *Brit. J. Educ. Psychol.* 71, 1 (2001), 133–149. DOI: https://doi.org/10.1348/000709901158433

[17] John B. Biggs. 1989. Approaches to the enhancement of tertiary teaching. *High. Educ. Res. Devel.* 8, 1 (Jan. 1989), 7–25. DOI: https://doi.org/10.1080/0729436890080102

[18] Jürgen Börstler and Thomas B. Hilburn. 2015. Team projects in computing education. *ACM Trans. Comput. Educ.* 15, 4 (Dec. 2015), 1–5. DOI: https://doi.org/10.1145/2808079

[19] Candido Cabo and Ashwin Satyanarayana. 2018. Promoting students' social interactions results in an improvement in performance, class attendance and retention in first year computing courses. In *Proceedings of the Frontiers in Education Conference.*

[20] A. Carpenter and J. R. McCusker. 2019. Retaking object-oriented programming quizzes for study habit insights and improvements. In *2019 ASEE Annual Conference & Exposition.*

[21] Leen Catrysse, David Gijbels, and Vincent Donche. 2019. A systematic review on the conceptualization and operationalization of students' levels of processing in functional magnetic resonance imaging studies. *Mind, Brain Educ.* 13, 3 (2019), 198–210. DOI: https://doi.org/10.1111/mbe.12199

[22] A. T. Chamillard and D. Karolick. 1999. Using learning style data in an introductory computer science course. In *Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education.* 291–295.

[23] A. T. Chamillard and Ricky E. Sward. 2005. Learning styles across the curriculum. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE'05).* Association for Computing Machinery, New York, NY, 241–245. DOI: https://doi.org/10.1145/1067445.1067512

[24] Donald Chinn, Judy Sheard, Angela Carbone, and Mikko-Jussi Laakso. 2010. Study habits of CS1 Students: What do they do outside the classroom? In *Proceedings of the 12th Australasian Conference on Computing Education (ACE'10).* Australian Computer Society, Inc., 53–62.

[25] Analía Cicchinelli, Eduardo Veas, Abelardo Pardo, Viktoria Pammer-Schindler, Angela Fessl, Carla Barreiros, and Stefanie Lindstädt. 2018. Finding traces of self-regulated learning in activity streams. In *Proceedings of the 8th international conference on learning analytics and knowledge (LAK'18).* Association for Computing Machinery, New York, NY, 191–200. DOI: https://doi.org/10.1145/3170358.3170381

[26] Frank Coffield. 2004. *Should We Be Using Learning Styles? What Research Has to Say in Practice.* Technical Report. Learning and Skills Research Centre, London.

[27] Daniel Coore and Daniel Fokum. 2019. Facilitating course assessment with a competitive programming platform. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE'19).* Association for Computing Machinery, New York, NY, 449–455. DOI: https://doi.org/10.1145/3287324.3287511

[28] Michelle Craig, Diane Horton, Daniel Zingaro, and Danny Heap. 2016. Introducing and evaluating exam wrappers in CS2. In *Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE'16)*. Association for Computing Machinery, New York, NY, 285–290. DOI: https://doi.org/10.1145/2839509.2844561

[29] Fergus I. M. Craik and Robert S. Lockhart. 1972. Levels of processing: A framework for memory research. *J. Verb. Learn. and Verb. Behav.* 11, 6 (Dec. 1972), 671–684. DOI: https://doi.org/10.1016/S0022-5371(72)80001-X

[30] Marcus Credé and Nathan R. Kuncel. 2008. Study habits, skills, and attitudes: The third pillar supporting collegiate academic performance. *Perspect. Psychol. Sci.* 3, 6 (Nov. 2008), 425–453. DOI: https://doi.org/10.1111/j.1745-6924.2008.00089.x

[31] Diana Cukierman. 2015. Predicting success in university first year computing science courses: The role of student participation in reflective learning activities and in i-Clicker activities. In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'15)*. Association for Computing Machinery, New York, NY, 248–253. DOI: https://doi.org/10.1145/2729094.2742623

[32] Diana R. Cukierman, Donna McGee Thompson, and Wayne Sun. 2019. The academic enhancement program: Assessing programs designed to support student success. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE'19)*. Association for Computing Machinery, New York, NY, 686–692. DOI: https://doi.org/10.1145/3287324.3287422

[33] Michael de Raadt, Richard Watson, and Mark Toleman. 2009. Teaching and assessing programming strategies explicitly. *New Zealand* 95 (2009), 10.

[34] D. Deryakulu, S. Büyüköztürk, and H. Özçinar. 2009. Predictors of academic achievement of student ICT teachers with different learning styles. *World Acad. Sci., Eng. Technol.* 58 (2009), 703–709.

[35] Cambridge Dictionary. 2021. taxonomy. Retrived 18 Feb, 2021 from https://cambridge.dictinary.com/taxonomy.

[36] Daniel L. Dinsmore. 2017. Toward a dynamic, multidimensional research framework for strategic processing. *Educ. Psychol. Rev.* 29, 2 (June 2017), 235–268. DOI: https://doi.org/10.1007/s10648-017-9407-5

[37] Daniel L. Dinsmore, Patricia A. Alexander, and Sandra M. Loughlin. 2008. Focusing the conceptual lens on metacognition, self-regulation, and self-regulated learning. *Educ. Psychol. Rev.* 20, 4 (Dec. 2008), 391–409. DOI: https://doi.org/10.1007/s10648-008-9083-6

[38] Angela Lee Duckworth and Patrick D. Quinn. 2009. Development and validation of the short grit scale (Grit–S). *J. Personal. Assess.* 91, 2 (Feb. 2009), 166–174. DOI: https://doi.org/10.1080/00223890802634290

[39] Tore Dybå and Torgeir Dingsøyr. 2008. Empirical studies of agile software development: A systematic review. *Inf. Softw. Technol.* 50, 9 (Aug. 2008), 833–859. DOI: https://doi.org/10.1016/j.infsof.2008.01.006

[40] Stephen H. Edwards, Joshua Martin, and Clfford A. Shaffer. 2015. Examining classroom interventions to reduce procrastination. In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'15)*. Association for Computing Machinery, New York, NY, 254–259. DOI: https://doi.org/10.1145/2729094.2742632

[41] Stephen H. Edwards, Jason Snyder, Manuel A. Pérez-Quiñones, Anthony Allevato, Dongkwan Kim, and Betsy Tretola. 2009. Comparing effective and ineffective behaviors of student programmers. In *Proceedings of the 5th International Workshop on Computing Education Research Workshop (ICER'09)*. ACM, New York, NY, 3–14. DOI: https://doi.org/10.1145/1584322.1584325

[42] Rylan Egan, Diana Cukierman, and Donna McGee Thompson. 2011. The academic enhancement program in introductory CS: A workshop framework description and evaluation. In *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education (ITiCSE'11)*. Association for Computing Machinery, New York, NY, 278–282. DOI: https://doi.org/10.1145/1999747.1999825

[43] Noel Entwistle and Paul Ramsden. 2015. *Understanding Student Learning (Routledge Revivals)*. Routledge. DOI: https://doi.org/10.4324/9781315718637

[44] N. J. Entwistle. 1991. Approaches to learning and perceptions of the learning environment: Introduction to the Special Issue. *High. Educ.* 22, 3 (1991), 201–4.

[45] Anthony Estey and Yvonne Coady. 2016. Can interaction patterns with supplemental study tools predict outcomes in CS1? In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'16)*. Association for Computing Machinery, New York, NY, 236–241. DOI: https://doi.org/10.1145/2899415.2899428

[46] Anthony Estey and Yvonne Coady. 2017. Study habits, exam performance, and confidence: How do workflow practices and self-efficacy ratings align? In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'17)*. Association for Computing Machinery, New York, NY, 158–163. DOI: https://doi.org/10.1145/3059009.3059056

[47] Anthony Estey, Hieke Keuning, and Yvonne Coady. 2017. Automatically classifying students in need of support by detecting changes in programming behaviour. In *Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE'17)*. Association for Computing Machinery, New York, NY, 189–194. DOI: https://doi.org/10.1145/3017680.3017790

[48] E. Fincham, D. Gašević, J. Jovanović, and A. Pardo. 2019. From study tactics to learning strategies: An analytical method for extracting interpretable representations. *IEEE Trans. Learn. Technol.* 12, 1 (2019), 59–72. DOI: https://doi.org/10.1109/TLT.2018.2823317

[49] S. Foo and G. S. Ng. 1996. Improving study methods of computer engineering undergraduates in Singapore. *Qual. High. Educ.* 2, 2 (1996), 131–142. DOI: https://doi.org/10.1080/1353832960020205

[50] The Joint Task Force for Computing Curricula. 2005. *Computing Curricula 2005 - The Overview Report*. Technical Report. ACM, AIS, IEEE.

[51] E. Fouh, D. A. Breakiron, S. Hamouda, M. F. Farghally, and C. A. Shaffer. 2014. Exploring students' learning behavior with an interactive etextbook in computer science courses. *Comput. Hum. Behav.* 41 (2014), 478–485. DOI: https://doi.org/10.1016/j.chb.2014.09.061

[52] Stephen Frezza, Mats Daniels, Arnold Pears, Åsa Cajander, Viggo Kann, Amanpreet Kapoor, Roger McDermott, Anne-Kathrin Peters, Mihaela Sabin, and Charles Wallace. 2018. Modelling competencies for computing education beyond 2020: A research based approach to defining competencies in the computing disciplines. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'18)*. Association for Computing Machinery, New York, NY, 148–174. DOI: https://doi.org/10.1145/3293881.3295782

[53] Luke K. Fryer, Paul Ginns, Richard A. Walker, and Kaori Nakao. 2012. The adaptation and validation of the CEQ and the R-SPQ-2F to the Japanese tertiary environment. *Brit. J. Educ. Psychol.* 82, 4 (2012), 549–563. DOI: https://doi.org/10.1111/j.2044-8279.2011.02045.x

[54] Judith Gal-Ezer, Tamar Vilner, and Ela Zur. 2009. The professor on your PC: A virtual CS1 course. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE'09)*. Association for Computing Machinery, New York, NY, 191–195. DOI: https://doi.org/10.1145/1562877.1562938

[55] Dragan Gaševic, Jelena Jovanovic, Abelardo Pardo, and Shane Dawson. 2017. Detecting learning strategies with analytics: Links with self-reported measures and academic performance. *J. Learn. Anal.* 4, 2 (2017), 113–128.

[56] Susan E. George. 2002. Learning and the reflective journal in computer science. In *Proceedings of the 25th Australasian Conference on Computer Science (ACSC'02)*. Australian Computer Society, Inc., AUS, 77–86.

[57] Maribeth Gettinger and Jill K. Seibert. 2002. Contributions of study skills to academic competence. *Study Skills* 31, 3 (2002), 350–365.

[58] A. Gholami and L. Y. Zhang. 2018. Student behaviour in unsupervised online quizzes: A closer look. In *Proceedings of the 23rd Western Canadian Conference on Computing Education*. DOI: https://doi.org/10.1145/3209635.3209650

[59] N. Gitinabard, Y. Xu, S. Heckman, T. Barnes, and C. F. Lynch. 2019. How widely can prediction models be generalized? Performance prediction in blended courses. *IEEE Trans. Learn. Technol.* 12, 2 (2019), 184–197. DOI: https://doi.org/10.1109/TLT.2019.2911832

[60] Seth Copen Goldstein, Hongyi Zhang, Majd Sakr, Haokang An, and Cameron Dashti. 2019. Understanding how work habits influence student performance. In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'19)*. ACM, New York, NY, 154–160. DOI: https://doi.org/10.1145/3304221.3319757

[61] Anabela Jesus Gomes, Alvaro Nuno Santos, and António José Mendes. 2012. A study on students' behaviours and attitudes towards learning to program. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'12)*. ACM, New York, NY, 132–137. DOI: https://doi.org/10.1145/2325296.2325331

[62] Irene Govender. 2009. The learning context: Influence on learning to program. *Comput. Educ.* 53, 4 (Dec. 2009), 1218–1230. DOI: https://doi.org/{10.1016/j.compedu.2009.06.005}

[63] Sheryl Guloy, Farimah Salimi, Diana Cukierman, and Donna McGee Thompson. 2017. Insights on supporting learning during computing science and engineering students' transition to university: A design-oriented, mixed methods exploration of instructor and student perspectives. *High. Educ.* 73, 3, SI (Mar. 2017), 479–497. DOI: https://doi.org/{10.1007/s10734-016-0097-6}

[64] Tamsin Haggis. 2003. Constructing images of ourselves? A critical investigation into "approaches to learning" research in higher education. *Brit. Educ. Res. J.* 29, 1 (2003), 89–104. DOI: https://doi.org/10.1080/0141192032000057401

[65] R. R. Halde, A. Deshpande, and A. Mahajan. 2016. Psychology assisted prediction of academic performance using machine learning. In *Proceedings of the IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT)*. 431–435. DOI: https://doi.org/10.1109/RTEICT.2016.7807857

[66] Wan Han, Ding Jun, Gao Xiaopeng, and Liu Kangxu. 2017. Supporting quality teaching using educational data mining based on OpenEDX platform. In *Proceedings of the IEEE Frontiers in Education Conference (FIE'17)*. 1–7. DOI: https://doi.org/10.1109/FIE.2017.8190730

[67] David Harel. 1980. On folk theorems. *Commun. ACM* 23, 7 (July 1980), 379–389. DOI: https://doi.org/10.1145/358886.358892

[68] Michael Haungs, Christopher Clark, John Clements, and David Janzen. 2012. Improving first-year success and retention through interest-based CS0 courses. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE'12)*. Association for Computing Machinery, New York, NY, 589–594. DOI: https://doi.org/10.1145/2157136.2157307

[69] Matthias Hauswirth and Andrea Adamoli. 2017. Metacognitive calibration when learning to program. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research (Koli Calling'17)*. Association for Computing Machinery, New York, NY, 50–59. DOI: https://doi.org/10.1145/3141880.3141904

[70] Bjorn Hedin and Viggo Kann. 2019. Improving study skills by combining a study skill module and repeated reflection seminars. *Educ. Res. Int.* (2019). DOI: https://doi.org/{10.1155/2019/9739854}

[71] Arto Hellas, Petri Ihantola, Andrew Petersen, Vangel V. Ajanovski, Mirela Gutica, Timo Hynninen, Antti Knutas, Juho Leinonen, Chris Messom, and Soohyun Nam Liao. 2018. Predicting academic performance: A systematic literature review. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'18)*. ACM, New York, NY, 175–199. DOI: https://doi.org/10.1145/3293881.3295783

[72] Barbara K. Hofer. 2000. Dimensionality and disciplinary differences in personal epistemology. *Contemp. Educ. Psychol.* 25, 4 (Oct. 2000), 378–405. DOI: https://doi.org/10.1006/ceps.1999.1026

[73] Roya Hosseini, Peter Brusilovsky, Michael Yudelson, and Arto Hellas. 2017. Stereotype modeling for problem-solving performance predictions in MOOCs and traditional courses. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP'17)*. Association for Computing Machinery, New York, NY, 76–84. DOI: https://doi.org/10.1145/3079628.3079672

[74] Janet Hughes and D. Ramanee Peiris. 2006. ASSISTing CS1 students to learn: Learning approaches and object-oriented programming. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITICSE'06)*. Association for Computing Machinery, New York, NY, 275–279. DOI: https://doi.org/10.1145/1140124.1140197

[75] Bowen Hui and Shannon Farvolden. 2017. How can learning analytics improve a course? In *Proceedings of the 22nd Western Canadian Conference on Computing Education (WCCCE'17)*. Association for Computing Machinery, New York, NY. DOI: https://doi.org/10.1145/3085585.3085586

[76] Lars Josef Höök and Anna Eckerdal. 2015. On the bimodality in an introductory programming course: An analysis of student performance factors. In *Proceedings of the International Conference on Learning and Teaching in Computing and Engineering*. 79–86. DOI: https://doi.org/10.1109/LaTiCE.2015.25

[77] Ersun Iscioglu. 2011. Study strategies of online learners. In *Proceedings of the International Conference on Virtual Learning*. 358–364.

[78] Ville Isomöttönen and Ville Tirronen. 2013. Teaching programming by emphasizing self-direction: How did students react to the active role required of them? *ACM Trans. Comput. Educ.* 13, 2 (July 2013). DOI: https://doi.org/10.1145/2483710.2483711

[79] Ville Isomöttönen and Ville Tirronen. 2016. Flipping and Blending—An action research project on improving a functional programming course. *ACM Trans. Comput. Educ.* 17, 1 (Sept. 2016). DOI: https://doi.org/10.1145/2934697

[80] Ville Isomöttönen, Ville Tirronen, and Michael Cochez. 2013. Issues with a course that emphasizes self-direction. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'13)*. Association for Computing Machinery, New York, NY, 111–116. DOI: https://doi.org/10.1145/2462476.2462495

[81] Viggo Kann and Anna-Karin Högfeldt. 2016. Effects of a program integrating course for students of computer science and engineering. In *Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE'16)*. Association for Computing Machinery, New York, NY, 510–515. DOI: https://doi.org/10.1145/2839509.2844610

[82] Avi Kaplan. 2008. Clarifying metacognition, self-regulation, and self-regulated learning: What's the purpose? *Educ. Psychol. Rev.* 20, 4 (Sept. 2008), 477. DOI: https://doi.org/10.1007/s10648-008-9087-2

[83] Staffs Keele. 2007. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. Technical report 2.3. EBSE.

[84] Päivi Kinnunen and Lauri Malmi. 2008. CS minors in a CS1 course. In *Proceedings of the 4th International Workshop on Computing Education Research (ICER'08)*. Association for Computing Machinery, New York, NY, 79–90. DOI: https://doi.org/10.1145/1404520.1404529

[85] Päivi Kinnunen, Robert McCartney, Laurie Murphy, and Lynda Thomas. 2007. Through the eyes of instructors: A phenomenographic investigation of student success. In *Proceedings of the 3rd International Workshop on Computing Education Research (ICER'07)*. ACM, New York, NY, 61–72. DOI: https://doi.org/10.1145/1288580.1288589

[86] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering—A systematic literature review. *Inf. Softw. Technol.* 51, 1 (Jan. 2009), 7–15. DOI: https://doi.org/10.1016/j.infsof.2008.09.009

[87] Alice Kolb and David Kolb. 2005. The Kolb learning style inventory—version 3.1 2005 technical specifications. (Jan. 2005). *Hay Resource Direct* 200, 72 (2005), 166–171.

[88] David Kolb. 1984. *Experiential Learning: Experience as the Source of Learning and Development.* Vol. 1. FT Press.

[89] Vitomir Kovanović, Dragan Gašević, Shane Dawson, Srećko Joksimović, Ryan S. Baker, and Marek Hatala. 2015. Penetrating the black box of time-on-task estimation. In *Proceedings of the 5th International Conference on Learning Analytics and Knowledge (LAK'15)*. ACM Press, 184–193. DOI: https://doi.org/10.1145/2723576.2723623

[90] Vitomir Kovanović, Srećko Joksimović, Oleksandra Poquet, Thieme Hennis, Pieter de Vries, Marek Hatala, Shane Dawson, George Siemens, and Dragan Gašević. 2019. Examining communities of inquiry in massive open online courses: The role of study strategies. *Internet High. Educ.* 40 (Jan. 2019), 20–43. DOI: https://doi.org/10.1016/j.iheduc.2018.09.001

[91] M. Kuehn, J. Estad, J. Straub, T. Stokke, and S. Kerlin. 2017. An expert system for the prediction of student performance in an initial computer science course. In *Proceedings of the IEEE International Conference on Electro Information Technology (EIT'17)*. 1–6. DOI: https://doi.org/10.1109/EIT.2017.8053321

[92] Agnieszka Kurczewska, Paula Kyrö, Krista Lagus, Oskar Kohonen, and Tiina Lindh-Knuutila. 2018. The interplay between cognitive, conative, and affective constructs along the entrepreneurial learning process. *Educ. + Train.* 60, 7/8 (Jan. 2018), 891–908. DOI: https://doi.org/10.1108/ET-09-2016-0148

[93] Lisa L. Lacher and Mark C. Lewis. 2015. The effectiveness of video quizzes in a flipped class. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE'15)*. Association for Computing Machinery, New York, NY, 224–228. DOI: https://doi.org/10.1145/2676723.2677302

[94] Leo Leppänen, Juho Leinonen, and Arto Hellas. 2016. Pauses and spacing in learning to program. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling'16)*. Association for Computing Machinery, New York, NY, 41–50. DOI: https://doi.org/10.1145/2999541.2999549

[95] Soohyun Nam Liao, Sander Valstar, Kevin Thai, Christine Alvarado, Daniel Zingaro, William G. Griswold, and Leo Porter. 2019. Behaviors of higher and lower performing students in CS1. In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'19)*. Association for Computing Machinery, New York, NY, 196–202. DOI: https://doi.org/10.1145/3304221.3319740

[96] Fang-Ling Lin and Chyi-In Wu. 2018. Exploring the network dynamics in a flipped classroom. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'18)*. IEEE Press, 855–860.

[97] Alf Lizzio, Keithia Wilson, and Roland Simons. 2002. University students' perceptions of the learning environment and academic outcomes: Implications for theory and practice. *Stud. High. Educ.* 27, 1 (Feb. 2002), 27–52. DOI: https://doi.org/10.1080/03075070120099359

[98] M. Lorås and T. Aalberg. 2020. First year computing study behavior: Effects of educational design. In *Proceedings of the IEEE Frontiers in Education Conference (FIE'20)*. 1–9. DOI: https://doi.org/10.1109/FIE44824.2020.9274285

[99] Jie Lu, K. L. Chin, Juan Yao, Jun Xu, and Jitian Xiao. 2010. Cross-cultural education: Learning methodology and behaviour analysis for Asian students in IT field of Australian universities. In *Proceedings of the 12th Australasian Conference on Computing Education (ACE'10)*. Australian Computer Society, Inc., AUS, 117–126.

[100] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 2018. Introductory programming: A systematic literature review. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'18)*. ACM, New York, NY, 55–106. DOI: https://doi.org/10.1145/3293881.3295779

[101] Jian Ma. 1994. Problem-based learning with database systems. *Comput. Educ.* 22, 3 (Apr. 1994), 257–263. DOI: https://doi.org/10.1016/0360-1315(94)90007-8

[102] Lauri Malmi, Judy Sheard, and Päivi Kinnunen. 2019. Computing education theories: What are they and how are they used? In *Proceedings of the 2019 ACM Conference on International Computing Education Research*.

[103] Eric D. Manley and Timothy M. Urness. 2014. Video-based instruction for introductory computer programming. *J. Comput. Sci. Coll.* 29, 5 (May 2014), 221–227.

[104] Delia Marshall and Jennifer Case. 2005. "Approaches to learning" research in higher education: A response to Haggis. *Brit. Educ. Res. J.* 31, 2 (2005), 257–267. DOI: https://doi.org/10.1080/0141192052000340242

[105] Joshua Martin, Stephen H. Edwards, and Clfford A. Shaffer. 2015. The effects of procrastination interventions on programming project success. In *Proceedings of the 11th Annual International Conference on International Computing Education Research (ICER'15)*. ACM, New York, NY, 3–11. DOI: https://doi.org/10.1145/2787622.2787730

[106] F. Marton and R. Säljö. 1976. On qualitative differences in learning: "outcome and process." *Brit. J. Educ. Psychol.* 46, 1 (Feb. 1976), 4–11. DOI: https://doi.org/10.1111/j.2044-8279.1976.tb02980.x

[107] Chris Masui, Jan Broeckmans, Sarah Doumen, Anne Groenen, and Geert Molenberghs. 2014. Do diligent students perform better? Complex relations between student and course characteristics, study time, and academic performance in higher education. *Stud. High. Educ.* 39, 4 (Apr. 2014), 621–643. DOI: https://doi.org/10.1080/03075079.2012.721350

[108] Patrícia Gomes Fernandes Matsubara and Caroline Lima Corrundefineda da Silva. 2017. Game elements in a software engineering study group: A case study. In *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track (ICSE-SEET'17)*. IEEE Press, 160–169. DOI: https://doi.org/10.1109/ICSE-SEET.2017.8

[109] Robert McCartney, Anna Eckerdal, Jan Erik Mostrom, Kate Sanders, and Carol Zander. 2007. Successful students' strategies for getting unstuck. In *Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE'07)*. ACM, New York, NY, 156–160. DOI: https://doi.org/10.1145/1268784.1268831

[110] J. H. F. Meyer and M. W. Muller. 1990. Evaluating the quality of student learning. An unfolding analysis of the association between perceptions of learning context and approaches to studying at an individual level. *Stud. High. Educ.* 15, 2 (Jan. 1990), 131–154. DOI: https://doi.org/10.1080/03075079012331377471

[111] Craig S. Miller and Amber Settle. 2011. When practice doesn't make perfect: Effects of task goals on learning computing concepts. *ACM Trans. Comput. Educ.* 11, 4 (Nov. 2011). DOI: https://doi.org/10.1145/2048931.2048933

[112] A. Moffat, B. Hughes, H. Søndergaard, and P. Gruba. 2005. Making connections: First year transition for computer science and software engineering students, Vol. 42. 229–238.

[113] P. Mota, F. S. Melo, and L. Coheur. 2015. Modeling students self-studies behaviors, Vol. 3. 1521–1528.

[114] Matti Nelimarkka and Arto Hellas. 2018. Social help-seeking strategies in a programming MOOC. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE'18).* Association for Computing Machinery, New York, NY, 116–121. DOI: https://doi.org/10.1145/3159450.3159495

[115] Philip M. Newton. 2015. The learning styles myth is thriving in higher education. *Front. Psychol.* 6 (2015). DOI: https://doi.org/10.3389/fpsyg.2015.01908

[116] Philip M. Newton and Atharva Salvi. 2020. How common is belief in the learning styles neuromyth, and does it matter? A pragmatic systematic review. *Front. Educ.* 5 (2020). DOI: https://doi.org/10.3389/feduc.2020.602451

[117] G. S. Ng and E. Y. K. Ng. 1997. Undergraduate students in a computer engineering course: A perspective of their learning approaches and motivation factors. *Innov. Educ. Train. Int.* 34, 1 (Feb. 1997), 65–69. DOI: https://doi.org/{10.1080/1355800970340110}

[118] M. Nkhoma, N. Sriratanaviriyakul, H. P. Cong, and T. K. Lam. 2014. Examining the mediating role of learning engagement, learning process and learning experience on the learning outcomes through localized real case studies. *Educ. Train.* 56, 4 (2014), 287–302. DOI: https://doi.org/10.1108/ET-01-2013-0005

[119] Sarath A. Nonis and Gail I. Hudson. 2006. Academic performance of college students: Influence of time spent studying and working. *J. Educ. Bus.* 81, 3 (Jan. 2006), 151–159. DOI: https://doi.org/10.3200/JOEB.81.3.151-159

[120] Ezekiel Okike. 2014. Investigating' achievements in computing science using human metric. *Int. J. Advanc. Comput. Sci. Applic.* 5, 4 (2014). DOI: https://doi.org/10.14569/IJACSA.2014.050426

[121] C. Ott, A. Robins, P. Haden, and K. Shephard. 2015. Illustrating performance indicators and course characteristics to support students' self-regulated learning in CS1. *Comput. Sci. Educ.* 25, 2 (2015), 174–198. DOI: https://doi.org/10.1080/08993408.2015.1033129

[122] H. Oysaed, R. Andersen, P. E. Garmannslund, A. Prinz, A. Lazareva, and M. Schmidt. 2017. Does the introduction of an overall study strategy empower students to use appropriate study strategies? In *Proceedings of the 10th International Conference of Education, Research, and Innovation* (ICERI'17). 4152–4157.

[123] Harold Pashler, Mark McDaniel, Doug Rohrer, and Robert Bjork. 2008. Learning styles: Concepts and evidence. *Psychol. Sci. Pub. Int.* 9, 3 (Dec. 2008), 105–119. DOI: https://doi.org/10.1111/j.1539-6053.2009.01038.x

[124] Arnold Pears, Philip East, Robert McCartney, Mark Bartley Ratcliffe, Ioanna Stamouli, Anders Berglund, Päivi Kinnunen, Jan-Erik Moström, Carsten Schulte, Anna Eckerdal et al. 2007. What's the problem? Teachers' experience of student learning successes and failures. In *Proceedings of the 7th Baltic Sea Conference on Computing Education Research (Koli Calling'07).* Australian Computer Society, Inc., AUS, 207–211.

[125] Andrew Petersen, Michelle Craig, Jennifer Campbell, and Anya Tafliovich. 2016. Revisiting why students drop CS1. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling'16).* ACM, New York, NY, 71–80. DOI: https://doi.org/10.1145/2999541.2999552

[126] Kerstin Pettersson, Maria Svedin, Max Scheja, and Olle Bälter. 2018. Approaches to studying in first-year engineering: Comparison between inventory scores and students' descriptions of their approaches through interviews. *High. Educ.* 75, 5 (May 2018), 827–838. DOI: https://doi.org/10.1007/s10734-017-0172-7

[127] Maria da Graça C. Pimentel, Cristiane A. Yaguinuma, Diogo S. Martins, and Isabela Zaine. 2019. Anchoring interactive points of interest on web-based instructional video: Effects on students' interaction behavior and perceived experience. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC'19).* Association for Computing Machinery, New York, NY, 2445–2452. DOI: https://doi.org/10.1145/3297280.3297521

[128] E. Ashby Plant, K. Anders Ericsson, Len Hill, and Kia Asberg. 2005. Why study time does not predict grade point average across college students: Implications of deliberate practice for academic performance. *Contemp. Educ. Psychol.* 30, 1 (Jan. 2005), 96–116. DOI: https://doi.org/10.1016/j.cedpsych.2004.06.001

[129] Arthur E. Poropat. 2009. A meta-analysis of the five-factor model of personality and academic performance. *Psychol. Bull.* 135, 2 (Mar. 2009), 322–338. DOI: https://doi.org/10.1037/a0014996

[130] James Prather, Brett A. Becker, Michelle Craig, Paul Denny, Dastyni Loksa, and Lauren Margulieux. 2020. What do we think we think we are doing? Metacognition and self-regulation in programming. In *Proceedings of the ACM Conference on International Computing Education Research (ICER'20).* Association for Computing Machinery, New York, NY, 2–13. DOI: https://doi.org/10.1145/3372782.3406263

[131] James Prather, Raymond Pettit, Kayla McMurry, Alani Peters, John Homer, and Maxine Cohen. 2018. Metacognitive difficulties faced by novice programmers in automated assessment tools. In *Proceedings of the ACM Conference*

on International Computing Education Research (ICER'18). ACM, New York, NY, 41–50. DOI: https://doi.org/10.1145/3230977.3230981

[132] Justus J. Randolph, George Julnes, Erkki Sutinen, and Steve Lehman. 2008. A methodological review of computer science education research. *J. Inf. Technol. Educ.: Res.* 7, 1 (Jan. 2008), 135–162.

[133] John T. E. Richardson. 2015. Approaches to learning or levels of processing: What did Marton and Säljö really say? The legacy of the work of the Göteborg Group in the 1970s. *Interchange* 46, 3 (Aug. 2015), 239–269. DOI: https://doi.org/10.1007/s10780-015-9251-9

[134] Anthony Robins, Janet Rountree, and Nathan Rountree. 2003. Learning and teaching programming: A review and discussion. *Comput. Sci. Educ.* 13, 2 (June 2003), 137–172. DOI: https://doi.org/10.1076/csed.13.2.137.14200

[135] Anna Sandström and Mats Daniels. 2000. Time studies as a tool for (computer science) education research. In *Proceedings of the Australasian Conference on Computing Education (ACSE'00).* ACM, New York, NY, 208–214. DOI: https://doi.org/10.1145/359369.359401

[136] M. Scheidt, R. Senkpeil, J. Chen, A. Godwin, and E. Berger. 2019. SAT does not spell success: How non-cognitive factors can explain variance in the GPA of undergraduate engineering and computer science students, Vol. 2018-October. In *IEEE Frontiers in Education Conference (FIE).* DOI: https://doi.org/10.1109/FIE.2018.8658989

[137] Howard Schuman, Edward Walsh, Camille Olson, and Barbara Etheridge. 1985. Effort and reward: The assumption that college grades are affected by quantity of study. *Soc. Forces* 63, 4 (1985), 945–966. DOI: https://doi.org/10.2307/2578600

[138] Amber Settle, John Lalor, and Theresa Steinbach. 2015. Evaluating a linked-courses learning community for development majors. In *Proceedings of the 16th Annual Conference on Information Technology Education (SIGITE'15).* Association for Computing Machinery, New York, NY, 127–132. DOI: https://doi.org/10.1145/2808006.2808031

[139] Steven C. Shaffer. 2005. Ludwig: An online programming tutoring and assessment system. *SIGCSE Bull.* 37, 2 (June 2005), 56–60. DOI: https://doi.org/10.1145/1083431.1083464

[140] J. Sheard, A. Carbone, D. Chinn, and M. Laakso. 2013. Study habits of CS 1 students: What do they say they do? In *Proceedings of the Learning and Teaching in Computing and Engineering Conference.* 122–129. DOI: https://doi.org/10.1109/LaTiCE.2013.46

[141] Varshita Sher, Marek Hatala, and Dragan Gaševiundefined. 2019. On multi-device use: Using technological modality profiles to explain differences in students' learning. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge (LAK'19).* Association for Computing Machinery, New York, NY, 1–10. DOI: https://doi.org/10.1145/3303772.3303790

[142] Adithya Sheshadri, Niki Gitinabard, Collin F. Lynch, Tiffany Barnes, and Sarah Heckman. 2018. Predicting student performance based on online study habits: A study of blended courses. In *Proceedings of the 11th International Conference on Educational Data Mining (EDM'18).*

[143] Nikki Sigurdson and Andrew Petersen. 2018. An exploration of grit in a CS1 context. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research (Koli Calling'18).* Association for Computing Machinery, New York, NY. DOI: https://doi.org/10.1145/3279720.3279743

[144] Simon, Sally Fincher, Anthony Robins, Bob Baker, Ilona Box, Quintin Cutts, Michael de Raadt, Patricia Haden, John Hamer, Margaret Hamilton, Raymond Lister, Marian Petre, Ken Sutton, Denise Tolhurst, and Jodi Tutty. 2006. Predictors of success in a first programming course. In *Proceedings of the 8th Australasian Conference on Computing Education (ACE'06).* Australian Computer Society, Inc., 189–196.

[145] J. F. Superby, J. P. Vandamme, and N. Meskens. 2006. Determination of factors influencing the achievement of the first-year university students using data mining methods. In *Proceedings of the Workshop on Educational Data Mining.*

[146] F. Sustelo and P. Guerreiro. 2010. Analyzing self-reflection by computer science student to identify bad study habits: Self-reflection performed by students of programming courses on the study habits and skills acquired through b-learning supported by an automatic judge. In *IEEE Educon 2010.* 263–270. DOI: https://doi.org/10.1109/EDUCON.2010.5492568

[147] M. Svedin and O. Bälter. 2016. Gender neutrality improved completion rate for all. *Comput. Sci. Educ.* 26, 2–3 (2016), 192–207. DOI: https://doi.org/10.1080/08993408.2016.1231469

[148] Claudia Szabo, Nickolas Falkner, Andrew Petersen, Heather Bort, Kathryn Cunningham, Peter Donaldson, Arto Hellas, James Robinson, and Judy Sheard. 2019. Review and use of learning theories within computer science education research: Primer for researchers and practitioners. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR'19).* Association for Computing Machinery, New York, NY, 89–109. DOI: https://doi.org/10.1145/3344429.3372504

[149] Neena Thota. 2011. Repertory grid: Investigating personal constructs of novice programmers. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research (Koli calling'11).* Association for Computing Machinery, New York, NY, 23–32. DOI: https://doi.org/10.1145/2094131.2094137

[150] Vincent Tinto. 1988. Stages of student departure: Reflections on the longitudinal character of student leaving. *J. High. Educ.* 59, 4 (1988), 438–455. DOI: https://doi.org/10.2307/1981920

[151] Denise Tolhurst. 2004. The influence of web-supported independent activities and small group work on students' epistemological beliefs. In *Proceedings of the 6th Australasian Conference on Computing Education (ACE'04)*. Australian Computer Society, Inc., AUS, 311–316.

[152] Tara Tressel, Susanne P. Lajoie, and Melissa C. Duffy. 2019. A guide for study terminology: Reviewing a fragmented domain. *Canad. Psychol./Psychologie canadienne* 60, 2 (2019), 115. DOI: https://doi.org/10.1037/cap0000138

[153] Marcel V. J. Veenman, Laura Bavelaar, Levina De Wolf, and Marieke G. P. Van Haaren. 2014. The on-line assessment of metacognitive skills in a computerized learning environment. *Learn. Indiv. Dif.* 29 (Jan. 2014), 123–130. DOI: https://doi.org/10.1016/j.lindif.2013.01.003

[154] Arto Vihavainen, Juha Helminen, and Petri Ihantola. 2014. How novices tackle their first lines of code in an IDE: Analysis of programming session traces. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research (Koli Calling'14)*. Association for Computing Machinery, New York, NY, 109–116. DOI: https://doi.org/10.1145/2674683.2674692

[155] Klinge Orlando Villalba-Condori, Sonia Esther Castro Cuba-Sayco, Evelyn Paola Guillen Chávez, Claudia Deco, and Cristina Bender. 2018. Approaches of learning and computational thinking in students that get into the computer sciences career. In *Proceedings of the 6th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'18)*. Association for Computing Machinery, New York, NY, 36–40. DOI: https://doi.org/10.1145/3284179.3284185

[156] Robert Vinaja. 2014. The use of lecture videos, EBooks, and clickers in computer courses. *J. Comput. Sci. Coll.* 30, 2 (Dec. 2014), 23–32.

[157] H. Wan, K. Liu, Q. Yu, and X. Gao. 2019. Pedagogical intervention practices: Improving learning engagement based on early prediction. *IEEE Trans. Learn. Technol.* 12, 2 (2019), 278–289. DOI: https://doi.org/10.1109/TLT.2019.2911284

[158] Rui Wang, Gabriella Harari, Peilin Hao, Xia Zhou, and Andrew T. Campbell. 2015. SmartGPA: How smartphones can assess and predict academic performance of college students. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'15)*. Association for Computing Machinery, New York, NY, 295–306. DOI: https://doi.org/10.1145/2750858.2804251

[159] F. Weng, F. Cheong, and C. Cheong. 2010. The combined effect of self-efficacy and academic integration on higher education students studying IT majors in Taiwan. *Educ. Inf. Technol.* 15, 4 (2010), 333–353. DOI: https://doi.org/10.1007/s10639-009-9115-y

[160] L. K. Westin and M. Nordstrom. 2004. Teaching OO concepts—A new approach. In *Proceedings of the 34th Annual Frontiers in Education Conference.* F3C–6. DOI: https://doi.org/10.1109/FIE.2004.1408620

[161] Keith J. Whittington, Dianne P. Bills, and Lawrence W. Hill. 2003. Implementation of alternative pacing in an introductory programming sequence. In *Proceedings of the 4th Conference on Information Technology Curriculum (CITC4'03)*. Association for Computing Machinery, New York, NY, 47–53. DOI: https://doi.org/10.1145/947121.947132

[162] Gita C. Williams, Richard Bialac, and Yi Liu. 2006. Using online self-assessment in introductory programming classes. *J. Comput. Sci. Coll.* 22, 2 (Dec. 2006), 115–122.

[163] Laurie Williams, Eric Wiebe, Kai Yang, Miriam Ferzli, and Carol Miller. 2002. In support of pair programming in the introductory computer science course. *Comput. Sci. Educ.* 12, 3 (Sept. 2002), 197–212. DOI: https://doi.org/10.1076/csed.12.3.197.8618

[164] Salla Willman, Rolf Lindén, Erkki Kaila, Teemu Rajala, Mikko-Jussi Laakso, and Tapio Salakoski. 2015. On study habits on an introductory course on programming. *Comput. Sci. Educ.* 25, 3 (July 2015), 276–291. DOI: https://doi.org/10.1080/08993408.2015.1073829

[165] Christopher A. Wolters, Sungjun Won, and Maryam Hussain. 2017. Examining the relations of time management and procrastination within a model of self-regulated learning. *Metacog. Learn.* 12, 3 (Dec. 2017), 381–399. DOI: https://doi.org/10.1007/s11409-017-9174-1

[166] Iman YeckehZaare, Paul Resnick, and Barbara Ericson. 2019. A spaced, interleaved retrieval practice tool that is motivating and effective. In *Proceedings of the ACM Conference on International Computing Education Research (ICER'19)*. Association for Computing Machinery, New York, NY, 71–79. DOI: https://doi.org/10.1145/3291279.3339411

[167] Sang Yim, Christopher Lowrance, and Eric Sturzinger. 2019. Effects of YouTube video as pre-lecture preparation. In *Proceedings of the ASEE Annual Conference & Exposition.* ASEE Conferences. DOI: https://doi.org/10.18260/1-2--32697

[168] Halil Yurdugul and Nihal Menzi Cetin. 2015. Investigation of the relationship between learning process and learning outcomes in e-learning environments. *Euras. J. Educ. Res.* 59 (2015), 57–74. DOI: https://doi.org/{10.14689/ejer.2015.59.4}

[169] D. Zehetmeier, A. Böttcher, A. Brüggemann-Klein, V. Thurner, and K. Schlierkamp. 2015. Finding competence characteristics among first semester students in computer science. In *Proceedings of the IEEE Frontiers in Education Conference (FIE'15)*. 1–9. DOI: https://doi.org/10.1109/FIE.2015.7344201

[170] Y. Zhang, T. Zhang, Y. Jia, J. Sun, F. Xu, and W. Xu. 2017. DataLab: Introducing software engineering thinking into data science education at scale. In *Proceedings of the IEEE/ACM 39th International Conference on Software Engineering:*

Software Engineering Education and Training Track (ICSE-SEET'17). 47–56. DOI: https://doi.org/10.1109/ICSE-SEET.2017.7

[171] H. Zhao, Y. Guo, K. Wang, M. Lu, and X. Yan. 2019. The Effects of ICT Use on Chinese college' study behavior in b-learning evaluating b-learning effectiveness via causal model. In *Proceedings of the 14th International Conference on Computer Science Education (ICCSE'19)*. 103–107. DOI: https://doi.org/10.1109/ICCSE.2019.8845333

[172] Daniel Zingaro, Michelle Craig, Leo Porter, Brett A. Becker, Yingjun Cao, Phill Conrad, Diana Cukierman, Arto Hellas, Dastyni Loksa, and Neena Thota. 2018. Achievement goals in CS1: Replication and extension. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE'18)*. Association for Computing Machinery, New York, NY, 687–692. DOI: https://doi.org/10.1145/3159450.3159452

[173] Daniel Zingaro and Leo Porter. 2016. Impact of student achievement goals on CS1 outcomes. In *Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE'16)*. Association for Computing Machinery, New York, NY, 279–296. DOI: https://doi.org/10.1145/2839509.2844553

[174] Gökhan Özsoy, Aysel Memiş, and Turan Temur. 2009. Metacognition, study habits and attitudes. *Int. Electron. J. Element. Educ.* 2, 1 (2009), 154–166.

Received October 2020; revised June 2021; accepted July 2021

Paper 4

Characteristics of the Student-Driven Learning Environment in Computing Education

Madeleine Lorås, Trond Aalberg

ITiCSE 2021

Authors' contributions: Lorås led the research design, data collection, analysis, and was the main author. Aalberg provided general supervision of the research and the paper writing.

172

Characteristics of the Student-Driven Learning Environment in Computing Education

A case study on the interaction between educational design and study behavior

Madeleine Lorås
Norwegian University of Science and Technology
Trondheim, Norway
madeleine.loras@ntnu.no

Trond Aalberg
Norwegian University of Science and Technology
Trondheim, Norway
trond.aalberg@ntnu.no

ABSTRACT

Important learning happens outside organized lectures and labs, but much of the interaction between these educational design constructs and the study behavior of computing students is unknown. In this study, we follow a group of computing students through their first semester in order to explore these dependencies. Through weekly reports, students tracked their study behaviors in a CS1 course. An exploratory cluster analysis was performed, mapping the students' organization, independent study, planning and priorities, time engagement, and use of different study environments. By comparing these aspects of student behavior to design parameters at both the program and course levels we get a holistic understanding of the student-driven learning environment. The results of this analysis confirm that there are close relationships between the educational design and when, where, and how students study. Three characteristics were identified: the home alone tendency, the executive action factor and the organized activities component. These results were used to outline the room for action, which can support computing educators to identify the adjustable educational design parameters that will most significantly affect the students' study behaviors.

CCS CONCEPTS

• **Social and professional topics → Computing education**; **Computer science education**.

KEYWORDS

computing education, learning environments, study behavior, educational design

ACM Reference Format:
Madeleine Lorås and Trond Aalberg. 2021. Characteristics of the Student-Driven Learning Environment in Computing Education: A case study on the interaction between educational design and study behavior. In *26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2021), June 26-July 1, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3430665.3456310.

ITiCSE 2021, June 26-July 1, 2021, Virtual Event, Germany.

ACM ISBN 978-1-4503-8214-4/21/06.
https://doi.org/10.1145/3430665.3456310

1 INTRODUCTION

Students' individual study behaviors are closely intertwined with the educational design of courses and study programs. Educators can explicitly adjust certain aspects of this ecosystem of learning, but not others. For example, we can change educational design parameters such as the number of courses, learning activities, and assessment regime. On the other hand, students' tacit study behaviors are not as easily altered, especially when, where, and how students study. It is where study behavior meets educational design that we find the *student-driven learning environment*, further defined here as the study activities that students engage in on their own, and the relation of these activities to the organized teaching and learning activities. Recently there has been increased interest in gathering and analyzing behavior data in order to learn about how students study and learn [19], but few contributions have focused on the holistic student experience. Therefore this paper aims to examine what characterizes the student-driven learning environment for first year computing students, and specifically the interaction between educational design parameters (lectures, labs, assignments) and study behavior (when, where and how).

By understanding what drives students' study behavior, educators can implement more effective designs and innovations, and it is essential that the computing discipline be investigated in this manner. Computing education has its own specific challenges along with the general issues highlighted by the learning sciences [1, 21]. From previous research on computing students' study behavior, we know that the classroom experience is not always the central aspect of a student's study day [25]. Instead of lectures and teachers, students tend to rely more on online resources and their own independent work. The behaviors of higher-performing students are characterized by soliciting help, seeking out extra resources, taking extensive course notes [16], starting assignments early, working incrementally [10], attending lectures [5], keeping to an average workweek [30], and applying consistent behaviors throughout the semester [11]. In contrast, lower-performing students are more inclined to memorizing code, getting answers from others without understanding them, not working on assignments post-deadline [16], using the internet, working with others, and relying on tutorials and model solutions [5]. In general, many researchers agree that study behaviors and non-cognitive factors contribute strongly to students' performance and achievement [7, 24, 29].

When it comes to the learning environment, research has found that students benefit from being part of a learning community [4], and that a holistic focus on all aspects of the learning process and environment is valuable for students and educators [27]. There seems to be a strong connection between the way that students study and certain educational design parameters, such as mandatory assignments [13] and individual assignments [12]. For example, assessment practices have been found to drive individual learning even when peer learning is advocated to students [12]. Also, mandatory tutorials have been found to increase submissions and early starts on assignments [30]. The structure and teaching of a course defines the learning environment, and educators should consider the implicit message that these factors convey to students [28].

1.1 Computing Education Design

The current research examines students in the first semester of several similar programs at one specific university. We use the word 'program' to describe the organization of students into a specific field of study, otherwise commonly referred to as major or school. Regardless of how the first year of a computing program is designed, there are some common elements. There will be organized teaching activities, such as lectures and labs, where an educator is present. In addition, there will be some forms of organized learning activity, such as assignments, project work or deliverables, often related to a form of assessment. Students also have access to resources, such as books, websites, teaching assistants (TAs) or other tools, as well as physical areas in which to study and meet peers. The way that students act when preparing for or taking part in these activities constitutes the students' study behavior [29].

General higher education can be viewed at three levels: program, course, and student level. The program level includes courses with specific learning outcomes, learning activities, teaching staff, and assessment methods, as well as overall learning outcomes and goals for the students within the program. The course level includes the teaching and learning activities for a specific course, and the student level includes the student body and student life. The program and course level will have certain design parameters that constitute the educational design as a whole. These parameters pose questions about design aspects that educators must consider. For instance, how many courses there are in a semester (program level), the use of assignments and assessment in a course, and if the course open to all students or reserved for one study program (open or closed enrollment).

2 THE STUDENT-DRIVEN LEARNING ENVIRONMENT

Learning environments are essential to student learning, but they are tricky to define and measure [9, 26]. Educational psychologist John B. Biggs described learning environments in his seminal work on student learning processes in the 1980s. In his 3P model of learning in higher education – presage, process, and product – he describes how "students undertake, or avoid, learning for a variety of reasons; those reasons determine how they go about their learning, and how they go about their learning will determine the quality of the outcome" [3, p.5]. An important part of the presage is the teaching context. In addition to the learning environment, presage includes the curriculum, assessment, and teaching methods. Common for these factors is that the institution controls them, whereas the other aspect of presage, the student characteristics, exists prior to the learning and relates to the student. The final two parts of the model, process and product, are related to the students' approaches to learning and the learning outcome. The current study focuses on one of the presage factors, namely the learning environment. How a student learns is influenced not just by the teaching context but by the student's perceptions of the learning environment [17]. Thus, the quality if the learning can be altered by changing the educational design parameters and importantly the student perceptions of the learning environment [9, 22].

As the 3P model suggests, there will be learning environments present within each course, as well as the at the program level. It is in these interactions that we have the *student-driven learning environment* (SDLE), which is based on the individual students' perspective and describes how they navigate and interact with the educational design constructs across courses within a program. It is student-driven because it is the student who has to navigate between organized activities and independent study, prioritizing and balancing the course load, managing their time, and using physical study spaces. The authors' previous work on the relation between computing students' study behaviors and educational design further divides the SDLE into the following five dimensions [18]:

Table 1: The five dimensions of the SDLE

Dimension	Description
Organization	How students interact with organized learning activities and manage their independent study.
Independent study	What tactics the student employs outside of organized learning activities.
Planning and priorities	Management of the course load.
Time engagement	When the students study: what days and what times of the day.
The study environment	Where the students study.

3 METHODOLOGY

The current study is designed as a case study [6, 31] aimed at describing and explaining aspects of how first-year computing students study. The case can be viewed as the first semester of a computing program, where the phenomenon of *studying* is researched holistically [2] by following a group of students throughout their studies.

To characterize the student-driven learning environment, we need to know what the students do when studying computing, what educational design parameters they interact with, and how this progresses over time. The research involves two main data sources: weekly learning reports handed in by the students along with their assignments, and the educational design parameters in the investigated study programs. It is important to note that the first author was part of the teaching staff, thus gaining essential insight into the educational design; however, that author was not involved in the assessment of the students. Ethical approval was granted by the Norwegian Centre for Research Data (841439).

3.1 The Case

The research was carried out at a large university in Norway during the 2019/20 academic year. Students follow a set plan, taking four equally weighted courses each semester. The courses in the first year vary somewhat from program to program, but all the programs involved in this study included some mathematics courses as well as a course in scientific philosophy in the first semester. Common to all programs are an introductory programming course using Python (CS1).

The current study aimed to investigate the students' journey through their first semester. The students begin the semester in mid-August with a two-week social and academic introduction program. After that, the 'regular' semester lasts for ten weeks, followed by an exam period of four weeks. This study is based on the common introductory programming course, but the research perspective is on the whole semester, including the other courses taken. The course is taught with theoretical, programming, and exercise lectures, as well as weekly assignments. Two of the ten assignments are 'mock exams', where instead of having a whole week to work on it, students must complete the assignment within a two-hour session in an auditorium. The assignments were assessed by TAs on a pass/fail basis but did not count towards the final grade. In order to qualify for the final exam, which accounts for the whole grade, the students must have completed eight out of ten weekly assignments, including at least one mock exam.

The students participating in this study were all enrolled in a computing study program: computing engineering, informatics, technology management, engineering and ICT, communication technology, or teaching and computing. There was a total of 544 students, of whom 203 (37%) consented to take part in the research study. The gender distribution in the course is approximately 70/30 male to female, and in the participation group, 60/40. The students' age and ethnicity were homogeneous, with an average age of 20 and no international students.

3.2 Data Collection and Variables

Along with the weekly assignments, participants handed in a learning report in which they recorded when, where, and how they had worked on the assignments. From these weekly reports, students' study behaviors were modeled and tracked. Organization was measured by students indicating how much time they spent on the following activities: lectures, sessions with TAs, collaboration, working alone in proximity to peers, or working alone. Independent study was measured by students indicating to what extent (very often – never) they used the following tactics: doing the assignment, examples from lectures, reading the book, taking notes, working self-made examples, using the internet, videos, or memorizing. Planning and prioritizing was measured by having students compare how much time they spent on other courses, such as mathematics or scientific philosophy, to their effort in CS1 (a lot more – a lot less). Time engagement was measured by tracking what days (Monday – Sunday) they were working on the assignment, as well as an indication of what times (morning, afternoon, evening, or all day). The study environment was measured by students reporting to what extent (very often – never) they used the following areas to work on the assignment: the open computing labs, the general study areas, the library, the cafeteria, their home, or somewhere else off-campus. The wording of the questions in the learning reports was created by combining various study behavior surveys and questionnaires [14, 15] and revising them for the current educational context.

3.3 Threats to Validity and Limitations

This study is based on self-reported data, which poses a threat to the validity of the research. Students could have been dishonest in their reporting or unmotivated to answer, or they might have had trouble remembering exactly what they did that week. These are always concerns when basing research on surveys and questionnaires; however, efforts were made to ensure that students felt comfortable reporting 'bad' behaviors. They were informed on several occasions that the researcher was not involved in the grading of assignments or the exam and that the reports were confidential. Examining the data, it is clear that many students were not afraid to be honest; however, that does not mean everyone was. On the other hand, the large number of observations (2035 in total) might offset an occasionally flawed report. In addition, there are some limitations to the case study methodology, especially with only one institution being involved [2]. Future research is needed to further explore the results from this study in other educational context.

Table 2: Cluster analysis of the SDLE dimensions

Study behavior	Clusters		
	k	Description	Freq.
Organization	1	Lectures	313
	3	Alone and lectures	286
	2	Alone	265
Independent study	1	Assignment	324
	4	Assignment and lecture examples	190
	5	Assignment and internet	179
	2	Internet and book	90
	3	Assignment and book	81
Planning and priorities	2	Spent more time on mathematics	188
	3	Spent more time on CS1	111
	1	Spent more time on non-CS	60
Time engagement	6	Late, weekends	177
	5	Late, work week	165
	1	Afternoon	148
	3	Late, towards deadline	134
	4	Work week	120
	2	Early	105
Study environment	3	Home	469
	4	Study area	107
	1	Home and lab	196
	2	Lab	92

4 ANALYSIS AND RESULTS

The analysis of the learning reports consists of two parts: a descriptive analysis and a cluster analysis. In order to explore the SDLE, and specifically how study behaviors interact with the educational design parameters, we examine how the five dimensions described in Table 1 developed over the semester. This was done by graphing the various study behavior variables by week. Note that mock exams were in weeks 4 and 8, and that week 11 was the first week of the exam period and had no lectures or assignments.

In addition, we wanted to examine the interconnections between the various elements of the dimensions. A cluster analysis was performed in Stata on the different study behavior variables. K-median clustering with random initial group centers was run until a fitting model was found, exploring the number of clusters from 1-20 as described by Makles [20]. Frequency tables of the best fitting clustering were used to describe the clusters. The results of the clustering analysis are presented in Table 2, sorted by the size of the cluster for each dimension. These clusters depict tendencies in when, where, and how the students study, and will be described in detail in the following subsections, along with the results from the descriptive analysis.

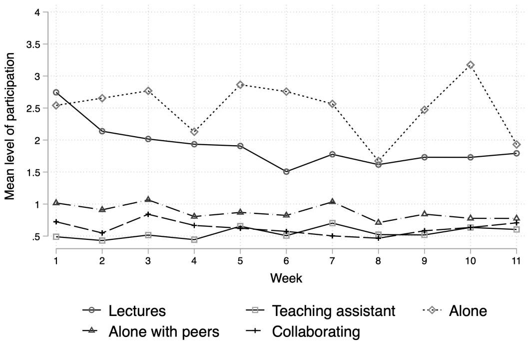


Figure 1: Organization over the first semester

4.1 Organization

For the organization dimension, we see from Figure 1 that time spent in lectures and studying alone are the most predominant characteristics. Lecture participation seems to go down after the first two weeks, while time working alone fluctuates according to the exams in weeks 4, 8 and 11. The remaining parameters, time with TAs, collaboration, and working with other students, were stably low.

The cluster analysis produced three clusters. The first cluster consists of students who spent most time in lectures but also working alone. The third cluster describes students who mostly worked alone, and the second was a combination of working alone and in lectures. All groups spent little time on collaboration, but some time with TAs and other students.

4.2 Independent Study

Three parameters stand out in the examination of independent study as shown in Figure 2: doing the assignments, using the internet, and working on examples from the lectures. Where the first two seem to dip in use in the weeks with exams, the use of lecture examples goes up.

Under independent study, five clusters were formed. Four clusters were related to doing the assignment (1) or doing the assignment along with either reading the book (3), doing lecture examples (4), or using the internet (5). The last cluster (2) was made up of students who preferred using the book and the internet. Common to all clusters was that self-made examples, videos, and memorizing were unpopular tactics.

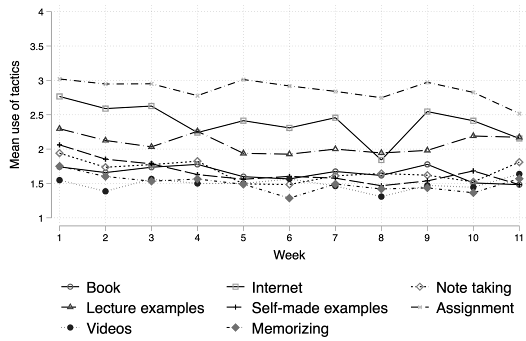


Figure 2: Independent study over the first semester

4.3 Planning and Priorities

The descriptive analysis of the planning and priorities dimension (Figure 3) indicates that mathematics courses have a higher priority than the introductory programming course, while the scientific philosophy course is consistently lower. The cluster analysis further explores this, finding three clusters. Cluster 1 describes students who, in general, spent more or the same time on calculus, discrete mathematics and philosophy, compared to CS1. Students who spent more time on mathematics (both calculus and discrete), but less on philosophy, were placed in the second cluster. The third cluster describes the students who spent the same or less time on all other courses, hence spent the most time on CS1.

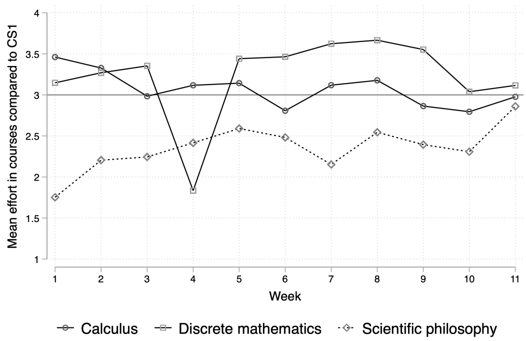
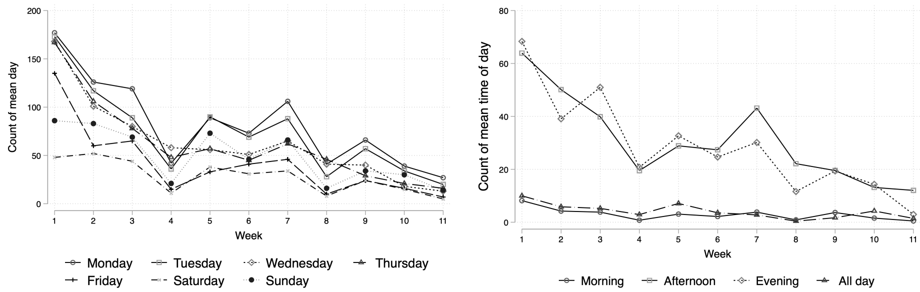


Figure 3: Priorities over the first semester, with CS1 presented as a uniform 3



(a)

(b)

Figure 4: Time engagement over the first semester

4.4 Time engagement

To examine students' time engagement, we look at *what days* they studied as well as what *time segment of the day*. The descriptive results in Figure 4 seem to indicate that the total time use goes down towards the end of the semester; however, this is probably due to using frequency instead of mean. All days of the week seem to be used for studying; however, weekdays are slightly above weekends. Furthermore, students seem to be studying more in the afternoon and evening than during the morning.

The cluster analysis of students' time engagement and found that students can be divided into six clusters. The first two clusters describe students who prefer to study in the afternoon (1) or early in the day (2) but tend to use all days of the week. The third cluster describes students who tend to study late in the day and more on the days before the deadline. Cluster four is for students who study according to a regular workweek. The last two clusters describe students who prefer studying later in the day during either the workweek (5) or the weekend (6).

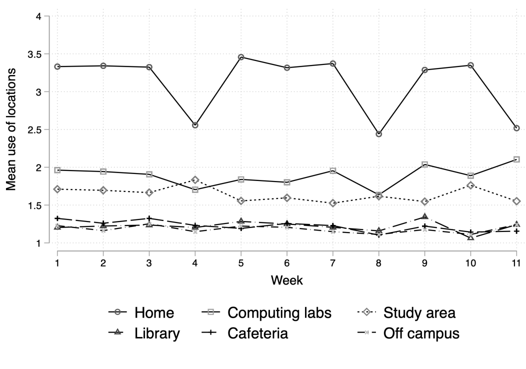


Figure 5: The study environment over the first semester

4.5 The Study Environment

When exploring where students are studying, two characteristics emerge from the descriptive results in Figure 5. The home environment seems to be the preferred place to study for these computing students. Next in line are areas on campus intended for studying: the computing labs or general study areas. Libraries, cafeterias, and off-campus sites were, to a large extent, not used. The cluster analysis found four clusters. Students tended to divide their time between home and the open computing labs (1), or mostly the lab (2), home (3), or the general study area (4). These three locations are popular across the clusters, while the library, cafeteria, and off-campus sites are equally unpopular for all groups.

5 DISCUSSION

Examining the results of the descriptive and cluster analysis collectively, we identify three main findings, which together constitute the characteristics of the SDLE for these computing students. The characteristics must be viewed in relation to the design parameters of the courses and programs in this case.

5.1 The Home Alone Tendency

Looking at organization and the study environment together, there seems to be a strong tendency for computing students to study at home and to study alone. Although we have not checked whether these are the same students, this is still a striking tendency. Previous research on the effect of the study environment is not clear on whether the home is an advantageous place to study; however, some studies have shown that studying in peace and quiet is preferred by most students [23]. On the other hand, we know that learning computing is a collaborative process and that students benefit from learning communities [4]. Another concern regarding the home alone tendency is that access to help and support is valuable [16], and for these computing students, help is found mainly on campus.

Possible explanations for the home alone tendency can be found in the educational design parameters. During this semester, all assignments in CS1 were individual, and very few of the other courses employed any form of collaborative activity. Furthermore, the computing labs and the general study areas on campus are known to be crowded. It can often be difficult to find a place to study, especially as these are new students.

5.2 The Executive Action Factor

When students manage their time and handle their course load, they are constantly making executive decisions, although in many cases these might be more reactive than proactive. This group of computing students seems to have a preference to avoid working in the morning and on weekends; no other clear trends can be found. Previous work has found that high performing students are likely to follow a regular workweek and not to work at nights and on weekends [30].

Considering the balance and priorities, the results suggest that mathematics was largely prioritized above CS1. It is important to note that all courses had equal credit, and that participation in lectures and labs was not mandatory and did not count towards the grade in any way. It is reasonable to assume that the students' executive actions would be affected if one or more courses implemented mandatory participation, perhaps guiding all students towards a more structured study week [8].

5.3 The Organized Activities Component

The results further indicate that the learning activities, in this case assignments, were even more of a driving factor for student behavior than lectures. The analysis of how students study independently shows a clear assignment-based approach, which is not unexpected [25]. The assignments are the backbone of this course, and when learning programming, it has been established many times that students must do programming in order to master it. This perception that the assignments drive student behavior is in line with previous research [13]. Across all behavior dimensions, it is evident that students study differently during weeks where there are mock exams (4 and 8) or after the assignments are finished (11). This indicates that the way students are assessed largely impacts when, where, and how they study. During mock exams, they spend less time at home and more in the study areas, use the internet less, and focus more on lecture examples, note-taking, and reading the book. Similarly, during the exam preparation week they spend more time on campus, memorizing and note-taking more, and make more use of videos.

5.4 Implications

The current study represents one case at one institution with one set of design parameters, but it does offer some generalizable features and areas for future research. First, we must consider the room for action within the SDLE, that is, what we can and cannot change. One dimension for consideration is time: what can be changed quickly and with short term effects, and what is more of a long-term change? All parameters at the program level are long-term because there are many other stakeholders involved, which brings us to the next dimension: control. The question of who controls the various parameters and can make decisions and implement change varies from institution to institution. Most parameters are managed by the responsible professor and are, therefore, department controlled at the course level. On the other hand, campus layout, scheduling, and semester design are controlled by the institution. Finally there is the dimension of resources: time and finances. The best example of this is in the course dimension, where making changes to the learning activities and assessment will often imply more educators or increased time commitments from the existing educators.

The role of the current study is to help computing educators find the educational design parameters that can be changed and that have the greatest impact on the students' study behavior. Considering the dimensions of the room for action and the findings from this study, we have some examples of changes to the design parameters that should be considered and researched further:

- Increased use of group activities is a learning activity that will decrease the time students spend alone, and perhaps encourage more time on campus. This is a short-term, low-resource change that can be implemented by the educator, with a potentially high impact.
- Scheduling of lectures and lab in a more coherent and holistic manner across courses can help students structure their studies better. This is a long-term, low-resource change at the program level, with potentially high impact.
- Changing the assignment structure or including the assignments in the formal assessment will change the students' time use and activity planning. This is a short-term, medium-resource change at the program level with potentially high impact.

6 CONCLUSIONS AND FUTURE WORK

This paper has investigated the relationship between study behavior and educational design parameters encompassed in the student-driven learning environment. By examining weekly reports from the student participants, we have found close relationships between the educational design and when, where, and how students study. Results from a cluster analysis indicate that students are studying all days of the week, and mostly later in the day. This might indicate that students are working evenly, but it might also be a symptom of a heavy timetable and fragmented study behavior. Furthermore, a majority of the students tend to prefer working from home, or to a lesser extent using the computing labs or study areas. Exactly what drives these choices, beyond the assignment structure, is not clear from this data and should be a question for future research.

For the programs studied in this research the educational design scope and course structure are strictly controlled at the department level or above. Nevertheless, the dimensions of the SDLE applied in this study can serve as a tool for other researchers and educators, and can help to identify the local room for action. Computing education is experiencing a surge of students while at the same time being urged to increase throughput without additional resources. Understanding when, where, and how first-year computing students are learning can inform educational design decisions and provide insight for innovations.

ACKNOWLEDGMENTS

The work in this paper was conducted at Excited Centre for Excellence, publicly funded through DIKU. We would like to especially acknowledge Simon for valuable feedback and editing, and Kshitij Sharma for input on the cluster analysis.

REFERENCES

[1] Vicki L. Almstrum, Orit Hazzan, Mark Guzdial, and Marian Petre. 2005. Challenges to Computer Science Education Research. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '05)*. ACM, New York, NY, USA, 191–192. https://doi.org/10.1145/1047344.1047415

[2] Michael Bassey. 1999. *Case Study Research In Educational Settings*. Open University Press, Buckingham.

[3] John B. Biggs. 1989. Approaches to the Enhancement of Tertiary Teaching. *Higher Education Research & Development* 8, 1 (Jan. 1989), 7–25. https://doi.org/10.1080/0729436890080102

[4] Candido Cabo and Ashwin Satyanarayana. 2018. Promoting students' social interactions results in an improvement in performance, class attendance and retention in first year computing courses. In *2018 IEEE FRONTIERS IN EDUCATION CONFERENCE (FIE) (Frontiers in education conference)*. 1–8. https://doi.org/10.1109/FIE.2018.8658548

[5] Donald Chinn, Judy Sheard, Angela Carbone, and Mikko-Jussi Laakso. 2010. Study Habits of CS1 Students: What Do They Do Outside the Classroom?. In *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103 (ACE '10)*. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 53–62.

[6] Louis Cohen, Lawrence Manion, and Keith Morrison. 2002. *Research Methods in Education* (5 ed.). Routledge, London, UK.

[7] Marcus Credé and Nathan R. Kuncel. 2008. Study Habits, Skills, and Attitudes: The Third Pillar Supporting Collegiate Academic Performance. *Perspectives on Psychological Science* 3, 6 (Nov. 2008), 425–453. https://doi.org/10.1111/j.1745-6924.2008.00089.x

[8] Marcus Credé, Sylvia G. Roch, and Urszula M. Kieszczynka. 2010. Class Attendance in College: A Meta-Analytic Review of the Relationship of Class Attendance With Grades and Student Characteristics. *Review of Educational Research* 80, 2 (June 2010), 272–295. https://doi.org/10.3102/0034654310362998

[9] Noel J. Entwistle. 1991. Approaches to Learning and Perceptions of the Learning Environment: Introduction to the Special Issue. *Higher Education* 22, 3 (1991), 201–4.

[10] James B. Fenwick, Jr., Cindy Norris, Frank E. Barry, Josh Rountree, Cole J. Spicer, and Scott D. Cheek. 2009. Another Look at the Behaviors of Novice Programmers. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE '09)*. ACM, New York, NY, USA, 296–300. https://doi.org/10.1145/1508865.1508973

[11] Seth Copen Goldstein, Hongyi Zhang, Majd Sakr, Haokang An, and Cameron Dashti. 2019. Understanding How Work Habits Influence Student Performance. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '19)*. ACM, New York, NY, USA, 154–160. https://doi.org/10.1145/3304221.3319757

[12] Sheryl Guloy, Farimah Salimi, Diana Cukierman, and Donna McGee Thompson. 2017. Insights on supporting learning during computing science and engineering students' transition to university: a design-oriented, mixed methods exploration of instructor and student perspectives. *Higher Education* 73, 3 (March 2017), 479–497. https://doi.org/10.1007/s10734-016-0097-6

[13] Vegard Hellem and Madeleine Lorås. 2020. The Effect of Mandatory Assignments on Students Learning Outcome and Performance in Introductory Programming Courses. In *2020 IEEE Global Engineering Education Conference (EDUCON)*. 704–712. https://doi.org/10.1109/EDUCON45650.2020.9125198

[14] Wayne H. Holtzman, William F. Brown, and W.G. Farquhar. 1954. The Survey of Study Habits and Attitudes: a New Instrument for the Prediction of Academic Success. *Educational and Psychological Measurement* 14, 4 (Dec. 1954), 726–732. https://doi.org/10.1177/001316445401400420

[15] Caesar R. Jackson. 2018. Validating and Adapting the Motivated Strategies for Learning Questionnaire (MSLQ) for STEM Courses at an HBCU. *AERA Open* 4, 4 (Oct. 2018), 2332858418809346. https://doi.org/10.1177/2332858418809346

[16] Soohyun Nam Liao, Sander Valstar, Kevin Thai, Christine Alvarado, Daniel Zingaro, William G. Griswold, and Leo Porter. 2019. Behaviors of higher and lower performing students in CS1. In *Proceedings of the 2019 ACM conference on innovation and technology in computer science education (ITiCSE '19)*. Association for Computing Machinery, New York, NY, USA, 196–202. https://doi.org/10.1145/3304221.3319740

[17] Alf Lizzio, Keithia Wilson, and Roland Simons. 2002. University Students' Perceptions of the Learning Environment and Academic Outcomes: Implications for theory and practice. *Studies in Higher Education* 27, 1 (Feb. 2002), 27–52. https://doi.org/10.1080/03075070120099359

[18] M. Lorås and T. Aalberg. 2020. First Year Computing Study Behavior: Effects of Educational Design. In *2020 IEEE Frontiers in Education Conference (FIE)*. 1–9. https://doi.org/10.1109/FIE44824.2020.9274285

[19] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 2018. A Review of Introductory Programming Research 2003–2017. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018)*. ACM, New York, NY, USA, 342–343. https://doi.org/10.1145/3197091.3205841

[20] Anna Makles. 2012. Stata Tip 110: How to Get the Optimal K-Means Cluster Solution. *The Stata Journal* 12, 2 (June 2012), 347–351. https://doi.org/10.1177/1536867X1201200213

[21] Lauren E. Margulieux, Brian Dorn, and Kristin A. Searle. 2019. Learning Sciences for Computing Education. In *The Cambridge Handbook of Computing Education Research*, Sally A. Fincher and Anthony V.Editors Robins (Eds.). Cambridge University Press, 208–230. https://doi.org/10.1017/9781108654555.009

[22] J. H. F. Meyer and M. W. Muller. 1990. Evaluating the quality of student learning. I—An unfolding analysis of the association between perceptions of learning context and approaches to studying at an individual level. *Studies in Higher Education* 15, 2 (Jan. 1990), 131–154. https://doi.org/10.1080/03075079012331377471

[23] E. Ashby Plant, K. Anders Ericsson, Len Hill, and Kia Asberg. 2005. Why study time does not predict grade point average across college students: Implications of deliberate practice for academic performance. *Contemporary Educational Psychology* 30, 1 (Jan. 2005), 96–116. https://doi.org/10.1016/j.cedpsych.2004.06.001

[24] Matthew Scheidt, Ryan Senkpeil, John Chen, Allison Godwin, and Edward Berger. 2019. SAT Does Not Spell Success: How Non-Cognitive Factors Can Explain Variance in the GPA of Undergraduate Engineering and Computer Science Students, Vol. 2018-October. https://doi.org/10.1109/FIE.2018.8658989

[25] Judy Sheard, Angela Carbone, Donald Chinn, and Mikko-Jussi Laakso. 2013. Study Habits of CS 1 Students: What Do They Say They Do?. In *2013 Learning and Teaching in Computing and Engineering*. 122–129. https://doi.org/10.1109/LaTiCE.2013.46

[26] Katrien Struyven, Filip Dochy, Steven Janssens, and Sarah Gielen. 2006. On the dynamics of students' approaches to learning: The effects of the teaching/learning environment. *Learning and Instruction* 16, 4 (Aug. 2006), 279–294. https://doi.org/10.1016/j.learninstruc.2006.07.001

[27] Neena Thota. 2011. Repertory grid: Investigating personal constructs of novice programmers. In *Proceedings of the 11th koli calling international conference on computing education research (Koli calling '11)*. Association for Computing Machinery, New York, NY, USA, 23–32. https://doi.org/10.1145/2094131.2094137

[28] Denise Tolhurst. 2004. The influence of web-supported independent activities and small group work on students' epistemological beliefs. In *Proceedings of the sixth australasian conference on computing education - volume 30 (ACE '04)*. Australian Computer Society, Inc., AUS, 311–316.

[29] Tara Tressel, Susanne P. Lajoie, and Melissa C. Duffy. 2019. A guide for study terminology: Reviewing a fragmented domain. *Canadian Psychology/Psychologie canadienne* 60, 2 (2019), 115. https://doi.org/10.1037/cap0000138

[30] Salla Willman, Rolf Lindén, Erkki Kaila, Teemu Rajala, Mikko-Jussi Laakso, and Tapio Salakoski. 2015. On study habits on an introductory course on programming. *Computer Science Education* 25, 3 (July 2015), 276–291. https://doi.org/10.1080/08993408.2015.1073829

[31] Robert K. Yin. 2009. *Case Study Research: Design and Methods* (4 ed.). Applied social research methods, Vol. 5. SAGE Publications, California.

Paper 5

The Importance of the Campus - A Study on the Effects of the Covid-19 Pandemic in a CS2 Course

Madeleine Lorås, Børge Haugseth, Hallvard Trætteberg

EDUCON 2021

Authors' contributions: Lorås led the research design, data collection, analysis, and was the main author. Haugseth and Trætteberg provided access to the course investigated and participated in the discussion of the results.

 182

The Importance of the Campus - A Study on the Effects of the COVID-19 Pandemic in a CS2 Course

Madeleine Lorås, Børge Haugset, Hallvard Trætteberg
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
madeleine.loras@ntnu.no

Abstract—**The educational context for students and educators across the world changed when the COVID-19 pandemic forced most educational institutions to shut down all on-campus activities in the spring of 2020. In this paper, we explore how the study behaviors of first-year computing students in a large scale CS2 course were affected by the rapid change from campus-based to online learning. This research aims to evaluate the effect of moving to an online-only mode of studying and learning, and consequently gaining insight into the role of the physical campus in computing education. A mixed-method research approach was taken to reach these goals by combining interaction tracking data with weekly student reports and interviews. Results indicate that campus-based activities provide essential scaffolding for students' study behaviors, specifically time management and organization. Additionally, the physical study environment provided an informal space for social and academic interactions not found in the online sphere. Furthermore, when moving to the online study environment, students struggled with adapting their study behaviors, spending less time on organized activities and not changing their independent habits. Lastly, the online environment seemed to create considerable differences between those who mastered studying and those who did not, generating a larger ability gap than on campus. In the paper, we provide further descriptions of these findings and some recommendations for computing educators facing similar challenges.**

Index Terms—**Computer Science Education, Computing Education, Higher Education, Study Behavior, CS2, Educational Design, Online Learning, Remote Learning, Study Environments**

I. Introduction

What happens to the students when all physical interaction suddenly disappears overnight? This is indeed a strange question to ask, or would have been at the beginning of 2020. However, we currently live in a world where almost all higher education institutions have had to close down all face-to-face teaching at some point. When the COVID-19 pandemic took hold of the world in the first part of 2020, institutions across the globe had to go from campus-based education to online education. Online education has been around for a while; however, for many educators, students, and administrators, this was a whole new situation.

When the pandemic hit Norway, all campus-based education was shut down on March 12th. Universities and schools across the country were given the order to transfer into the online setting and complete the semester digitally as best we could. In this paper, we will take a close look at how a classic campus-based CS2 programming course dealt with this sudden change. The focus of this study is on the student perspective and how they experienced the change from campus-based to online learning.

Soon after it became clear that we had to go online for the foreseeable future, the authors of this paper began collecting data and identifying ways to learn from this situation. The goal of the current study was twofold. On the one side, we were interested in evaluating the effect of going online on the student experience since we are looking at a minimum of one more year with very limited use of the campus. On the other hand, this unfortunate situation provides an interesting natural experiment on what happens without a campus. In other words, what is the importance of the physical learning environment for the students learning experience? This last part can enlighten our understanding of what aspects of the traditional campus-based design are most important to the students learning, what they can 'live without,' and where we can adjust and improve. In this paper, we aim to answer the following research questions:

- RQ1: How did the students interact with the changes made to the educational design due to COVID-19?
- RQ2: How did the students' study behavior change when going from a campus-based to an online study environment?

The paper is structured as follows. Section II describes the course design before and after the pandemic forced an online transformation. Section III describes the methodology, data collection, and data sources, while Section IV presents analysis and results. Lastly, in Sections V and VI, we discuss the results, as well as reflect on lessons learned. This contribution provides an illustration of how a course was fully digitized within a short time frame and explores the effects of these changes on the student experience which can inform our educational designs coming back to the campus.

A. Study Behavior of Computing Students

This study focuses on the student experience in the light of study behavior development through the abrupt transition from campus-based to online learning due to the pandemic. Therefore, it is necessary to clarify some concepts and theories regarding study behavior in computing education before moving on to describing the study and educational context further. Study behavior is a complex concept and has seen many definitions and terms over the years, both in general education research and in computing education. In this study, we understand study behavior in the broadest sense, relating to any actions students take when preparing for, or taking part in, study-based activities, based on the definition in Tressel, Lajoie and Duffy's review in 2019 [1]. This definition includes the students' interaction with organized learning activities (i.e., lectures, labs, assignments) and how they do their independent studies (i.e., time management, revision, strategies, attitudes). The current study is based on previous work by the authors on computing students' behaviors, specifically, and the interaction with educational design [2]. In this framework, students' study behaviors are divided into the following five dimensions, which will be used for the analysis:

- **Organization**: How students interact with organized learning activities and manage their independent study
- **Independent study**: What tactics the student employs outside of organized learning activities
- **Planning and priorities**: Management of the course load
- **Time management**: When the students study: what days and what times of the day
- **The study environment**: Where the students study

Previous research on the study behavior of computing students' relevant to the current study has suggested that meaningful learning happens during students' independent study [3], [4], and that organized activities in the classroom does not seem to be the primary driver of learning [5]. In general, study behaviors have been found to affect academic performance and learning significantly [6]. How students do their assignments and to what extent they learn from such assignments has also been investigated, finding that assignments help students structure their studies and ensures progression [3], [7]. On the other hand, when discussing the independent study, procrastination is an issue several studies have investigated, finding that it indeed is an issue for computing students and very often leads to decreased academic performance [8], [9]. Most educators and researchers within computing education agree that in order to master any computing concept, students must learn by doing [10], [11]. Moreover, in the online environment investigated in the current study, the students are required to manage this learning and doing alone. Therefore, the current study of students' study behaviors in an introductory computing course comparing on-campus and online behaviors is important to the community.

II. The CS2 Course

The research presented in this paper is based in an undergraduate object-oriented programming (OOP) course at a large university in Norway. The course yields 7.5 ECTS and goes over 14 weeks with a final four-hour exam, in the end, accounting for the whole grade. For the first nine weeks, the course was campus-based, as described in Section II-A, while for the last five weeks, the course was online (Section II-B). It is relevant to mention that two-thirds of the way through the semester, there was a two-week break for Easter, with no scheduled teaching and learning activities. The course has one professor, three head teaching assistants (graduate students), and 40 teaching assistants (undergraduate students). The programming language used is Java, and the course covers topics such as classes and objects, encapsulation, object structures, exception handling, and inheritance. Students generally take this course in their second semester and are required to have completed an introduction to information technology course, which includes programming in Python. The course is mandatory for all the various computer science and computer engineering programs and serves as an elective course for many other engineering programs.

A. The Campus Environment

In the following sections, we will describe the CS2 course design in the campus-based environment before the pandemic caused an online transformation. This course design described below has been in place for eight years, with revisions to the assignments being made regularly. The student feedback is generally positive; however, the workload has been criticized somewhat. On the other hand, students report that the amount of practice and experience with programming in the course is very useful.

1) Tools and Communication: The course uses the learning management system Blackboard (BB) to host all communication and information. Teachers use BB for announcements, sharing slides and resources, and organizing assignments. In addition to BB, the course used Piazza to host discussions and answer questions. The Piazza platform allowed students, teaching assistants, and faculty to interact with each other and has options for anonymity.

2) Lectures and Labs: During the semester, there are four hours of topic lectures and two hours of exercise lectures a week. The topic lectures are given by faculty and cover theoretical perspectives as well as practical examples. The exercise lectures are given by the head teaching assistants and focus on the assignment given that week. The exercise lectures introduce the assignments, give tips on relevant techniques for the upcoming assignments, and go through solutions for previous assignments.

In addition to lectures, there are open labs where students can get help. These labs are staffed with teaching assistants and are open from 0800-1800 every weekday. Students who need help or have questions can drop by at any time; however, each student is placed in a group with a designated teaching assistant (TA). With this designated TA, they will be prioritized in the event of queues. Although this system seems complicated, it has proved to be an effective system for maximizing the chance that students will get help when they need it and utilizing all TAs. The course also has a course wiki page with content about OOP and Java.

3) Assignments and Support: There were ten mandatory assignments in total. They did not count towards the final grade; however, each assignment was awarded points between 50-100, and to qualify for the exam, the student had to reach a total of 750 points. The assignments were based on the curriculum for the current week and the week before. Automatic tests are integrated with the assignments so that both students and teaching assistants (TAs) can quickly check the code. To pass the tests, students have to code correctly for all edge cases, as well as name their methods according to the task description. All students must hand in the assignment individually; however, collaboration is allowed as long as it is labeled.

The assignments are delivered online but have to be demonstrated in-person to their designated TA within a week after the deadline. TAs are generally older students who have completed the course, hired by the department to give feedback, help students with their assignments, and assign points to each assignment. Each TA is responsible for 20 students and is available in the open labs for at least six hours every week.

4) Exam and Assessment: The course grade is based on a final exam. The exam lasts four hours and is given in a secure online assessment platform under supervision. This system allows students to write their code with syntax highlighting but does not provide any other integrated development environment (IDE) features, including compiling. Over the last five years, the average grade for the exam has been a C, and the failure rate has been between 16-23%. If students fail the exam, they have the opportunity to retake the exam at the end of the summer, before the next semester begins.

B. The Online Environment

When the government ordered a total shutdown of all physical interaction on campus, the course had to go digital and create an online environment for remote learning. Table I outlines the changes made to the educational design, which will be further described in the following subsections.

1) Tools and Communications: In addition to the already existing BB and Piazza sites, the course administrators (lecturers and head TAs) also opened a Microsoft Teams site for the course. The goal of this Teams site was to ease the interaction between lecturers, TAs, and students. It was an important consideration to only use tools that were accessible, secure, and in line with privacy rules (GDPR). Since the university uses Microsoft products, Teams was available for all staff and students, and the required security and privacy requirements had already been vetted and cleared.

On March 13th (the day after the announcement), all students were invited to join the new course Teams site. The Teams site had six channels: General (announcements and general remarks), Lectures (links to video lectures), Lectures – Q&A (questions about the lectures), Exercise lectures (links to video lectures), Exercise lectures – Q&A (questions about the lectures) and Support (see Section II-B3). The reason for having separate channels for lectures and Q&A was so the lecture links did not drown in questions and would remain easy to find for the students.

For the remaining five weeks of the semester, all essential information would be given on BB, while the Teams site was used as an additional recourse. Video lecture links were posted on both sites. The Piazza forums remained in use.

2) Lectures and Labs: The lecturer and head TAs started producing video versions of their lectures soon after the initial setup. They decided to go for an asynchronous design, where the video lectures would be posted as soon as they were done, and students were free to watch them in their own time. The lectures were grouped by topic, which in turn, related to an assignment. The videos were posted on the university platform for video sharing, which during the time period changed from Mediasite to Panapto.

At this point, a second lecturer, who had taught the course for several years previously, was recruited to help with the course. The two lecturers would set up the lecture as a conversation, where one would do the coding while sharing his screen. While coding, one instructor would tell the other what he was doing and why, while the other would comment and ask questions. This setup aimed to simulate a more interactive setting, and both lecturers remarked how they enjoyed doing the videos together in this way, as opposed to just filming themselves alone. The head TAs chose the same setup for their exercise lectures.

When it comes to the open lab set up on campus, this was directly transferred to Teams. The TAs would work the same hours in the digital lab as they had in the physical lab, answer questions, support students, and follow up with their designated students.

3) Assignments and Support: The remaining four assignments went as planned, although Assignment 6, which had a deadline on March 13th, was given a one-week extension. The students were still required to hand in their code on BB and demonstrate their work to their TA via video chat in Teams. Each TA was given the task to create a private channel in Teams for his/her students to arrange these demonstrations.

In addition, the TAs were required to be available the digital lab during their normal work hours. This digital lab was accessed through the "Support" channel in Teams. A student in need of support would post "I need help" in the channel, and the next available TA would call them up via video chat. In order to keep track of who was getting help, the TA would like the post to indicate it was taken care of.

4) Exam and Assessment: Pretty soon after the online transformation, both students, educators, and administrators started thinking about the exams. The university soon announced that all traditional exams were canceled and needed to be either oral (via video call) or a home exam. In addition, all course teachers could, if they wanted, change the grading system to pass/fail. This course decided to keep the grading scheme and do a four-hour home exam. This decision was discussed extensively internally and with the students, causing quite a debate. Many considered the pass/fail option as more gentle on the students considering the situation they were in, as well as easier to administer, control, and grade fairly. On the other side, many viewed the grades as important motivators for the students to learn and were concerned that students who had put in the effort so far would not be rewarded the good grade they deserved.

TABLE I: Overview of course design in the campus based (pre pandemic) and online environment (post pandemic).

	Design parameter	The campus environment	The online environment
Course structure	Open or closed enrollment	Open for all students at university	Open for all students at university
	Number of students	841	841
	Class schedule	4 hours lecturing a week	Asynchronous video lectures of remaining topics
		2 hours exercise lecturing a week	Asynchronous video lectures of remaining topics
		Open labs on campus all week (08-18)	Open labs in Teams at the same times
	Mandatory attendance	No	No
Learning activities	Individual or group-based activities	Individual, but collaboration is allowed	Individual, but collaboration is allowed
	Number of assignments and/or projects	Weekly/biweekly mandatory assignments	Weekly/biweekly mandatory assignments
	Learning management system etc.	Blackboard, Piazza	Blackboard, Piazza, Microsoft Teams
	Available resources	TAs in open labs	TAs available on Teams
Educators	Number of lecturers	1	2
	Lecturer-student contact	Mainly through lectures	Mainly on Piazza
	Number of TAs	2 Head TAs	2 Head TAs
		1 TA per 20 students	1 TA per 20 students
Assessment	Type of assessment and exams	End of semester school exam accounts for the whole grade	End of semester home exam accounts for the whole grade

III. Methodology

The rapid change from campus-based to online education provides a natural, although, unplanned experiment. In this study, a class of 841 CS2 students spent the first eight weeks of the semester following a traditional campus-based course. In week 9, the course was changed to be all online. As this was not planned, we do not have all the data points one would expect from an experimental study; however, we do have some data from before and after the intervention, as well as post-intervention data [12], [13]. In general, this study's research design can be viewed as a mixed-method quasi-experimental empirical investigation of a course [14].

A. Data Collection

The data collected in this study comes from three data sources: learning reports, tracking of interaction, and interviews. The learning reports were a mandatory part of each assignment where students were required to self-assess through reporting when, where, and how they had worked on the assignment. These learning reports provide insight into the students' study behaviors, in this case, both before and after the transition to online learning. In addition to the pedagogical benefits of self-reflection, these reports are a part of ongoing research on study behavior; hence, the students have provided consent to use their data for research purposes.

The second data point is the tracking of interaction with the various digital platforms. We were able to track the students' engagement in Piazza both before and after the online transformation. Additionally, we tracked the students' interaction in Teams and views of the video lectures. As this data was not connected to the individual student, but a count of the frequency of use, the need for informed consent is void. BB was not included in the tracking data because the students did not interact with BB outside of submitting assignments. Since the assignments were mandatory, there was no change to BB's interaction patterns throughout the semester.

Lastly, the researchers were able to conduct interviews with seven students after the transformation. Four of the interviews were done via written chat in Teams, while three were done over video chat. The students could choose which medium they preferred. The audio from the video chats was transcribed and added to the written logs. All interviews were directed by an interview guide, created by the authors based on findings from a preliminary survey among students and educators in the first weeks after the intervention [15]. The text from the interviews was merged and coded into the categories used in this analysis.

B. Participants and Considerations

The students participating in this study were all enrolled in a computing study program: computing engineering, informatics, technology management, engineering and ICT, communication technology, or teaching and computing. The gender distribution in the course is approximately 70/30 male to female. The students' ages and nationalities were homogeneous, with an average age of 20 and no international students. Among the 841 enrolled students, 452 consented to use their learning report data for research purposes (54%). We did not gather gender data for the learning reports; however, there is no reason to believe the gender distribution should be any different from the course. Four of the students participating in the interviews were female, and three were male.

All participants were granted informed consent for the collection of learning report data and the use of interview transcriptions for research purposes. The Norwegian Centre for Research Data has approved this. It is important to state that the first author of this paper was not involved in the planning or implementation of the course but was granted access to all the tools and platforms. This independent person handled the data collection and analysis, and the course teachers (remaining authors) were only involved in the discussion of results.

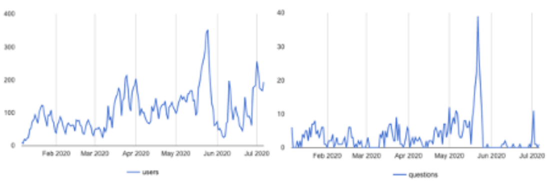
IV. Analysis and Results

In order to answer the first question of how students interacted with the changes to the educational design, we present the tracking and interview data. For the second question, regarding the change in the students' behavior in the online learning environment, we additionally present the data from the learning reports.

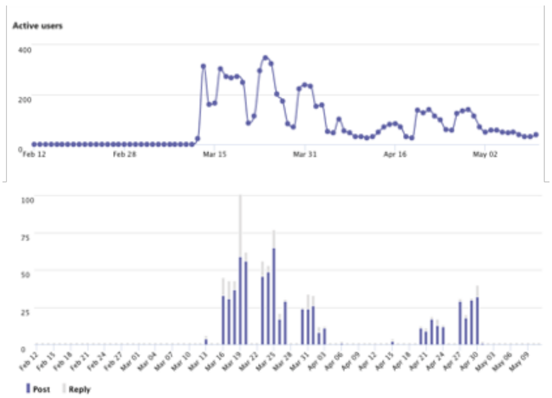
A. Student Interaction

Using the same division as in the previous sections, we will, in the following, present the relevant results for each aspect of the course in addition to describing the method of data extraction and analysis.

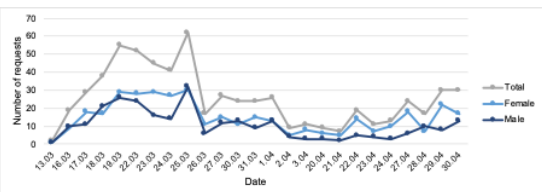
1) Tools and Communications: From the Piazza platform, we were able to extract data on the number of engaged users per day, as well as the number of questions posted. These data can be viewed in Fig. 1a, from the beginning of the semester (late January) until the results of the exam were published in July. It is evident that after March 12th, there was a slight increase in the number of engaged users; however, the number of posts does not seem to see the same steady surge. There is, however, a large peak right around the exam (May 25th). From examining the posts' content, it is clear that there were many questions posted in the days before the exam, as well as several comments on the exam after the fact. In total, 794 students enrolled, and 226 students made a sum of 2402 contributions.



(a) Piazza usage



(b) Teams usage



(c) Teams support interaction

Fig. 1: Study behaviors over the first semester.

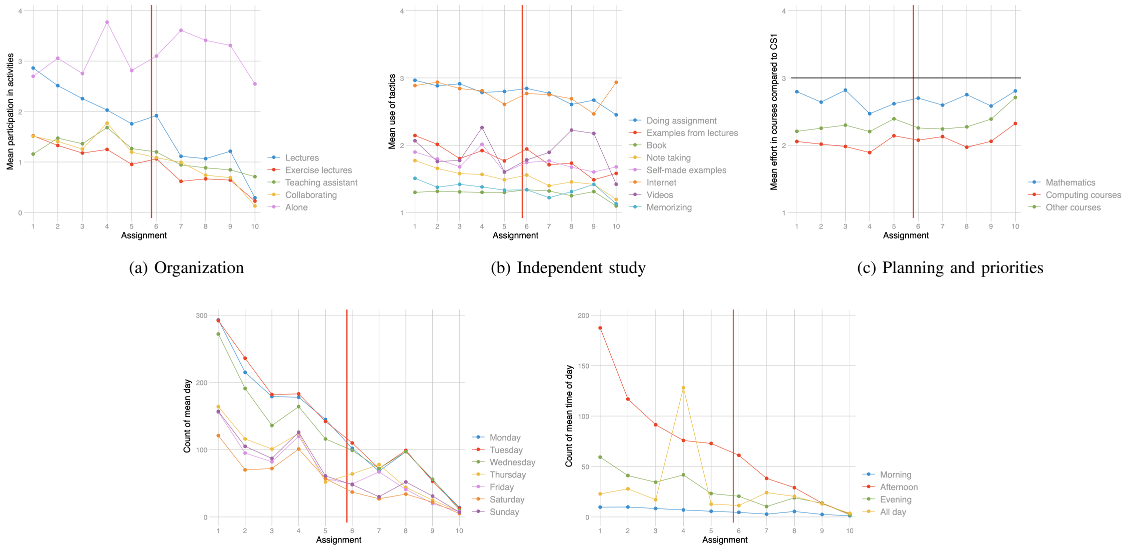
We have similar data for the Teams platform; however, only for the period after March 12th. Fig. 1b depicts the number of engaged users and posts from February to the beginning of May. In contrast to Piazza, the Teams' engagement shut down after lectures and assignments had ended, indicating that no exam preparation or commentary happened on Teams. Similar to the activity on Piazza, the number of active users and posts grew in the immediate aftermath, decreased towards the Easter break, and then grew slightly again towards the end of the lectures (end of April). In total, 931 students and TAs engaged, and 694 contributions were made. Unfortunately, Teams does not allow us to differentiate between students, TAs, and teachers, so we have no way of systematically identifying who made these posts.

Data from the interviews revealed that the students, in general, were content with the tools and communication used in the online setting. Interestingly, many of the students said they did not participate in the discussions or ask many questions but learned a lot from reading through what others wrote. Several students commented on the fact that the number of tools used in total for all their courses was overwhelming at times; however, they were very satisfied with the CS2 course.

2) Lectures and resources: The researcher collected viewing data from all the posted videos manually after the exam. Since there were several platforms in use, this was the only way to collect a full overview of engagement with the videos outside of the students' self-reported data. The most viewed lecture video was the first one made (718 views), while the average was 350. It is evident that the first video of each topic gained the most views, decreasing views until the next topic. The most viewed exercise lecture had 436 views and was the first of the course summary videos, while the average was 287. There seems to be a similar trend with exercise videos, with decreasing views throughout the series, but not as significant a difference as regular lecture videos.

In the interviews, students reported different experiences. On the one hand, some students seemed very positive to the freedom of asynchronous video lectures. They said they enjoyed being able to regulate their learning pace by choosing when to watch them, adjust speed, and re-watch sections they did not understand at first. On the other hand, some students reported that using video lectures took a lot more time, was harder to follow, and less motivating than in-person lectures. Generally, the latter group of students reported that studying from home was less effective than on-campus studying.

3) Assignments and support: The use of support through the open labs was tracked by manually counting each post in the Support channel in Teams. The results of this exercise can be found in Fig. 1c. Each post was categorized by gender. As seen in the figure, the number of help-seeking posts peaks close to the assignment deadlines, with the most significant peaks coinciding with the deadlines for Assignment 7 (March 20th) and 8 (March 25th). At this point, the students who had finished all eight assignments most likely had reached the threshold of 750 points, which probably explains the decrease of posts for the remainder of the semester.



(a) Organization

(b) Independent study

(c) Planning and priorities

(d) When: Days

(e) When: Time of day

Fig. 2: Study behaviors by assignments.

We were interested in observing any gender differences in help-seeking behaviors, and there seemed to be a higher number of female students using the online open labs than males. Since the gender distribution of the class population is unbalanced, the percentage of females using the online support channel is significantly higher than for males. However, we cannot be sure of this conclusion since we were not able to count unique posts. Additionally, we do not know how this compares to the on-campus open labs.

When it comes to the interview results, the experience with the assignments and support structures showed similar tendencies to the experience with lectures. Some students were very favorable to the change; some even said the new system worked better than the old one, while other students said the exact opposite. The latter found it more complicated to find help and found calling TAs over video intimidating.

4) Exam and assessment: In the period after assignments were done and before the exam, students seemed to use Piazza rather than Teams to ask questions and discuss. During the interviews, students commented on the fact that a home exam was going to be new to them and expressed some nervousness about that. Besides, the fact that the exam would be graded came up repeatedly. Several other courses chose to change the assessment to pass/fail, while this CS2 course kept the graded regime. In the interviews, the students consistently said that this course would be prioritized since it was graded and that they were motivated to study for the exam.

Immediately after the exam, discussions about how the exam started and continued far into the summer. In general, the discussion was on the level of difficulty on the exam. Many students expressed that the exam was too hard; however, the grade distribution was in line with previous years. The average grade was C, failure rate 21%, and the grade distribution as a whole was very similar to previous years.

B. *Change in Study Behavior*

Thus far, we have looked at engagement and interaction with the organized learning and teaching activities; however, we were also interested in exploring students' independent studying and priorities. Following the framework presented in the introduction, we will examine the students' study behavior across four dimensions: Independent study, Organization, Planning and Priorities, and Time Management (TM). Data were extracted from the students' learning reports, giving us one datapoint per student for each assignment (N=2084).

Fig. 2 depicts the results from the students' learning reports across these dimensions by assignment. For all ten assignments, the mean of each behavioral construct for the student population as a whole was calculated and plotted. Based on this plot, we see some interesting tendencies. There seems to be little change between the campus and the online environment for the organization and independent study dimensions. Planning and priorities seem to be the same throughout the semester; however, time management sees a steady decline throughout the semester.

The tentative findings from these graphs were further explored statistically by looking at each dimension's individual behavioral constructs. However, this proved to be a challenge as the research design, and data collection were not planned for this purpose. Therefore, it did not entirely fit any of the traditional methods of analysis. After some time was spent exploring variable transformations and various non-parametric tests, the authors landed on dividing the dataset into two random groups in order to create independent subsets [13]. The students were randomly placed into one of two groups, with their accompanying learning report data. Group one was analysed using data from assignment 1-5 only (campus environment, n=742), and group two used data from assignment 6-10 (online environment, n=300), thus creating two independent groups. The *n* here refers to the number of valid learning reports used in the analysis. Then, a Wilcoxon-Mann-Whitney test was performed in Stata [16] to examine the difference in study behaviors on campus and online, that is, before and after the pandemic hit. The dependent variables were the different study behavior constructs illustrated in Fig. 2 and were investigated individually against the independent variable. The independent variable was dichotomous, indicating if the assignment was campus-based (0) or online (1). As seen in Table II, these tests provide a slightly different picture of the situation for students. For organization, the tests indicate that there was a difference in study behavior on campus and online, similar to the plots. However, for independent study, the tests found a significant change in all behaviors except for the use of the book, internet and videos, which is not evident in the plots. When it comes to planning and priorities, both the tests and plot indicate no significant differences, while the time management dimension, on the other hand, seems to differ in both.

TABLE II: Summary of differences between the campus and online environment on Wilcoxon-Mann-Whitney rank sum test

	Campus	**Online**	
Organization	Rank sum	Rank sum	z-value
Lectures	427266.5	116136.5	9.39***
Exercise lectures	417672	125731	7.47***
TAs	410832	132571	5.78***
Collaboration	412549.5	130853.5	6.12***
Alone	373270	170133	-3.18**
Independent Study			
Doing assignment	397035	146368	2.55*
Book	386949.5	156453.5	-0.001
Note taking	396441	146962	2.65***
Self made examples	391671	151732	1.22
Lecture examples	400379	143024	3.32***
Internet	394448.5	148954.5	1.78
Videos	384844	158559	-0.52
Memorizing	391904.5	151498.5	1.59
Diagrams	421227.5	122175.5	8.60***
Planning and Priorities			
Math	386888	156515	-0.02
Other computing courses	382223	161180	-1.10
Other	386167	157236	-0.18
TM: Days			
Monday	498253	45150	25.57***
Tuesday	498253	45150	25.57***
Wednesday	498253	45150	25.57***
Thursday	482803	60600	22.02***
Friday	490013	53390	23.67***
Saturday	498253	45150	25.57***
Sunday	498253	45150	25.57***
TM: Time of day			
Morning	396912.5	146490.5	2.68**
Afternoon	419575.5	123827.5	7.55***
Evening	399054.5	144348.5	3.06**
All day	388558.5	154844.5	0.441
N	742	300	

$*p < 0.05, **p < 0.01, ***p < 0.001$

In the interviews, there were some consistent tendencies when it comes to how their study behavior changed. Firstly, the students who described their routines in the campus-based environment as very structured, all had set up similar structures at home, however, the students who were less structured before reported struggling in the online environment. The latter group referenced challenges getting up in the morning, watching all the lectures, and getting started on assignments. They said they missed the lectures and interactions on campus and commented on how that used to help them progress in their learning. Secondly, many students reported that their study hours were changed. Some students said they kept regular working hours, while others reported studying later in the day, and on weekends (something they did not do before). Lastly, many students commented on the social aspects of not being on campus, and several mentioned that they were lonely and felt very isolated. Although many students said they had started informal study groups with friends meeting online, the students consistently commented on the fact that not meeting their peers was challenging. In general, the students who reported negative experiences seemed to be the students who lacked structure in their study behavior, and who might have struggled in the campus environment as well.

V. Discussion and Related Work

This study set out to investigate the differences in computing students' study behavior in the campus-based and online environment, and their interaction with the changes in educational design caused by the COVID-19 pandemic. Firstly, there is a plethora of research on campus, online and blended learning and study environments, both in general education research and the computing education field. With the growth in usage of MOOCs/SPOCs, blended and flipped instructional designs, gamification and online assessment systems, there are many avenues to explore in the literature. To clarify terminology, one could argue that the course investigated in this paper was never fully on-campus, as the students could 'get away' with only meeting their designated TA on campus once a week. All the assignments and submissions were accessible via online platforms, and the lectures and labs were not mandatory to attend. Nevertheless, the authors would argue that the course was not a blended course because the educational design was not intended for the online environment. The lectures were not recorded, and all support was offered only on campus. If the students chose not to utilize these on-campus resources, there was no online alternative. In other words, it was expected of the students to spend time on campus and participate in the educational activities.

In the following discussion, we will explore the results of the current study in light of the research questions and related research. Additionally, it has been pointed out by researchers in the field that we must be careful not to directly compare "emergency remote teaching" to online learning [17]. In the following discussion we aim to explore the online *environment* created by emergency remote teaching and how the students' behaviors developed in this new context.

A. Student Interaction

When it comes to the use of tools and communication channels in the online environment, it is interesting to compare the students' engagement in Piazza and Teams. Piazza received a higher engagement overall, which is not surprising, considering it was used throughout the semester. However, it is striking how Piazza seemed to be the preferred platform for communication when there were no organized activities in place. When the TAs and educators were active on Teams, the students engaged in the tool; however, they preferred Piazza when there were no scheduled activities. One reason for this might be that Piazza was the more familiar platform considering it had been in use in the campus-based environment as well. On the other hand, previous research on computing students' self-regulation strategies proposes that targeted scaffolding will help students adapt their learning [18], [19]. In this case, we can view the scheduled activities within the online environments as a way of scaffolding students' study behavior, which explains the interaction patterns. Additionally, a contributing factor might be that Piazza allows anonymous interactions. The researcher noticed that nearly all students used the option to post anonymously on Piazza, which is not an option in Teams. Lastly, the findings from the interviews regarding the number of tools might also explain this trend; perhaps the students simply preferred to use just one platform.

The current findings on online lecture views are aligned with previous research in the field [20], [21]. Students will watch the early videos but gradually watch less. Previous studies on student viewing patterns have found that the viewing of complete videos decreases as the complexity of the content increases [20], [22] as well as high correlations to assignment timelines [21]. When it comes to the student experience, the interview findings were similar to the general feedback on face-to-face lectures. There is a large discrepancy in how individual students perceive the effectiveness of lectures. Therefore, it is important to consider that with the social component of meeting friends in lectures gone, many students might opt out of watching lectures online [23], [24].

When considering the assignments and support-seeking interaction, the results indicate that the online system worked well. The fact that the number of support requests in the open lab was low relative to the total number of students in the course is somewhat discouraging; however, the students seemed very content with the system. The interview findings suggest that the students also used their designated TA in private channels, and were satisfied with the support they got. On the other hand, previous research on help-seeking behavior and meta-cognition in online and blended environments has found that the students struggle to identify their need for support in time [19]. Additionally, the gender distribution of these posts is interesting, suggesting that female students ask for help more often than males, and it would be interesting to explore this further in relation to similar studies [25].

In general, these results suggest that there is a larger difference between students' study behaviors in the online environment than the campus-based. The interviews indicated similar trends in large individual differences when it comes to lectures, resources and support in the online environment. Based on the collective results, it seems like the difference between the students who mastered the online study environment and those who did not was larger than in the campus environment. In other words, students who were successful and experienced mastery with their study behavior on campus were able to transfer to the online environment without issues. In contrast, the students who struggled on campus struggled even more online. To the best of our knowledge, this has not been identified in any previous research. Furthermore, it is difficult to say whether this is a computing specific finding or general for all students. As second-semester computing students, these students should be accustomed to independently developing their programming skills by transferring the knowledge from lectures and assignments to skills and competencies in CS2.

B. Change in Study Behavior

The change of study behaviors in the campus and online study environment was explored further through plotting means over time and statistically testing the differences. Looking at the graphs as a whole, there are some interesting findings to point out. Firstly, some assignments differ from the rest. Assignment 4 seemed to provoke an increase in most behaviors and in time spent. This discrepancy can be explained by the nature of assignment 4, which was a project-based assignment where the students themselves defined the project over two weeks (the teachers defined the remaining assignments). Furthermore, assignment 8 and 9 see similar tendencies, although not as large. This might be due to the fact that most students would be finishing their required 750 points with a full score on assignment 8/9. Lastly, assignment 10 has largely the opposite results, except for independent study tactics, which was most likely due to the students changing strategies because they are preparing for the exam, and not actually the finishing of assignment 10. Nevertheless, there seems to be a connection between student behavior and assignments also in the online environment [3], [7].

The way the students organized their time seemed to change somewhat in the two different study environments: students spent the same time alone; however, the time spent in lectures, with TAs and collaborating with other students, decreased. When it comes to independent study, students, to a large extent, utilized similar tactics on campus and online, with the exception of videos that seemed to increase slightly. Comparing the effort in CS2 to other courses, there seemed to be no change in the campus-based and online environment.

From the plotting of when students studied, it is evident that the total time spent studying likely decreased since the use of all days and times of day seemed to go down. In general, this is true for the whole semester, and the pandemic might not have had an impact here.

Under organization, it can be observed that with the exception of time spent alone, there seemed to be a statistically significant difference between all the behaviors in the campus-based environment and online. Similarly, for independent study, only reading the textbook, using the internet and videos stayed the same after the online transformation. When it comes to planning and priorities, no statistical difference was found, which is in line with the plots in Fig. 2. Lastly, the analysis of when students studied indicated that both the days and time of day students studied changed.

All of the significant tests indicated that students spent less time or participated less in the online environment activities, something that is clear from the plotted means as well. These somewhat conflicting results can be interpreted in three ways; the students spent less time studying and participated less 1) because of the pandemic, 2) because it was closer to the end of the semester, or 3) a combination of the two. In previous studies comparing campus, online and blended environments, it has been found that time management and effort regulation positively influence grades [26]. Furthermore, study strategies focusing on effectively scheduling, planning, and self-managing study time, while correctly allocating resources and effort despite potential distractions, is more challenging for online learners and more important in a highly autonomous study environment. Seen in connection to the finding on larger differences between students who master the online environment and those who do not, these are the behaviors that seem to be the cause of this difference.

One last finding that is important to discuss comes mainly from the student interviews, and it is difficult to quantify in any statistical way, is the importance of the informal study environment provided by the campus. The social interactions between students, educators and TAs in lectures and labs, happening in breaks, queues, and between various organized activities seemed to be missing in the online environment. Online, students need to know each other's full names in order to contact each other, and it requires scheduling to be working on the same courses at the same time [27], [28]. Connecting with peers has been found to be a sizable challenge for students in an online environment, especially for informal learning interactions [26]. Although informal academic socializing did seem to happen in ad hoc groups, these are invisible to the whole student group, and we are certain many students were left out. The campus provides an open environment, where names and schedules are irrelevant when students naturally meet. Going into a third semester of uncertainty about the availability of a campus, creating an informal academic environment is the hardest challenge we aim to solve.

C. Limitations

In retrospect, there are many things we would do differently, although, considering the sometimes chaotic circumstances, we believe this research is of value. This study has a somewhat unorthodox research design, where the data collection was guided by the access to data, rather than the research questions, which provides some limitations to the research. Mainly, the lack of longitudinal data for all data sources and the fact that we did not have the opportunity to test learning or performance in any meaningful way.

In addition, the transformation for students from a campus-based to an online environment was not the only change for the students during this time period. The country was in full lock-down for several weeks, and the students lost not only access to the campus but also all other infrastructure such as gyms, cafes and public spaces. Many students also moved from their student housing to their parents' house, where their whole family was also most likely working from home.

VI. Implications and Key Takeaways

This research aimed to both expand our understanding of the role of the campus-based study environment for computing students, and provide some lessons learned for other educators in the future. It is clear that the campus plays an important role in many students' study day. Campus-based activities provide scaffolding for students' study behaviors, specifically time management and organization, as well as providing an informal space for social and academic interactions. When moving to an online study environment, students seem to struggle with adapting their study behaviors. They spend less time on organized activities and do not change their independent study habits. Lastly, there seem to be larger differences between those who master studying and those who do not in the online environment, creating a greater ability gap among the student group. Although we did not investigate the effect of this gap on performance, there is reason to believe that this will lead to a significant knowledge and skill gap.

In addition to these important lessons learned for the online environment, this research has also given us some valuable insight on the importance of the campus. Specifically, aspects of campus-based education created indirectly by the educational design. It is clear that valuable learning happens between lectures and labs, in various nooks and crannies of the campus. Learning to learn is an essential competency for future computing engineers and professionals, and one of the important findings of this study is that many students struggle with this skill. It is important now to look back at the traditional educational design and reflect on what we can improve. When hopefully returning to the campus based environment soon, we should use this opportunity to reflect on that practices we take with us from this experience with emergency remote learning. Based on the results in the current study the following questions can be used to kick off this discussion:

- When returning to the campus, how can we maximize the potential of the informal learning spaces? It seems like the campus is essential to the students. At the same time, we know many students spend a lot of time alone at home during a traditional semester. What can we do to engage these students on campus?

- Is the fact that lecture attendance is low in many courses an even greater problem than we thought? Should we be more worried about the students who do not actively use the campus based environment?
- Is the scaffolding provided by the set time-tables and educational structure doing the students a disservice? What can we do to improve students' ability to study and learn independently?
- How do we use online tools in a way that creates interaction, accessibility and engagement? Do we need to consider teaching the skills to use these tools effectively? Furthermore, how can we support students in creating effective help-seeking behaviors?

Researchers and educators spend significant time and resources on designing, implementing and evaluating different learning activities and innovative approaches, however the current study suggest that there are important things happening outside our designs. Viewing these results though the lens of learning theories, the prevalence of constructivism in computing education can further guide this work [29]. Assuming that learning is achieved through students constructing knowledge, these results indicate that many of these constructive interactions happen outside the educational design constructs. Designing computing courses and programs that facilitates the creation of informal learning spaces and supports the development of effective study behaviors will be essential for educators in the future, regardless of the study environment. Students will need the knowledge and skills to be able to construct knowledge independently, both on campus and online, in order to be prepared for the unpredictable future.

References

[1] T. Tressel, S. P. Lajoie, and M. C. Duffy, "A guide for study terminology: Reviewing a fragmented domain." *Canadian Psychology/Psychologie canadienne*, vol. 60, no. 2, p. 115, 2019.

[2] M. Lorås and T. Aalberg, "First Year Computing Study Behavior: Effects of Educational Design," in *2020 IEEE Frontiers in Education Conference (FIE)*. Uppsala, Sweden: IEEE, 2020.

[3] S. N. Liao, S. Valstar, K. Thai, C. Alvarado, D. Zingaro, W. G. Griswold, and L. Porter, "Behaviors of higher and lower performing students in CS1," in *Proceedings of the 2019 ACM ITiCSE*. New York, NY, USA: ACM, 2019, pp. 196–202.

[4] J. Sheard, A. Carbone, D. Chinn, and M. Laakso, "Study Habits of CS 1 Students: What Do They Say They Do?" in *2013 Learning and Teaching in Computing and Engineering*, Mar. 2013, pp. 122–129.

[5] D. Chinn, J. Sheard, A. Carbone, and M.-J. Laakso, "Study Habits of CS1 Students: What Do They Do Outside the Classroom?" in *Proceedings of the Twelfth ACE- Volume 103*. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2010, pp. 53–62.

[6] M. Credé and N. R. Kuncel, "Study Habits, Skills, and Attitudes: The Third Pillar Supporting Collegiate Academic Performance," *Perspectives on Psychological Science*, vol. 3, no. 6, pp. 425–453, Nov. 2008.

[7] V. Hellem and M. Lorås, "The Effect of Mandatory Assignments on Students Learning Outcome and Performance in Introductory Programming Courses," in *2020 IEEE Global Engineering Education Conference (EDUCON)*, Apr. 2020, pp. 704–712.

[8] J. Martin, S. H. Edwards, and C. A. Shaffer, "The Effects of Procrastination Interventions on Programming Project Success," in *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, ser. ICER '15. New York, NY, USA: ACM, 2015, pp. 3–11.

[9] S. H. Edwards, J. Martin, and C. A. Shaffer, "Examining classroom interventions to reduce procrastination," in *Proceedings of the 2015 ACM ITiCSE*. New York, NY, USA: ACM, 2015, pp. 254–259.

[10] A. Luxton-Reilly, Simon, I. Albluwi, B. A. Becker, M. Giannakos, A. N. Kumar, L. Ott, J. Paterson, M. J. Scott, J. Sheard, and C. Szabo, "Introductory Programming: A Systematic Literature Review," in *Proceedings Companion of the 23rd Annual ACM ITiCSE*, ser. ITiCSE 2018 Companion. New York, NY, USA: ACM, 2018, pp. 55–106.

[11] A. Berglund and M. Wiggberg, "Students learn CS in different ways: insights from an empirical study," in *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education*, ser. ITICSE '06. New York, NY, USA: Association for Computing Machinery, Jun. 2006, pp. 265–269.

[12] Trochim, Donnelly, and A. Kanika, *Research Methods: The Essential Knowledge Base*. Cengage Learning, Jan. 2015.

[13] D. T. Campbell and J. C. Stanley, *Experimental and quasi-experimental designs for research*, 2nd ed. Boston: Houghton Mifflin Comp, 1967.

[14] B. Johnson and L. Christensen, *Educational Research: Quantitative, Qualitative, and Mixed Approaches*. SAGE, Nov. 2010.

[15] M. Lorås, R. Hjelsvold, A. Bahmani, S. S. Nykvist, and A. Krokan, "Online Teaching and Learning - First impressions from students as NTNU transitions to an online only mode of learning," NTNU, Trondheim, Tech. Rep., 2020.

[16] StataCorp, "Signrank — Equality tests on matched data," Tech. Rep., 2020.

[17] C. Hodges, S. Moore, B. Lockee, T. Trust, and A. Bond, "The Difference Between Emergency Remote Teaching and Online Learning," p. 12, 2020.

[18] K. Falkner, R. Vivian, and N. Falkner, "Identifying computer science self-regulated learning strategies," in *ITiCSE '14*, 2014.

[19] M. Hauswirth and A. Adamoli, "Metacognitive calibration when learning to program," in *Proceedings of the 17th Koli Calling*. New York, NY, USA: ACM, 2017, pp. 50–59.

[20] M. N. Giannakos, J. Krogstie, and T. Aalberg, "Video-based learning ecosystem to support active learning: application to an introductory computer science course," *Smart Learning Environments*, vol. 3, no. 1, p. 11, Jul. 2016.

[21] M. N. Giannakos, L. Jaccheri, and J. Krogstie, "Exploring the relationship between video lecture usage patterns and students' attitudes," *British Journal of Educational Technology*, vol. 47, no. 6, pp. 1259–1275, 2016.

[22] O. Ozan and Y. Ozarslan, "Video lecture watching behaviors of learners in online courses," *Educational Media International*, vol. 53, no. 1, pp. 27–41, Jan. 2016.

[23] J. L. Hill and A. Nelson, "New technology, new pedagogy? Employing video podcasts in learning and teaching about exotic ecosystems," *Environmental Education Research*, vol. 17, no. 3, pp. 393–408, Jun. 2011.

[24] B. von Konsky, J. Ivins, and S. Gribble, "Lecture attendance and web based lecture technologies: A comparison of student perceptions and usage patterns," *Australasian Journal of Educational Technology*, vol. 25, Sep. 2009.

[25] L. L. Werner, B. Hanks, and C. McDowell, "Pair-programming helps female computer science students," *Journal on Educational Resources in Computing*, vol. 4, no. 1, pp. 4–es, Mar. 2004.

[26] J. Broadbent, "Comparing online and blended learner's self-regulated learning strategies and academic performance," *The Internet and Higher Education*, vol. 33, pp. 24–32, Apr. 2017.

[27] W.-F. Chen, "An Investigation of Varied Types of Blended Learning Environments on Student Achievement: An Experimental Study," *International Journal of Instructional Media*, vol. 39, no. 3, p. 205, Jun. 2012.

[28] M. J. Kruger-Ross and R. Waters, "Predicting online learning success: Applying the situational theory of publics to the virtual classroom," *Comput. Educ.*, 2013.

[29] M. Ben-Ari, "Constructivism in Computer Science Education," *Journal of Computers in Mathematics and Science Teaching*, vol. 20, no. 1, p. 29, 2001.

Paper 6

The Effect of Mandatory Assignments on Students Learning Outcome and Performance in Introductory Programming Courses

Vegard Hellem, Madeleine Lorås

EDUCON 2020

Authors' contributions: Lorås supervised the research design, data collection, analysis, and led the paper writing. Hellem led the process and collected the data as a part of his master thesis project. Lorås performed additional analysis after the thesis and prepared the data for publication.

194

The Effect of Mandatory Assignments on Students Learning Outcome and Performance in Introductory Programming Courses

1st Vegard Hellem
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
vhellem@gmail.com

2nd Madeleine Lorås
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
madeleine.loras@ntnu.no

***Abstract*—In a world in high demand of engineering professionals, higher education should be effective and quality conscious. A better understanding of what type of activities that are best suited for improving students' learning could enable further improvements. In this paper, the effect of mandatory assignments on students' learning outcome in introductory programming courses is explored through a quasi-experimental research study. One group of students were exempted from the mandatory weekly assignments and followed up via biweekly sessions. A control group was recruited to follow an assignment regime in parallel. Through pre- and posttests the learning outcome of the two assignment structures was statistically evaluated. The results indicated that the group of students exempt from mandatory assignments achieved the same learning outcome as the control group. Similarly, no difference was found between the two groups on exam performance. Students have individual learning behaviors and learn to program in different ways, and the instructional design should facilitate individual learning trajectories.**

***Index Terms*—assessment, performance, mandatory assignments, computing education**

I. Introduction

Increasing student numbers provides challenges for teachers and educators at higher education institutions across the world as the search for scalable and effective teaching designs continues. In the Norwegian engineering education, mandatory assignments are a common way to ensure student engagement in a course between lectures [1]. Most often, these assignments are done on a weekly or biweekly basis and are not counted towards the final grade. Instead, such assignments are assessed on a pass/fail basis, where students are required to pass a fixed amount of assignments in order to qualify for the exam. Having assignments besides the exam is for the assignments to address other learning outcomes of the course than those of the exam. The exam could be too short to test all that should be learned in the course, and such the assignments are needed as a supplement in certifying that the students have learned all that they are supposed to. For example, this could be practical knowledge like a chemistry lab, which is unfeasible to test during an exam. Math assignments in a math course are more straight forward learning to prepare students for the exam. Programming courses fall in between these two examples, with assignments often mainly focusing on preparing students, but may test larger collaboration projects and coding challenges for which the exam does not have enough time. In this paper, we will explore a different approach to this traditional instructional technique: removing mandatory assignments.

During the spring semester 2019, a research study was done exploring the effects of mandatory assignments in an introductory programming course. Extensive resources go into grading these mandatory assignments, resources that could be spent on more effective evaluations such as formative feedback [2]. Therefore, the focus of the study was to measure and compare how learning outcome and student performance was affected by having or not having mandatory assignments in an introductory programming course. The research questions were as follows:

- What is the effect of mandatory assignments on students learning outcome?
- What is the effect of mandatory assignments on students performance?

The difference here between learning outcome and performance is based on the measurements. Learning outcome is measured with pre- and posttest, whereas performance is measured with exam grades. Of course, exam performance also measures learning outcome, but we have found that differentiating the learning dimension in this way provides a more nuanced insight.

A. Assessment

In order to explore assignments, we need to discuss assessment, and an important distinction is made between formative and summative assessment [3]. This contrast was first described by Scriben in 1967 [4]. He explained summative evaluation as assessment used to judge the value of an educational program, what had the student learned. Formative assessment targeted improvement for the student, and how they could improve learning. Bloom extended this definition of the purpose of formative evaluation to "Provide feedback and correctives at each stage in the teaching-learning process"

[5]. Multiple studies have found that a formative approach outperforms summative assessment [2], [6], [7]. An important finding is that when the number of formative evaluations increases, students will learn more [2], [7], also for the most low-performing students [8].

B. Introductory programming

In an evaluation of different teaching approaches to introductory programming from 2015, Koulouri et al. studied three distinctive factors of how to improve introductory programming [3]. The choice of programming language and teaching problem solving before programming were found to yield significant improvements in student performance; however, it had variable effects on the acquisition of basic concepts in programming. The last factor was how to use feedback effectively and formatively. Here, they found that formative feedback was not useful unless students actively sought out and responded to feedback. In order to be effective, feedback should be timed and targeted to specific features that one wants students to improve [9]. As computer programs are files that can be run by a computer, there has been significant research into how to automatically grade and correct programming assignments, reducing the strain on teaching resources. These have plenty of issues that need to be looked at, especially for a system that grades the student based on these assignments [10].

Numerous other studies have also investigated what type of activities are most useful to teach computer science. A systematic review by Luxton-Reilly et al. in 2018 found, among other things, that self-paced learning had few examples of usage in universities worldwide [11]. Self-paced learning is a form of mastery learning where students are supposed to demonstrate they have achieved an appropriate level of mastery of a topic before they can move on the next, more advanced, topic in the course. They also found that problem-based learning could increase motivation and social interactivity. However, little evidence that it increases the learning outcome of the students. Problem-based learning was mainly project-based, answering open-ended questions [12], [13]. Through the review, they found evidence that students preferred structured assignments [14].

II. Mandatory assignments

The reason for having mandatory assignments in a course is often twofold. Compulsory assignments could be there to qualify students for the exam, or it could be to ensure they learn skills and knowledge that can not be assessed by the exam. A Norwegian study from 2018 [15] argues why the number of mandatory assignments in engineering education should be reduced based on findings that the use of mandatory assignments has increased without any quality improvement in students learning outcome.

A. Previous work on homework

As the literature on assignments at university level is limited [16], it is interesting to investigate the studies done on homework, in general, from pre-university education. Multiple studies have found a positive relationship between achievement and homework [17]–[19] in mathematics, while others find a non-relation, or even a negative impact on achievement, among these a study from 2010 on 28 different schools, where neither frequency nor homework time had any relation to performance in class according [20]. Similar inconsistent results have been shown in studies linking homework and science achievement. Some found positive relations [21], however, others did not [22]. A variety of factors may have contributed to these inconsistent findings. For instance, the type of homework, grading, how achievement is measured, and what kind of homework indicators that have been used. Studies have been convened on different data, including total time spent on homework, the frequency of homework, the percentage that was completed, the effort needed to complete the work, or the grade given to the homework if being evaluated by the teacher. In summary, the research reviewed has not indicated that there is a clear correlation between feedback on homework and student motivation or achievement gain [23]. It should be noted that homework completion rate has been shown to have an effect, but not the actual deliverance of the homework.

B. Assignments at university level

An interesting study on university-level calculus investigated the relationship between compulsory, graded assignments and assignments with weekly quizzes [16]. The results revealed that there was no statistically significant grade difference between these two groups. This result builds on early results that monitoring assignment completion, rather than just giving them out as an aid in learning the curriculum, does not affect students' performance [24]. However, if students are not given any exercises to aid in learning the syllabus, some results put them at a disadvantage compared to students getting mandatory assignments [25].

Similar results were found for a college degree economic course in a study looking at feedback and grading of assignments [26]. They tried out a concept called selective grading, where only a few select assignments were graded, and it had no effect on students' learning outcome; they produced at the same quality and delivered the same number of assignments.

Research on whether mandatory assignments are helpful in programming courses are limited. A review from 2016 by Danielsiek et al. [27] about ways to teach computer science found no evidence that results on assignments were any indication of how students would perform on the exam. This was regardless of whether the assignments counted towards the final grade, or whether it was just a stepping stone for being allowed to take the exam.

A Norwegian analysis by Haugan and Lysebo from 2018 [15] argues why the number of mandatory assignments in engineering education should be reduced. They concluded with multiple important findings. Among them that the students with less mandatory work, spent more time on each course than before, one of the most important reasons for having mandatory work in the first place. They also found that the average grade increased after the restructuring of the teaching program. This also included, to their surprise, the result for the students with the worst results on a preliminary test.

III. Methodology

The work presented in this paper is based on a master thesis project from 2019 [28]. A quasi-experimental research design was set up and implemented in order to investigate the effect of an intervention on a research population but without random selection [29], [30]. The intervention, in this case, was not having mandatory assignments, and the aim was to measure the effect on learning outcome and performance. An overview of the experiment is shown in Fig. 1, and in the following section, the course, participants, experiment, measurements and analysis will be described further.

A. Course description

The experiment was set up in an undergraduate object-oriented programming (OOP) course at a large university in Norway. The course yields 7.5 ECTS and goes over 14 weeks with a final four-hour exam, in the end, accounting for the whole grade. The programming language used is Java, and the course covers topics such as classes and objects, encapsulation, object structures, exception handling and inheritance. Students generally take this course in their second semester and are required to have completed an introduction to information technology course, which includes programming in Python. The course is mandatory for all the various computer science engineering programs and serves as an elective course for all engineering programs.

During the semester, there are four hours of topic lectures and two hours of exercise lectures a week, as well as mandatory assignments evenly spaced throughout the semester. There are ten assignments in total. They do not count towards the final grade; however, each assignment is graded on a point basis between 50-100, and to qualify for the exam, the student has to reach 750 points. The assignments are based on the curriculum for the current week and the week before. Automatic tests are integrated with the assignments so that both students and teaching assistants (TAs) can easily check the code. To pass the tests, students have to code correctly for all edge cases, as well as name their methods according to the task description. The assignments are delivered online but have to be demonstrated in-person to a TA within a week after the deadline. TAs are generally older students who have completed the course, hired by the department to give feedback, and help students with their assignments, as well as assign points to each assignment. Each TA is responsible for 20 students, and are available in open labs at least six hours every week.

This course design has been in place for seven years, with revisions to the assignments being made regularly. The student feedback is generally positive; however, the workload has been criticized somewhat. On the other hand, students report that the amount of practice and experience with programming in the course is very useful.

B. Participants

Among over 700 students taking the course, 40 students volunteered to be part of the experiment, either as a part of the experimental group with no mandatory assignments or as a part of a control group. The experimental group were exempt from doing the mandatory assignments and were instead given the freedom to choose what learning resources to use. These resources could include the proposed assignments for the course, but the students were not required to deliver them. They were, however, required to attend biweekly meetings with a TA where they had to describe what they had learned in the previous weeks and explain how they reached the learning objectives for that week. These meetings along with the pre and posttest, served as the experimental group's qualifying activities for the exam.

The students participating in the experiment were from various study programs within computer science. 47.5% were from computer science engineering, 20% from computer science and business, 15% from computer science, 15% from communications, and 12.5% from engineering and ICT. The gender distribution of the participants was 50/50 male and female.

C. Experimental design and ethical concerns

Both the experimental group (N=22) and the control group (N=18) were required to hand in weekly reports, as well as take a pre and post programming test. As naturally, they have learned much more during the semester; the second test was more difficult and involved more object-oriented programming principles than the first test. Both these tests were corrected by one of the authors, using anonymized IDs that did not indicate to which group the writer of the answers belonged. In addition to the weekly reports and pre-/posttests, some of the participants also volunteered to attend an informal interview about their experience at the end of the experiment. Lastly, the participants consented to their exam answers and results being collected for analysis.

The reason the selection of students was not random, was because the teaching team had concerns about implementing such a change to the students without certainty that their learning would not be affected negatively. Therefore, we decided that students would have to volunteer to be part of the experiment, which subsequently limited the number of participants as well. The experiment was approved by the Norwegian Centre for Research Data.

D. Measurements

In order to analyze the difference in learning outcome and performance for students with and without mandatory assignments, we created two hypotheses'.

- H_{11}: There is a difference in learning outcome for students with mandatory assignments and students without.
- H_{12}: There is a difference in performance for students with mandatory assignments and students without.

The learning outcome was measured through either the change of learning or with a modified pretest. The change

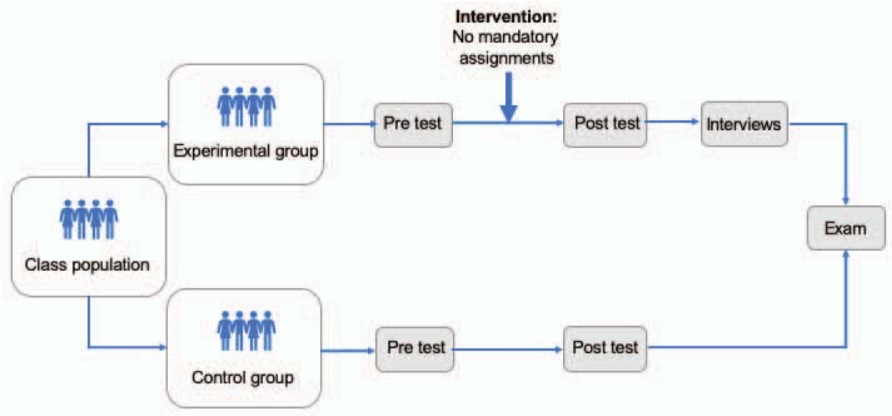


Fig. 1. Overview of experimental setup

of learning was measured as learning gain by subtracting the posttest score from the pretest score for each student. In order to deal with the quasi-experimental designs with non-randomized participants and the low number of participants, an adjusted pretest score was created in order to compensate for the nonequivalent groups design in a Reliability Corrected Analysis of Covariance model [31]. The reliability was calculated using Cronbach's Alpha, giving a reliability score of 0.817. This reliability was used to calculate adjusted pretest scores for feeding into the statistical model. The performance was measured with final exam grade, ranging from 0-5, where 0 is an F and 5 is an A.

Consequently, *post test score* and *exam grade* were the dependent variables, as indicated in bold in Table I. The independent variable *group* differentiated between the experimental and the control group. Additionally, *adjusted pretest score* acted as a covariate when analysing learning outcome and *grade in introductory programming (CS1)* for performance. All variables used to create these measures, as well as the variables used in the analysis, are summarized by group in Table I.

TABLE I
SUMMARY OF VARIABLES BY GROUP

Variable	Experimental		Control	
	μ	σ	μ	σ
Posttest score	49.54	20.69	53.36	18.28
Exam grade	2.43	1.65	2.17	1.82
Pretest score	54.30	17.04	61.17	16.73
Adjusted pretest	54.17	13.92	61.17	13.67
Gain score	-4.76	16.21	-7.81	11.62
Grade in CS1	3.74	0.96	3.83	1.04
N	22		18	

E. Analysis

In order to test the difference in learning outcome and performance, t-tests and ANCOVA models were run using posttest scores and exam grades as dependent variables, respectively.

Firstly, a t-test was used to compare the mean of the change between the posttest and the pretest to look for a statistically significant difference in learning gain. Secondly, an Analysis of Covariance (ANCOVA) was used to estimate the difference between groups on the posttest and exam, after having adjusted for initial differences in the pretest. Similarly, t-tests were performed to compare the exam performance of the two groups, using the grade from the previous introductory programming course (CS1) as a covariate.

IV. Results

One of the assumptions of an ANCOVA test is that the covariate (adjusted pretest score and grade in CS1) does not vary among the groups. The interaction between group and adjusted pretest score was not significant, $F(3,37)=12.15$, $p=0.799$, similarly for group and grade in CS1 ($F(3,37)=1.45$, $p=0.0549$. Furthermore, conducting the statistical tests, the necessary conditions for normality and homoscedasticity were confirmed using the Shapiro-Wilk test and Levene's test. Since the assumptions were not violated, linear models were created for learning outcome and performance.

A. Learning outcome

The t-test for learning outcome found that there was no improvement or reduction in learning outcome for students that did not have mandatory assignments. Running the t-test on the results of the gain score, yielded no significant difference for these groups, $t(39) = 0.672$, $p = 0.505$. The means of both the control group and the experimental group were well inside the 95% confidence interval of these two variables, mainly due to a high standard deviation of the dataset. The t-test tries to explain whether there is a substantial statistically probability that the dataset differs because of the independent variable, the different treatment in assignments that the groups had. Running this test gave the result of it not being statistically probable that the group variable could explain the difference.

In order to verify the result of the t-test, reliability corrected analysis of covariance model was run. This yielded, like the t-test, no statistically significant differences for explaining the posttest scores based on the group (p=0.773, adjusted R^2 = 0.467). Although the model as a whole is statistically significant, the R^2-value comes mainly from the adjusted pretest score, which explains 47% of the differences in the posttest score. The results from the statistical analysis of learning outcome can be seen in Table II.

TABLE II
LEARNING OUTCOME MODEL

Post test score	β	σ	t
Group	-3.17	4.62	-0.69
Adjusted pretest score	0.998	0.165	6.05***
Adjusted R^2	0.469		
F(2,38)	18.65***		

* p<0.05, ** p<0.01, *** p<0.001

Consequently, we argue that there is not enough statistical evidence to accept the alternative hypothesis', so we accept the null hypothesis' that there was no improvement or reduction in learning outcome based on assignment regime.

B. Performance

The results of the t-test yielded no significant change in performance between students who did mandatory assignments and students who did not, t(39) = 0.494, p = 0.624. Like the t-test, the linear model indicated no significant differences in explaining the exam performance in OOP based on the exam performance in CS1 and the group (Table III). Consequently, we argue that there is not enough statistical evidence to accept the alternative hypothesis', so we accept the null hypothesis' that there was no improvement or reduction in performance based on assignment regime.

TABLE III
PERFORMANCE MODEL

Exam grade	β	σ	t
Group	-0.677	0.496	-1.36
Grade CS1	0.403	0.246	1.63
Adjusted R^2	0.038		
F(2,38)	1.80		

* p<0.05, ** p<0.01, *** p<0.001

C. Other models

To see if there were any other factors that could have played out on the results, several statistical tests were conducted with new models. Among others, an analysis of whether the mean of the groups' posttest scores, when looking at the grade in CS1. The result here is interesting, although a statistical t-test showed no significant statistical results, as seen in Table IV.

New models were also run with the modified ANCOVA model to see if any other variables better could explain the difference in the posttest score. The grade in CS1 was encoded into two groups of high-performing (A and B) and lower-performing students (C and D), to see whether this variable

TABLE IV
LEARNING OUTCOME MODEL BY GROUP AND GRADE IN CS1

Group	N	μ	σ	t
Students with an A in CS1				
Experimental	5	42.5	12.7	
Control	6	58.9	7.63	-1.16
Students with a B in CS1				
Experimental	10	45.2	6.05	
Control	5	55.8	10.9	-0.926
Students with a C in CS1				
Experimental	5	62.0	6.70	
Control	5	51.1	6.49	1.10
Students with a D in CS1				
Experimental	3	55.0	4.27	
Control	2	38.6	0.750	2.931

* p<0.05, ** p<0.01, *** p<0.001

better explained the differences. This yielded approximately the same results as before, with the adjusted pretest score still being mainly responsible for explaining the difference, although now with an adjusted R^2-value of 0.47 (p=0.791).

Running other models on exam performance yielded similar results. Using the adjusted pretest score as a covariate instead of grade in CS1 increases the R^2-value (0.167); however, the model was still not significant (p=0.198).

To summarize, there was no indication that any variables, outside of the pretest score, could explain the differences in the posttest score or exam performance in any significant way. No statistical models gave evidence to reject the null hypothesis.

D. Student feedback

Informal interviews were conducted with some of the students from the experimental group to get some qualitative insights into how the students experienced having no mandatory assignments, how they felt about the exam, and how prepared they felt that they were for further studies.

A majority of these students reported that they followed the assignments that the rest of the class did; however, they enjoyed not having to deliver the assignments. They reported that the lack of mandatory assignments made it more fun to work with the course compared to other classes. As exemplified by this statement:

> *"It has been inspiring to follow a different type of assignment scheme. I have had to work differently, more independently, I have taken responsibility for myself, and I have reacted positively to that. I get to decide for myself how I want to learn and what to learn."*

A number of the students reported that they followed web-based courses to learn the curriculum. Most of these courses were based on small videos explaining a subject and many practical assignments. Many of these felt that they were unsure whether the courses fulfilled the curriculum, and therefore ended up doing more work by looking at the exercises as well. Having to focus on the learning goals, and not assignments, meaning they focused more on what they were supposed to learn, and not just passing tests. For example, one student stated:

> *"I have completed the assignments to learn something, not just because I have to. I think I have learned more by that, and it has been more motivating and fun to work with the course. I've looked more at the learning goals of this course."*

On the other hand, some felt that it was easier to neglect the course when they had other courses with deadlines coming up. For instance:

> *"There have been times where I have not worked with OOP in a week because I have done other things. Then I work more next week. This has caused me to not work as evenly as I could have done if it was mandatory."*

In addition, the students seemed very aware of their personal learning preferences. Many said something alongside, "it works for me, but not necessarily for everyone else." The biweekly meetings with TAs were pointed out by many as a good thing to enforce workflow when having to prepare for these meetings. They all mostly agreed that the motivation was high. However, it had gone up and down during the semester. Especially when other courses were deadline heavy, motivation to work with OOP was lower.

V. Discussion

A. Learning outcome

The results yielded no significant change in learning outcome between students that did mandatory assignments and students that were given autonomy and freedom. Assignments are an excellent way to check whether the student has learned something, and for following alongside similar tasks as will be given on the exam. The current study has indicated that measuring whether a student has done the assignment is not necessarily more helpful than merely helping them with the assignments or projects, letting them learn however they like.

The results found in this experiment is consistent with some of the results from the literature on the fact that monitoring assignment completion does not increase learning outcome [16], [24], [26]. These studies also showed no statistically significant difference for similar experiments, with some students having compulsory assignments, and others given more autonomy.

The formative assessment given by the TAs is well known for providing positive results for students that are open for feedback. However, students that are not open for feedback are being spent many resources on checking whether they have done the assignments. These resources could be better spent on more receptive students, focused on teaching the students what they need to learn when they are open for learning it, instead of a fixed schedule for every student that does not provide any autonomy.

The results from the interviews summarize that the majority of the students were happy to be free from mandatory assignments and that they felt this fitted better to their learning style. It is interesting that many chose to follow the assignments, even when not having to do them. It is noted that those who decided to do so did it because they wanted to be sure they learned everything that was related to the exam and not miss out on anything. This shows a considerable focus on the exam and the grade that is given there, while not the most important for a university to teach. The university wants students to have learned the learning goals of the course, and the exam is a summative way to measure that. Many things in a course are not asked about on the exam, due to time or practical constraints, and students choose not to focus their time on such knowledge. This is also consistent with previous findings, where students still delivered the assignment when only a select few were graded [26].

Additionally, many noted that it was more fun to do the assignments when they did not have to do it, and did not have to complete everything, but rather focus on the learning goals. This is what teachers also want to achieve with assignments, to focus on learning goals, and that the students have learned something, not just performed a task successfully. Their biweekly conversations with TAs also achieved a more formative feedback session, where they focused on whether something was learned, and how the student could improve. This setting should be further explored in further work to see how students could benefit most from a session with an experienced student.

B. Performance

The results looking at the difference between the grades are particularly interesting, even though there were not a statistically significant enough difference. The assumption was that more autonomy and more freedom would be better for the higher-performing students, which manage to learn on their own, and are not in a significant need for guidance. However, the results indicate the exact opposite, with A and B students in the experimental group getting outperformed by A and B students in the control group, and the opposite for C and D students. Contrary to popular belief, that may indicate that students that lower-performing students may not require as much guidance, but rather need autonomy to work at their own pace, instead of being forced through a specific set of assignments. It could also mean that lower-performing students might find other and more unethical ways to complete assignments. As they are unable to do them, when given more autonomy, they complete the assignments without having the pressure of a deadline. This result is also more consistent with the findings from Haugan and Lysebos study, where the lower performing students in the pretest did even better on the exam [15]. It should be noted that the number of observations within each grade is very low, and a higher number of participants is needed to get a more meaningful result. It may also be that stronger students attribute more of their learning to the exam period and learn more in a shorter period of time, and therefore have delayed more of the work until the end of the semester.

When discussing lower and higher performing students, it should also be discussed how to allocate resources per student. In the current assignment system, all students have to meet their TA to demonstrate their code and understanding. They may meet as often as they want to get help in understanding the assignment and complete it. With resources that could focus less on the approval of exercises and more on teaching and guiding students, resources could be further utilized by the students that need them. Some mechanisms might also be in place to get the lower performing students to use the available resources. There will always be students that do not put in the effort needed when not giving strict guidelines for what and how to learn and when to deliver. The discussion should, therefore, be on whether it is more important to provide more autonomy to the students who want it than to force everyone through the same mandatory arrangement.

C. Implications

The results of this experiment have indicated there is no statistical difference in learning outcome or performance for students with mandatory assignments, and for students without mandatory assignments. Feedback from the students has also indicated that students being released from deliverance of compulsory assignments will do the assignment nevertheless. They do so because they want to learn the subject and prepare for the final exam. It is unsure whether they would have done that if they knew that their classmates not necessarily had done these assignments. It could be that when they knew everyone else had to do these, they were afraid of falling behind. Whether they did exercises or not, the result indicated that the average time spent on the course per week was less for the students not having to do assignments, even though they achieved the same learning outcome.

Going back to why we have mandatory assignments, there were mainly two reasons. One is forcing students to work evenly throughout the semester, and guiding them in what part of the curriculum they should have gained an understanding of at any given time. Secondly, tests are used to test specific parts of the curriculum that are unpractical due to time or resources to test at the exam. It is hard to let go of mandatory assignments, as still, these parts would have to be tested somehow. When it comes to the first reason, this is just one of many possible options to teach students the material and to help them work. While assignments can be beneficial for many students, there is no appropriate documentation that they are helpful for everyone, and lots of resources are spent on testing whether the students have done them. This also adds extra stress for the students, who must go from deadline to deadline to complete an assignment. Freeing students from thinking about what to deliver to a deadline, may make them more subject to thinking what they should learn in any given week. Focus on what to learn instead of what to complete shifts the focus to what is essential for both professors and students alike, and if the admittance of mandatory assignments as a failure can help in that regard, it should be seriously considered.

As multiple studies pointed out [2], [6], [7], the summative feedback of delivering homework or exercises does not give benefits for the students, and the results of this experiment support these statements. Assignments are a helpful tool for preparing students for the exam, guiding them into learning more about the curriculum of the week, and measuring their progress, but the assessment of the exercises does not necessarily benefit the students. It is interesting that a majority of the students in the control group believed otherwise, and that should also be taken into consideration before launching an all-out experiment testing such an arrangement. Lastly, a reasonable question to ask is whether one should consider grading the assignments and including these marks in the final grade. In this case, the Norwegian university law prohibits using TAs for grading that counts towards the final grade, which makes it nearly impossible to implement in a course with 700 students and 10 assignments. On the other side, there are course designs that could incorporate more formative grading throughout the course and these results on mandatory assignments can help inform these design regardless of grading scheme.

This experiment has been conducted on students from different study programs. All study programs have a high focus on computer science but are built up in different ways. Different study programs may learn and be motivated by different things, and this is important to keep in mind when designing a class. Students from different study programs may have a different learning style, while the same can also be said of students from the same study program. Designing a university course for different learning styles means having to give up inflexible systems for adaptable ones.

The most important implication is the need to give engineering students the best tool and guidance for learning and studying, and to educate the engineers for tomorrow. The world needs technologists in the future with the ability to learn and adapt, and educational institutions should take their part when it comes to finding the best possible way of teaching computer science.

Given that assignments, or at least mandatory assignments, do not seem to be any help in students learning, the focus onward should be on how students study and learn, and what is the best way to aid in their learning process. The students approach to learning could be helped along by various exercises or assignments, be them mandatory or not, to guide in this process. The choice of method could be exercises, group projects, pair programming, or other practical tools for teaching computer science. However, if only given compulsory assignments, that will not leave room for self-study and for learning styles that are not aligned towards exercises as a learning activity.

D. Limitations

Due to the quasi-experimental nature of the experiment, the small number of participants, and a variety of other factors, many biases could have affected the results of this experiment. Students may be colored by their experiences with other courses, and their extensive use of mandatory assignments in other classes parallel to the trial in this course. They may thus be tired of deadline sprints and give a more positive review to different types of learning approaches than what they usually would do.

Students who signed up for the control group have chosen to not sign up for the experimental group, and have as such chosen to do the assignments themselves. They would naturally be motivated to do assignments and are typically among the most motivated students. The same goes for the experimental group, especially when it comes to learning outcome, that they might be the type of students that learn best when given autonomy and freedom, and as such, does not represent the entirety of the student population sufficiently. The fact that many of them chose to do assignments anyway leads to thinking that they want assignments to learn anyway, and as such, discredits that bias. As for the entire experiment, conducted with such a low number of students, there are significant reasons why the result could be as it is. The students following alongside know full well that they work best given autonomy, and therefore signed up for the experimental group. The control group, while given the option of freedom, chose to follow alongside a strict schedule. There are, of course, outliers here, with the probability that several of the participants signing up for the experimental group because they did not want to do assignments, and wanted to have more free time and do less work throughout the semester.

The experimental group also have certain threats to validity. They have volunteered and chosen to be part of a small test group. This could lead to them being more positively inclined than what they otherwise would have been and felt more pushed to work harder in the course than they would have done if they knew they were not being measured.

As another threat to validity, much of the reduction in gain score between the pretest and the posttest seemed to be because people were unable to complete the test, thus giving an extra advantage to fast typers, and students solely focusing more on the quality of the first assignments, and then not having enough time for the last part. This could have skewed the results, highlighting more individual traits than the learning outcome that could have come out of distinct groups.

VI. Conclusion

In summary, the experiment found that there were no statistically significant differences between learning outcome and performance for students following a mandatory assignment program and students that were given more autonomy to obtain the necessary course knowledge through their own means. This result indicates that mandatory assignments are not necessarily helpful for learning the course in introductory programming courses. The implication of this is that there should be a consideration of whether resources going into grading assessments are better spent otherwise. Assignments are also given out to test curriculum that can not be tested on the exam, but the emphasis on how much of the course is assignments, and how much is self-study should be reconsidered. Assignments along the semester helps to push students into effective study and learning behavior, and give them goals to work towards that are not too far into the future, like the exam. However, there should be more focus on formative evaluation and self-study throughout the semester.

Reducing the number of mandatory assignments in a course can be one way of bringing together the best of both worlds, avoiding students' procrastination while at the same time giving them time to focus on learning the curriculum through self-study. This study does not aim to get rid of assignments all together, as exercises are beneficial for gaining knowledge, and knowing what you have learned and what you have missed. However, collecting and grading the assignments may not be as helpful as we once thought.

References

[1] P. Lauvås jr. and T. Sandnes, "Mandatory coursework in higher Norwegian IT education," in *Norwegian Conference for Education and Didactics in IT subjects*, Narvik, Norway, 2019. [Online]. Available: https://ojs.bibsys.no/index.php/NIK/article/view/648

[2] J. Hattie and H. Timperley, "The power of feedback," *Review of educational research*, vol. 77, no. 1, pp. 81–112, 2007.

[3] T. Koulouri, S. Lauria, and R. D. Macredie, "Teaching introductory programming: A quantitative evaluation of different approaches," *ACM Transactions on Computing Education (TOCE)*, vol. 14, no. 4, p. 26, 2015.

[4] R. W. Tyler, R. M. Gagné, and M. Scriven, *Perspectives of curriculum evaluation*. Rand McNally, 1967, vol. 1.

[5] B. S. Bloom, "Some theoretical issues relating to educational evaluation," *Educational evaluation: new roles, new means: the 63rd yearbook of the National Society for the Study of Education*, no. part II, pp. 26–50, 1969.

[6] T. J. Crooks, "The impact of classroom evaluation practices on students," *Review of educational research*, vol. 58, no. 4, pp. 438–481, 1988.

[7] P. Black and D. Wiliam, "Assessment and classroom learning," *Assessment in Education: principles, policy & practice*, vol. 5, no. 1, pp. 7–74, 1998.

[8] A. Vihavainen, J. Airaksinen, and C. Watson, "A systematic review of approaches for teaching introductory programming and their influence on success," in *Proceedings of the tenth annual conference on International computing education research - ICER '14*. Glasgow, Scotland, United Kingdom: ACM Press, 2014, pp. 19–26. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2632320.2632349

[9] T. Koulouri, S. Lauria, and R. D. Macredie, "Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches," *Trans. Comput. Educ.*, vol. 14, no. 4, pp. 26:1–26:28, Dec. 2014. [Online]. Available: http://doi.acm.org/10.1145/2662412

[10] B. Cheang, A. Kurnia, A. Lim, and W.-C. Oon, "On automated grading of programming assignments in an academic institution," *Computers & Education*, vol. 41, no. 2, pp. 121–131, 2003.

[11] A. Luxton-Reilly, Simon, I. Albluwi, B. A. Becker, M. Giannakos, A. N. Kumar, L. Ott, J. Paterson, M. J. Scott, J. Sheard, and C. Szabo, "Introductory Programming: A Systematic Literature Review," in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE 2018 Companion. New York, NY, USA: ACM, 2018, pp. 55–106, event-place: Larnaca, Cyprus. [Online]. Available: http://doi.acm.org/10.1145/3293881.3295779

[12] M. Lykke, M. Coto, S. Mora, N. Vandel, and C. Jantzen, "Motivating programming students by problem based learning and LEGO robots," in *2014 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2014, pp. 544–555.

[13] N. Avouris, S. Kaxiras, O. Koufopavlou, K. Sgarbas, and P. Stathopoulou, "Teaching introduction to computing through a project-based collaborative learning approach," in *2010 14th Panhellenic Conference on Informatics*. IEEE, 2010, pp. 237–241.

[14] D. C. Cliburn, S. M. Miller, and E. Bowring, "Student preferences between open-ended and structured game assignments in CS1," in *2010 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2010, pp. F2H–1–F2H–5.

[15] J. Haugan and M. Lysebo, "Hvorfor antall arbeidskrav bør reduseres," *Uniped*, vol. 41, no. 03, pp. 347–360, 2018.

[16] B. Gutarts and F. Bains, "Does Mandatory Homework Have a Positive Effect on Student Achievement for College Students Studying Calculus?" *Mathematics and Computer Education*, vol. 44, no. 3, p. 232, 2010.

[17] S. Areepattamannil and B. Kaur, "Factors predicting science achievement of immigrant and non-immigrant students: A multilevel analysis," *International Journal of Science and Mathematics Education*, vol. 11, no. 5, pp. 1183–1207, 2013.

[18] J. R. Cheema and K. Sheridan, "Time spent on homework, mathematics anxiety and mathematics achievement: Evidence from a US sample," *Issues in Educational Research*, vol. 25, no. 3, p. 246, 2015.

[19] W.-C. Mau and R. Lynn, "Gender differences in homework and test scores in mathematics, reading and science at tenth and twelfth grade," *Psychology, Evolution & Gender*, vol. 2, no. 2, pp. 119–125, 2000.

[20] R. De Jong, K. J. Westerhof, and B. P. Creemers, "Homework and student math achievement in junior high schools," *Educational research and Evaluation*, vol. 6, no. 2, pp. 130–157, 2000.

[21] J. Wilson and J. Rhodes, "Student Perspectives on Homework." *Education*, vol. 131, no. 2, 2010.

[22] J. D. House, "The effect of student involvement on the development of academic self-concept," *The Journal of Social Psychology*, vol. 140, no. 2, pp. 261–263, 2000.

[23] E. F. Christensen, *The effect of homework choices on achievement and intrinsic motivation*, 2002.

[24] G. Weems, "The impact of homework collection on performance in intermediate algebra," *Research and Teaching in Developmental Education*, pp. 21–25, 1998.

[25] C. M. Cartledge and J. E. Sasser, "The Effect of Homework Assignments on the Mathematics Achievement of College Students in Freshman Algebra." 1981.

[26] E. Miller and G. Westmoreland, "Student response to selective grading in college economics courses," *The Journal of Economic Education*, vol. 29, no. 3, pp. 195–201, 1998.

[27] H. Danielsiek and J. Vahrenhold, "Stay on these roads: Potential factors indicating students' performance in a CS2 course," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. ACM, 2016, pp. 12–17.

[28] V. Hellem, "The Effect of Mandatory Assignments on Students' Learning Outcome and Motivation in Introductory Programming Courses," Master's thesis, NTNU, Trondheim, 2019. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2625825

[29] L. Cohen, L. Manion, and K. Morrison, *Research Methods in Education*, 5th ed. London, UK: Routledge, 2002.

[30] D. T. Campbell and J. C. Stanley, *Experimental and quasi-experimental designs for research*, 2nd ed. Boston: Houghton Mifflin Comp, 1967, oCLC: 247359300.

[31] Trochim, Donnelly, and K. Arora, *Research Methods: The Essential Knowledge Base*. Cengage Learning, Jan. 2015.

204

Paper 7

Creating Learning Environments Within the Constraints of Higher Education - a Case Study of a First-Year Computing Program

Madeleine Lorås, Trond Aalberg

EDUCON 2021

Authors' contributions: Lorås led the paper writing and was the main author. Lorås and Aalberg designed and supervised the study. Lorås collected the data, performed the analysis and wrote the paper. Aalberg provided general supervision of the research and the paper writing.

 206

Creating Learning Environments Within the Constraints of Higher Education - a Case Study of a First-Year Computing Program

Madeleine Lorås
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
madeleine.loras@ntnu.no

Trond Aalberg
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
trond.aalberg@ntnu.no

***Abstract*—Designing a good learning environment is key to improve the student experience and ensure learning. However, it is becoming increasingly challenging to create such environments due the growing number of students and the push to optimize the use of learning facilities. The increased administration of higher education creates a limited room for action for educators to innovate and develop effective educational designs. This paper describes a case study of how one group of educators attempted to solve certain challenges within one university's constraints. The problem observed was that the first-year students were exposed to fragmented scheduling and limited access to collaborative spaces, resulting in a reduced sense of belonging and ineffective study behaviors. At the same time, these students were enrolled in large introductory courses from various departments where we did not have the mandate to make any substantial changes. The solution we came up with was a Study Day Initiative where all the first year computing students were invited to participate in a low threshold study day where teaching assistants were prepared to help with any and all assignments. We were able to clear a full day in the students time table and found an appropriate area within the department's lab spaces. The Study Day Initiative has been in place for three years, receiving very good feedback from students who report being satisfied, making friends and having improved study habits. In this paper we will describe the process behind this initiative, how the constraints of a large university were overcome and present results from the surveys of the participating students.**

***Index Terms*—Computer Science Education, Computing Education, Higher Education, Study Behavior, Educational Design, Learning Environments**

I. Introduction

Students within the computing and engineering disciplines often follow an educational design consisting of lectures, labs, and assignments to do individually or in groups. These are the explicit design parameters implemented by educators. In addition, these designs imply a substantial individual effort in processing lecture notes, preparing for labs, working on assignments, and other individual study behaviors. In order to be successful in these activities, students need continuous time, physical space, and enough support, all key elements of an effective learning environment. Educators' ability design and impact the learning environment across and between courses is constrained by the the current trend of increased administration in higher education does. This paper will describe a case study of how one group of educators attempted to solve certain challenges within one university's constraints.

Educational psychologist John B. Biggs described the learning environment process in his seminal work on student learning processes in the 1980s. In his Presage, Process, and Product (3P) model of learning in higher education, he described how "students undertake, or avoid, learning for a variety of reasons; those reasons determine how they go about their learning, and how they go about their learning will determine the quality of the outcome" [1, p.5]. An important part of the presage is the teaching context, which, in addition to the learning environment, includes the curriculum, assessment, and teaching methods. Common for these factors is that the institution controls them, whereas the other aspect of presage, the student characteristics, exist prior to the learning and relate to the student. The final two parts of the model, process, and product are related to the students' approaches to learning and the learning outcome, respectively. In the current study, we focus on one of the presage factors, namely the learning environment. Students' perceptions of the learning environment influence how they learn as well as the context is self [2]. Furthermore, there exists learning environments within each course in addition the class environment [3]; however, in this case we will only be examining the *student-driven learning environments* created outside the organized classrooms and between scheduled lectures.

As educators, we aim to implement the most effective educational designs and pedagogical activities for our students in order to ensure they learn the content and skills needed. Even though educators have the best pedagogical intentions, they must often make decisions based on organizational and structural constraints. Educators must navigate in a jungle of rules, guidelines, deadlines, best practices, and educational innovations. This jungle, or educational context, is different from institution to institution. The current case study illustrates how one group of educators navigated one institution's jungle of constraints in order to solve a pedagogical problem of fragmented student learning. With this work, we aim to explore a framework for discussing educational design parameters so that educators across institutions can communicate more effectively about structural innovations and their effects. Hence, the research question *how can educators develop educational designs to improve students' learning environment within the constraints of a large university?*

A. The case

The problem in the case presented in this paper was that the first-year computing and engineering students were exposed to fragmented scheduling and limited access to collaborative spaces, resulting in a reduced sense of belonging and ineffective study behaviors. At the same time, these students were enrolled in large introductory courses from various departments where we did not have the mandate to make any substantial changes. The concern was that these ineffective behaviors would develop further and become a challenge for the students later on, and limit their general competency as future engineers and professionals. These worries were backed up by data from the annual The National Student Survey, where the learning environment indicators were below the national average for computing students [4]. Through evaluation questionnaires and focus groups, local investigations into this phenomenon found that the fragmented study week was one possible problem.

The solution we came up with was a Study Day Initiative (SDI) where all the first-year computing students were invited to participate in a low threshold study day were teaching assistants (TAs) were prepared to help with any and all assignments. We were able to clear a full day in the students' timetables and found an appropriate area within the department's lab spaces; however, this required some intricate scheduling negotiation and room allocation trickery. Both this process and the student's experience will be systematically examined through a case study approach. In the next section, we will briefly explore related work on the connection between educational design constraints and the student learning experience. Following that, we describe the methodology and results, ending in discussion and implications.

II. Educational Design Innovation and Constraints

Previous research by the authors has explored the relation between educational design and study behavior within computing education, aiming to model the intricate relationship between learning activities, pedagogical design, and the learning outcomes [5]. The results of this initial work is the model presented in Fig. 1, which illustrates the structure of the student-driven learning environment. On the left side, the model depicts the tacit dispositions, and behaviors students input into the various planned and implemented teaching and learning activities (middle). The students' study behaviors interact with the educational conditions, and the outcome is learned skills, knowledge, and competency. In relation to Biggs' 3P framework, this model of computing students' study behavior depicts the interaction between presage and process. For the purpose of the current case study, the educational conditions are of main interest. These are the aspects of their study day and week students are focused on, and it is what drives their study behavior and learning process.

Taking a closer look at these conditions, it is interesting to differentiate where the control lies in the institutional ladder. For example, at the case institution, the scheduling of lectures lies on the institution level, whereas the course teacher sets the content of the lecture. These distinctions are important because when educators aim to develop and implement holistic educational innovations that take into account the whole educational experience, they must know who has the deciding power. Although these structures and control dynamics may be different from institution to institution, the framework presented in Table I describes the general parameters and how they relate to the current case. In this framework, education is viewed at three levels: institution (macro), program (meso), and course (micro) [6]. The institution level describes the central or highest level, which varies in size and control. The program level here refers to wherever the students are enrolled. In some educational contexts, this might be a school of engineering or a major; however, students are organized into study programs in this case. Lastly, the course level is perhaps the most universal construct. Although higher education institutions are organized in many different ways, this framework aims to incorporate most designs and highlight the interconnected complexity [6].

Previous research has found that institutional policies and mechanisms are central to the student experience, and design parameters such as class size and physical learning environment can either support innovation or present significant barriers to it [7]. Furthermore, institutions need to cultivate

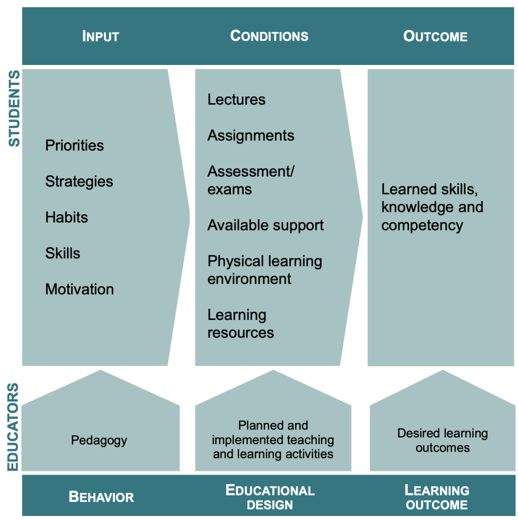


Fig. 1: Model of student behavior and educational design.

TABLE I

Level of control	Description	Parameters
Institution *Rector/pro-rectors,* *central administration*	Admission Learning environment Scheduling and timetables	Prerequisites, enrollment structure Campus layout Lecture and lab time slots
Program *Program leaders, dean*	Program design	Number of semesters Weight of a course (number of credits) Enrollment and admission regime Parallel vs. modular courses
Course *Course teacher,* *department head*	Course structure Learning activities Assessment	Open or closed enrollment Number of students Pedagogical design Number of lectures Number of assignments and/or projects Individual or group-based activities Type of assessment and exams

and stimulate a culture for innovation among educators, which involves supporting practices that conflict with institutional design [8]. There are interconnected complexity and conflicting visions among the course, program, and institution levels, which need to be thoughtfully navigated in support of innovative assessment and pedagogies in higher education [6].

III. Methodology

Case study methodology is a good way to describe, explain, or explore a phenomenon [9]. The case study presented in the current paper has a holistic design with one unit of analysis investigated over three years and be characterized as evaluative [10]. The unit of analysis is the development of the Study Day Initiative (SDI), implemented in order to meet certain student needs within the constraints of one large university in Norway. Furthermore, the case investigated is the population of students who participated. The case study is reflective in nature, looking back at various data points in an integrated way, providing opportunities to transform teaching and learning practices.

A. Data Sources and Analysis

In order to answer the question of how educators can develop new educational designs within the constraints of a large university based on this case, the analysis and results section will be divided into two parts: the design and implementation of the SDI (1) and the evaluation of the initiative (2).

Part 1 will be analyzed using the design tensions paradigm developed by Tatar [11]. The design tensions paradigm provides a concrete framework to understand design decisions in complex systems while emphasizing the balance of considerations in producing an entire system, especially the user group experience. Specifically, the design tensions highlight the vision of what *is* and what *ought to do be*, and illustrates the constraints of getting from one to the other. The use of design tensions is inspired by similar studies, where this framework was used as a productive tool to understand design and implementation challenges that exist in practice [6], [12].

Part 2 will look at student feedback and observations to evaluate the student perspective and the authors' reflections in relation to Part 1. It is also important to document the impact of the initiative, as it is part of the cost benefit assessment of developing new educational designs.

The data comes from three sources. In order to describe the challenges and solutions, we rely on the educator's descriptions of the process, in this case, the authors. Questionnaires from the students provide the measurement to investigate the student experience. In addition, we have a set of structured observations done by teaching assistants (TAs) in the last semester. These sources combined provide the grounds to explore how educators can develop new educational designs within the constraints of a large university.

B. Participants

The participants in this study includes both the students who were the target of the SDI as well as the educators and TAs involved in designing and implementing it. There were two educators in charge of the project (the authors) as well as 8-12 TAs for each semester. The number of TAs grew as the project grew. Most TAs were involved over several years which was very beneficial for the transfer of knowledge and improvements. As for the students, we invited a new class of approximately 300 students each year. Out of this population, 60-120 students showed up every week. It varied somewhat from week to week and semester to semester how big the turnout was. For ethical reasons, we had no way of counting 'unique users' every week, so we unfortunately do not know for sure how many of each 300 class attended at least one Study Day. Our estimation is that approximately 40% of the total student population attended at some point. In our attempts to increase attendance, several efforts were made to 1) reach students who were not there and 2) find out why and what we could do to reach them. Although, we could not seem to significantly improve the number of students attending we learned that the students who did not attend the SDIs reported that they did not see the need.

IV. Part 1: Design and implementation of the Study Day Initiative

The overall goal of the Study Day Initiative was to improve the academic and social learning environment of first-year computer science and engineering students. The solution we identified was to create a full day where students could come together and work on all their courses. This might seem like a modest idea; however, putting this into practice was not a simple task. Because of the university's overall organization, working across courses in this manner involves a complex network of administrators, course teachers, and support systems.

All courses are scheduled at the university level in order to ensure that the students' timetables are collision-free and allocated appropriate rooms. In practice, this means that educators at the program level, trying to design a pedagogical study week, do not have the mandate to schedule lectures or labs. With a little bit of luck and a good amount of negotiating with individual course teachers, we were able to clear six hours in a row each semester of the project. The next step was to recruit a number of teaching assistants who could support the students in all courses. This meant finding older students who were comfortable in a role with no insider information in any course. They would have to be able to answer questions about both introductory programming

TABLE II: The Study Day design tensions

VISION	**Is:** Students experience a fragmented learning environment	**Ought:** A holistic study experience, with alignment between courses, activities and support.
APPROACH	**Project drivers:** Centrally administrated planning based on courses, bureaucratic distance between educational innovators and decision-makers.	**Values:** Student-centered schedules, a community of educators and decision-makers.
PROJECT TENSIONS	Pedagogical intentions vs. structural constraints. Educator vs. system. Cost for educator vs. benefit for students.	
AS CREATED SCENARIO	Creating a student-centered learning environment for students.	

and computing, calculus, discrete mathematics, and scientific philosophy, which were the mandatory courses in the first semester. In practice, this meant the TAs had to take on a learning facilitator role, often helping students help each other or sitting down and doing the whole assignment with them. Lastly, we had to broadcast this initiative and ensure participation among the students. On the one hand, we emailed all students, used the courses we could influence to broadcast the initiative, and had student counselors communicate it in their channels. We also used another trick; serving food. We started each Study Day with a simple breakfast prepared by the TAs, hoping to motivate the students to get up in the morning and at the same time building social bonds. In summary, there where four items on the 'to-do-list' when implementing the SDI:

1) **Scheduling student time tables:** First, try to get the SDI on the formal schedule (Institution level). If that does not work, find the least full day and attempt to move all activities from that day to other days (Course level).
2) **Scheduling a room:** First, try to get a room booked for SDI through the central room reservation system (Institution level). If that does not work, use the rooms allocated to 'your' course, or negotiate with other courses in your department (Course level).
3) **Hiring and training TAs:** Using normal channels, aim to hire outgoing, proactive students with adequate performance in the central courses. Training the TAs includes supporting them in implementing the study day every week.
4) **Informing students:** Using whatever channels you have, make sure the students know when and where the SDI happens, as well as communicating that all students are welcome to work on any course. If you have the budget, serve food and coffee.

The first implementation of the SDI kicked off in 2017 and has been going strong since then. Every year, the educators in charge must be proactive and make sure the time and location schedule is in place. Weekly implementation of the SDI was mainly done by the TAs. The authors only had to be there in the first few weeks but tried to drop by as much as possible. Every week, the TAs set up the room, ordered and prepared the food, and most importantly, helped the students. During the study day, the TAs were instructed to go around to all the participants and interact with them individually, even if they did not request help. Furthermore, since the students could work on many different courses, the TAs developed an internal competency map, where they would send the most proficient TA to help students in any given problem. Every week, we would do a short stand-up meeting, where we discussed student challenges and decide on future interventions. Often these discussions were mostly about what questions and assignments the TAs struggled to help the students with and then designating one TA to do some research into that before next week.

From the educators' perspective, it was the design and preparation of this initiative that was challenging, not the weekly implementation. Using the design tensions framework, Table II outlines these constraints. The authors identified three tensions that provided the main hurdles for the innovative process. Firstly, the educators' pedagogic intentions were met with significant structural constraints (scheduling and room allocations). In attempting to navigate that situation, the educators were obstructed by a system that did not facilitate cross course designs. The system is aimed towards course teachers, and there is no support for educators operating mainly in the program level. For example, we reached out to the central coordinators for time- and room scheduling to get the SDI into the formal system; however, we were told that it would not be possible to add the SDI to the schedules because it was not a course. We then attempted to go through our local people, contacting course teachers, the department head and dean, but eventually were directed to the same central coordinators. This back and forth process went on for over a year, while we continued to adjust and negotiate the schedules in parallel so we could run the SDI. No matter how important the people forwarding us to the central coordinators were, and how adamant their emails were we never got SDI into the formal system. The closest we got was one cooperative scheduler who promised to try his best to keep one day cleared for our group of students. Again, we were reliant on individuals and their good will.

The cost of going around the system and negotiating with the structural constraints was, in this case, outweighed by the perceived benefit for the students, which will be further described in the next section. The created scenario here is a system in which the students' study experience is in the center of the design, opening up for holistic approaches such as the SDI.

V. Part 2: Evaluation and student experiences

In order to fully explore the effect of the initiative on the students, we also examine the student experience through the questionnaire and observational data. Questionnaires were distributed to the students during the last two weeks of the semester. TAs distributed the questionnaire on paper to all students participating and transferred the data into a digital format after the fact, providing total anonymity for the students. In addition, this ensured that all participating students answered the questionnaire. The questionnaire consisted of three sections: their participation and use of the traditional educational design elements (i.e., lectures, labs, TAs), their experience of the learning environment, and their use and evaluation of the SDI. In total, we received 136 responses over three years.

The analysis of the questionnaire data found that students' were very positive about the initiative. Two questions were asked about their study habits, one asking about their level of efficiency and one about their level of study compared to other days of the week. In addition, one question was asked about their level of motivation during SDI, if they had made a stronger connection to their peers and if they received the support they needed. The mean score for each of these variables was between 3.5-4.2, where 5 is the highest. As depicted in Fig. 2, there seemed to be little difference between genders. A chi-Squared test confirmed no statistically significant difference between male and female students on these variables (95% confidence interval). Taking a closer look at these answers, we found that 74% of the students reported that they were more effective during the study day compared to other days of the week, and 61% said they studied better. Furthermore, 72% reported that they were more motivated on the study day, and 66% said they made new friends. 90% reported getting the help they needed, and 98% wanted a similar initiative next semester.

It is evident from the evaluation results that the participating students were very content with the initiative, and according to their reports, we seemed to hit the mark. Students studied efficiently, made friends, and to a large extent, got the support they needed, indicating a good academic and social learning environment. It is also of interest to try and explain why this initiative seems to be successful among the students and what is actually going on. In addition, the 2019 implementation of the initiative included a structured observation performed by the TAs. At some point during the study day, TAs were asked to count the number of students in total, how many were working alone and in groups, what courses they were working on, as well as describe the general mood in the room. Every

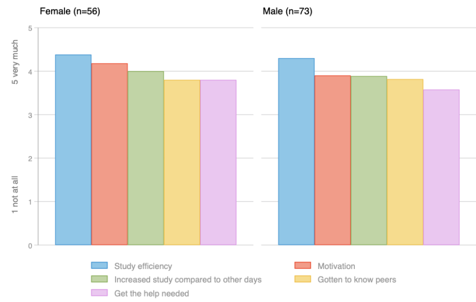


Fig. 2: Mean score of student experiences by gender.

week, there was a distribution of students working alone, in small groups, and in bigger groups. However, there seemed to be a link between this distribution and the course most students were studying. Some weeks more students were working on collaborative assignments, and therefore more students were working in groups. Furthermore, the room was set up with mostly group tables, so even though students were working individually, they were studying together with their peers. When it comes to what courses students were working on, that seemed to be largely driven by what assignments were due. TAs observed that the same students worked on different courses from week to week and reported what deadlines the students were talking about. Lastly, TAs were asked to give a report on the mood in the room, describing the efficiency and stress levels. With the exception of the last week, the stress level seemed to be moderate and the efficiency high. The last week was also the last week before exams, which probably accounts for the increased stress level.

Lastly, the questionnaire also included an open text question asking students to elaborate on what was good about the SDI and what needed improvement. Some students described why they participated with some very enlightening words. One student wrote about the availability of help and support:

> *When you're surrounded by three people who want to help you, study days are awesome!*

Another student also talked about the security of knowing you can get help if you need it:

> *Study days are brilliant because you just come here, and you know that there is someone here that can help you with everything and anything.*

Other positive comments were similar, while most of the improvements were directed towards the food services.

As the SDI project has continued from semester to semester over the last three years, these student perspectives have motivated the continuation. Seeing and hearing from the students how this initiative seems to have greatly improved their learning environment has encouraged us to keep organizing it. However, the overhead of negotiating time tables and room allocations ahead of each implementation is larger than it should be. We are currently attempting to transition the SDI from initiative to permanent activity, but this has proved challenging for the same reasons.

VI. Discussion

In this paper, we have explored the design and implementation of a Study Day Initiative for first-year computer science and engineering students and evaluated their experiences in order to shed some light on how educators can develop new educational designs to improve students' learning environment within the constraints of a large university. STEM higher education programs and majors are often built on the same foundation of courses, making the first-year complex both for students and educators. Students have courses in core engineering subjects such as mathematics and programming, in addition to their engineering disciplines. This often leads to large courses with students from different programs, which further creates a fragmented learning environment for the individual student. In the current case, the computing students had no courses where they were only computing students. In such circumstances, it becomes even more important to create time and spaces for students to come together and learn in a computing learning environment.

A. *Educational Design Tensions*

We can conclude that the SDI was a welcome improvement for the students. However, the overhead of implementing this for the educators within the constraints of large universities is not a sustainable organization for everyone. For one, this process was largely reliant on individual educators and their efforts to be proactive and negotiate solutions outside of the formal university structure. The role of the authors, in this case, was that we wanted to create a better learning environment for a specific class of students at one study program. In other words, we had to consider all their courses and were therefore operating at the program level (Table I). However, the educators were outsiders here, seeing as how we were on the course level, influencing only one of the courses. It turned out to be a major challenge that the initiative-taking educators were positioned at the program level, where their influence was limited. This was not a realization we had going into the project, and the driver was a vision of what ought to be. Retrospectively, the design tension analysis provided the terminology and framework to identify the issues at hand. Based on these experiences, we conclude that there is a need to move the perspective from courses to study programs in order to ensure the students' learning environment.

When it comes to the process of navigating the constraints of a large university, the educators had to manage three dimensions; time scheduling, physical space allocations, and availability of resources. In this specific case, the latter was the least complicated since the computing department has been part of a nationally funded center for education. Time and space allocations, however, were substantially more complicated. Similar to other larger universities, these processes are managed on the institution level, meaning the individual educator has limited to no influence on these outcomes. In this case, we were able to negotiate with individual course teachers to create room in the schedule and on campus to organize the SDI as planned. This is, however, not a sustainable solution in the long term.

B. *The Student-Driven Learning Environment and SDI*

The evaluation of the SDI indicates that the project was successful in enhancing their learning experience from the student perspective. The reason for the initiative was based on the fragmented learning environment created by the conditions of the educational design, and the holistic approach of the SDI met that challenge. The SDI provided a space where students could come together and work on assignments, learn and get support, combining many of the conditions driving learning identified in Fig. 1. According to the evaluation questionnaires, students were more efficient and motivated, as well as making stronger academic, social bonds. The SDI is not in itself the most revolutionary innovation; the notion of providing time, space, and support for students in the same place every week is at its essence very similar to many traditional designs. However, previous research on the effects of learning environments can provide some insights into why this relatively simple design seems to be so successful. Research has shown that how students perceive the learning environment and the way they approach their learning in relation to these perceptions are major intervening factors between teaching and learning outcomes [3], [13]. In this case, the SDI served as a stable, constant, and low threshold space where the students had positive learning experiences; hence their perception of the learning environment was improved. Without the SDI, the students would be on their own, filling the time between organized learning activities and finding help and support themselves.

There exist projects similar to the SDI both in design and effect on the students. One notable example is the redesign of the Electronic System Design and Innovation study program (ELSYS) at the Norwegian University of Science and Technology. Although this was a much larger initiative, one key element was the project-based course in the first semester, which consisted of one full day of integrated teaching activities and project-based learning [14]. The ELSYS-educators have reported success in creating an improved learning environment, specifically fostering self-efficacy and socialization [14], [15]. To the authors' knowledge, they have not reported on the design and implementation process, beyond the fact that the approach to create a holistic learning experience has been successful in the student learning process as well as a positive experience for the educators. What the ELSYS example has in common with the SDI is that the organized learning activities and the student-driven learning environment are integrated across courses and student-centered. One could also argue that learning communities [16] and program integrating courses [17] do much of the same for the students in creating learning environments across courses. Although, these examples were not focused on educational design constraints or learning environments directly, they suggest flipping the narrative of higher education design can benefit the students. Creating a learning environments centered on the student and their journey, instead of students having to from course to course, does improve the student experience.

C. Implications

Practical implications for the individual educator from this study is a tool for identifying the room for action in order to improve the student-driven learning environment: combining time, physical space, and support resources. Depending on the institution, these constructs may be easier or harder to identify and control than the case presented; however, the framework presented in Table I and II can provide an example and starting point.

For those in power, the key takeaway here is to flip the design process to put the learning environment in the center. The current case is an example from one institution; however, the general constraints of most larger universities are similar. Our recommendation is to move the cost of implementing educational innovations away from the individual educator by flipping the design process. If the educators on the course and program level are free to design holistic learning environments, combining the organized learning activities with the student-driven learning environment, the benefit of the student experience can be improved, as exemplified by this study. There should be a system in place to ensure that educators with good pedagogic intentions are able to implement interventions without having to negotiate and navigate outside the formal system.

D. Limitations

There are some important limitations with the SDI as well as the current study to consider. The biggest limitation of the SDI is that it benefits only the participating students, and there was never 100% attendance. As this was intended as a low threshold initiative, we were reluctant to enforce participation, even if we could. However, we are confident that information was not the main constraint for students who did not attend. Efforts were made to gain insight into reasons for not participating, but these students were hard to reach. Although the participating students seemed very content with the SDI, we do wonder if we were able to reach the students who 'needed it' the most.

When it comes to limitations in the research design, there are always concerns with rigor and generalizability with case studies. By describing the context and unit of analysis for the current case, we aim to reach an adequate level of analytic generalizability [9], [10], where other researchers and educators can extract information for their context.

VII. Conclusions and Future Work

Although there were many obstacles to being able to organize the Study Day Initiative, the effects of the intervention have been positive. The goal of the initiative was to ensure students had an appropriate learning environment, which was achieved. The room for action we had as educators was limited by scheduling challenges, restricted physical spaces, and constraints on resources; however, the pedagogical intention of creating an effective learning environment was successful by the metrics at hand. It is a fact that for many educators, the practical constraints often outweigh the pedagogical intentions. We believe this case study can illustrate to other educators and researchers how relatively small design changes can be influential and how one can effectively navigate the constraints of a large university.

Acknowledgements

The work presented in this paper was conducted at the Excited Centre for Excellence in IT Education. Excited receives public funding through DIKU, Norwegian Agency for International Cooperation and Quality Enhancement in Higher Education. We would like to thank the TAs who were involved in this project, their insight and problem-solving perspectives were an important reason for the success we have had.

References

[1] J. B. Biggs, "Approaches to the Enhancement of Tertiary Teaching," *Higher Education Research & Development*, vol. 8, no. 1, pp. 7–25, Jan. 1989.

[2] A. Lizzio, K. Wilson, and R. Simons, "University Students' Perceptions of the Learning Environment and Academic Outcomes: Implications for theory and practice," *Studies in Higher Education*, vol. 27, no. 1, pp. 27–52, Feb. 2002.

[3] K. Struyven, F. Dochy, S. Janssens, and S. Gielen, "On the dynamics of students' approaches to learning: The effects of the teaching/learning environment," *Learning and Instruction*, vol. 16, no. 4, pp. 279–294, Aug. 2006.

[4] NOKUT, "Studiebarometeret," 2019.

[5] M. Lorås and T. Aalberg, "First Year Computing Study Behavior: Effects of Educational Design," in *2020 IEEE Frontiers in Education Conference (FIE)*. Uppsala, Sweden: IEEE, 2020.

[6] J. Lock, B. Kim, K. Koh, and G. Wilcox, "Navigating the Tensions of Innovative Assessment and Pedagogy in Higher Education," *Canadian Journal for the Scholarship of Teaching and Learning*, vol. 9, no. 1, Apr. 2018.

[7] A. Gilbert, S. Tait-McCutcheon, and B. Knewstubb, "Innovative teaching in higher education: Teachers' perceptions of support and constraint," *Innovations in Education and Teaching International*, vol. 0, no. 0, pp. 1–12, Jan. 2020.

[8] K. Smith, "Cultivating innovative learning and teaching cultures: a question of garden design," *Teaching in Higher Education*, vol. 16, no. 4, pp. 427–438, Aug. 2011.

[9] R. K. Yin, *Case Study Research: Design and Methods*, 4th ed., ser. Applied social research methods. California: SAGE Publications, 2009, vol. 5.

[10] M. Bassey, *Case Study Research In Educational Settings*. Buckingham: Open University Press, May 1999.

[11] D. Tatar, "The Design Tensions Framework," *Human–Computer Interaction*, vol. 22, no. 4, pp. 413–451, Nov. 2007.

[12] R. Johnson, S. Severance, W. R. Penuel, and H. Leary, "Teachers, Tasks, and Tensions: Lessons from a Research-Practice Partnership," *Journal of Mathematics Teacher Education*, vol. 19, pp. 169–185, Apr. 2016.

[13] N. J. Entwistle, "Approaches to Learning and Perceptions of the Learning Environment: Introduction to the Special Issue," *Higher Education*, vol. 22, no. 3, pp. 201–4, 1991.

[14] P. Wallin, T. Bolstad, L. M. Lundheim, and P. T. M. Tybell, "Socialization and Curriculum Integration: Emergent premises in student experiences of a first-year electrical engineering course." SEFI Brussels, Belgium Brussel, Belgia, 2018.

[15] L. M. Lundheim, T. Bolstad, B. B. Larsen, P. T. M. Tybell, and P. Wallin, "A Whiteheadian Approach to Self Efficacy and Creative Confidence in Electrical Engineering." SEFI Brussels, Belgium Brussel, Belgia, 2018.

[16] A. Settle, J. Lalor, and T. Steinbach, “Evaluating a linked-courses learning community for development majors,” in *Proceedings of the 16th annual conference on information technology education*, ser. SIGITE ’15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 127–132.
[17] V. Kann and A.-K. Högfeldt, “Effects of a program integrating course for students of computer science and engineering,” in *Proceedings of the 47th ACM technical symposium on computing science education*, ser. SIGCSE ’16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 510–515.

Paper 8

Challenges Faced by Teaching Assistants in Computer Science Education Across Europe

Emma Riese, Madeleine Lorås, Martin Ukrop, Tomáš Effenberger

ITiCSE 2021

Authors' contributions: Riese and Lorås led the research process and paper writing. Riese, Lorås, Ukrop and Effenberger collected data from their respective institutions, and Riese and Lorås analysed the data.

 216

Challenges Faced by Teaching Assistants in Computer Science Education Across Europe

Emma Riese
KTH Royal Institute of Technology
Sweden
riese@kth.se

Madeleine Lorås
Norwegian University of Science and Technology
Norway
madeleine.loras@ntnu.no

Martin Ukrop
Masaryk University
Czech Republic
mukrop@mail.muni.cz

Tomáš Effenberger
Masaryk University
Czech Republic
tomas.effenberger@mail.muni.cz

ABSTRACT

Teaching assistants (TAs) are heavily used in computer science courses as a way to handle high enrollment and still being able to offer students individual tutoring and detailed assessments. TAs are themselves students who take on this additional role in parallel with their own studies at the same institution. Previous research has shown that being a TA can be challenging but has mainly been conducted on TAs from a single institution or within a single course. This paper offers a multi-institutional, multi-national perspective of challenges that TAs in computer science face. This has been done by conducting a thematic analysis of 180 reflective essays written by TAs from three institutions across Europe. The thematic analysis resulted in five main challenges: *becoming a professional TA*, *student focused challenges*, *assessment*, *defining and using best practice*, and *threats to best practice*. In addition, these challenges were all identified within the essays from all three institutions, indicating that the identified challenges are not particularly context-dependent. Based on these findings, we also outline implications for educators involved in TA training and coordinators of computer science courses with TAs.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; **Computer science education**.

KEYWORDS

Teaching assistants, TAs, challenges

ACM Reference Format:
Emma Riese, Madeleine Lorås, Martin Ukrop, and Tomáš Effenberger. 2021. Challenges Faced by Teaching Assistants in Computer Science Education Across Europe. In *26th Annual Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE '2021), June 26-July 1, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3430665.3456304

ITiCSE 2021, June 26-July 1, 2021, Virtual Event, Germany.

ACM ISBN 978-1-4503-8214-4/21/06.
https://doi.org/10.1145/3430665.3456304

1 INTRODUCTION

To teach and to learn computer science (CS) has been viewed as challenging and difficult by many previous studies [13]. Enrollment in CS courses at the university level has continued to increase [30]. Specifically, at the introductory level, courses with hundreds or even thousands of students are not rare. To manage these courses, teaching assistants (TAs, students who are employed to assist the faculty), are commonly used in CS [18]. However, the TA perspective is not fully explored, and previous research has mostly reported on experiences from a single institution, course, intervention, or TA training initiative [18]. To the authors' knowledge, no previous multi-national, multi-institutional study on TAs in CS have been conducted. This study aims to fill that gap by presenting and comparing data from three institutions in three different European countries. This paper aims to explore which main challenges TAs in CS face in their work and investigate whether or not the challenges differ between the three institutions. We define a challenge as something that is directly or indirectly described as an issue or difficulty. This paper is focused on two research questions (RQs):

(RQ1) Which challenges do TAs in computer science face?
(RQ2) Are the identified challenges similar or different across institutions and countries?

By increasing our knowledge and understanding of what our TAs experience as challenging in CS courses, instructors can make more informed decisions regarding their course structures and TA training. By providing a multi-national perspective on the perceived challenges, we aim to provide a more generalizable and nuanced picture relevant to the CS education community.

2 RELATED RESEARCH AND THEORY

TAs have been employed to assist faculty in many CS courses at multiple institutions [18]. Using TAs makes it possible to offer students individual guidance and feedback, also in large classes [18, 24, 25]. The TAs' work tasks differ between universities and courses, but often include conducting tutorials, assisting students during programming labs, developing course material, and assessing homework or exams [18]. Grading students' work, referred to as summative assessment [10, 33], is not a work task for all TAs. Some universities have strict rules stating that TAs are explicitly not allowed to grade students, which is only carried out by senior staff members or faculty members [16, 36]. On the other hand, TAs can be responsible for tutoring and providing feedback to the students throughout the course, referred to as formative assessment [10, 33]. The assessment carried out during a course should also be clearly linked to the intended learning outcomes and learning activities, referred to as constructive alignment [3]. At some institutions, the TAs take an active role in constructing learning activities and assessment tasks [1, 2, 27, 35], which entails that the TAs are also contributing to the course structure and content to some extent. Previous research has also shown that TAs who conduct assessments in a group setting achieve higher reliability [14], compared to in a solo-setting.

TAs have been found to be a contributing factor for student success [8]. Students can also view their TA as their main teacher within a course [28], that is, the person they have most interactions with and turn to for help. The fact that TAs are themselves also students has been argued to make the TAs more approachable than professors or senior lectures [9, 25]. Furthermore, the fact that the TAs were, often recently, enrolled in a similar course helps them relate to the students and foresee possible misconceptions [25]. However, previous research has also shown that some TAs view their students as their friends, which can make the TA role challenging and can cause conflicts of interest to arise when grading [27]. Both students [28] and TAs [8, 21, 26] have experienced that they are not always properly trained for the TA role and lack pedagogical skills and pedagogical content knowledge (PCK). PCK, as introduced by Shulman [31, 32] is described as a combination or overlap between content knowledge and pedagogical knowledge, that is, knowledge on how to teach the specific content. This framework was later extended by Mishra and Koehler [19] to also include a technology knowledge component. The technology knowledge dimension intersects with both the pedagogical knowledge and the content knowledge dimension, in what is referred to as technological pedagogical content knowledge (TPACK) [19]. Both the PCK and the TPACK framework have previously been applied to CS contexts, mainly in the K-12 teacher education settings [4, 7, 37].

Efforts to support the TAs by offering TA training have been reported and presented in a number of publications, such as [6, 11, 12, 16, 34, 35]. Furthermore, training has been reported to be an important factor in the TAs' professional development [17, 20]. One institution reported positive results with a team-teaching approach, where novice and experienced TAs were paired to work together when conducting tutorials [23]. The offering of introductory TA training has been suggested to bridge the gap between desired and actual competency among newly employed TAs [11]. The social environment and the intensity of lab sessions have also been found to affect job satisfaction among TAs [22]. Some institutions have reported on high interest among students to become TAs [29, 36], however, this is not the case at all institutions. How the TAs are recruited to the courses could also differ between universities [18], and a rubric to make the decision transparent and fair has been proposed in a previous study [15].

3 METHOD

In order to investigate which challenges TAs in CS face (RQ1), we collected reflection essays from TAs from three different institutions in three different European countries (RQ2). This paper does not aim to evaluate the three institutions' use of TAs, but understanding their characteristics is important in order to understand the results. The different institutions and their TA programs are therefore described in Table 1. At each institution, we asked TAs to reflect on their own practice by answering these questions:

> *"Describe an interesting situation or interaction you have experienced as an assistant. It can be something you found challenging, an ethical dilemma, or just something that has been on your mind. Reflect on how you handled the situation. What did you do well? What would you have done differently? Is there something you would like feedback on or questions you have?"*

At NTNU and KTH, the essays were collected as part of introductory TA training courses. At NTNU, the data was collected during 2018 and 2019, towards the ends of respective semesters, and at KTH, during the beginning of fall semester 2020. The essays were not graded on the content, but the TAs had to hand them in to complete their TA training course. At KTH, the TAs were also allowed to describe a fictive situation that they thought could occur since some of the TAs enrolled in the training course were very recently employed and had not yet gained much TA experience (but all of whom had been on the student side of TA-student interactions for years). At MUNI, the essays were collected during summer 2020 through the distribution of a digital survey asking the above-presented questions. The survey was distributed to all TAs enrolled in a voluntary TA training course in the previous four years and all the TAs of the second-largest undergraduate programming course. It was completely voluntary for all TAs to let their (anonymized) essays be part of this study, and informed consent was collected from all TAs. The data collection consist of 180 essays (119 from NTNU, 32 from KTH and 29 from MUNI). The essays were each half a page to a page long. A majority of the essays were written in the official languages of the given country, and a few were written in English. The essays from NTNU and KTH were analyzed in their original languages, while the essays from MUNI were first translated to English.

The essays were analyzed using a thematic analysis [5] aiming to identify common themes (the challenges) the TAs had written about. We followed the six steps outlined in [5], but with some adaption to the specific data set at hand. The analysis was carried out for the data from one institution at the time and then merged at the final stage of the analysis. For the set of essays from each institution, two researchers first coded all essays independently and summarized both the initial codes, and identified themes of their respective analysis. The two researchers then met to discuss and compare the findings of their independent analysis. This resulted in an agreement of the final themes and codes identified for each subset of data. The analysis was also conducted with some time between to minimize the interference of the previously found themes. Once the analysis of all three subsets was completed, we started to view the data as a complete set and merged the identified codes and themes. While doing so, we created a copy of the codes and themes that omitted which institution they originated from. This was done to not be influenced by the origin of the codes (since RQ2 aims to investigate potential differences). This data was, however, kept separate so we cold backtrack and validate the origin after this step was completed, and the writing up of the results began.

Table 1: Comparison of the three participating institutions, their CS departments and TA situation

		NTNU, Norway	KTH, Sweden	MUNI, Czechia
Institution	Overall	research-focused university, 8 faculties, 7 000 employees, 42 000 students	research-focused university, 1 faculty, 5 000 employees, 15 000 students	research-focused university, 9 faculties, 6 000 employees, 35 000 students
	CS department	3 000 students, approx. 500 TAs	4 400 students, approx. 150 TAs	2 000 students, approx. 150 TAs
	CS courses	7.5 ECTS, 50–3 600 students/course	3–9 ECTS, 20–250 students/course	2–8 ECTS, 10–700 students/course
	TAs' level	bachelor, master, doctoral students	bachelor, master, doctoral students	bachelor, master, doctoral students
TA position	Responsibilities	hold open lab hours, assess assignments (often oral), facilitate project work, (rarely) lecture	hold open lab hours, assess assignments (often oral), grade exams, conduct tutorials	conduct tutorials, hold open lab hours, grade assignments, grade exams, (rarely) lecture
	Assessment	cannot formally grade, but can assign pass/fail to assignments	grade assignments (pass/fail or A–F), the examiner is formally responsible	grade assignments (pass/fail or points), sometimes grade exams
	Payment	both teaching and preparation	both teaching and preparation	both teaching and preparation
	Recruitment	faculty-wide system based on grades and experience	course coordinators recruit independently based on their requirements	course lecturers recruit independently based on their requirements
TA training	Format	20 hours; several teaching blocks throughout the semester	6 hours; 3 online modules and 2 workshops before the semester	30 hours; weekly seminars during the whole semester
	Participation	only new TAs, mandatory	only new TAs, mandatory	any TAs, optional
	Compensation	paid for the time in training	paid for the time in training	3 ECTS credits for training

The final merging and formulation of the codes and themes was also carried out by both researchers independently, followed by a discussion resulting in the final themes. When this was completed, we revisited the essays and previous codes to validate our findings and backtrack the origin of themes. We also decided to cut out the parts of the data that were only about constraints or challenges caused by the COVID-19 pandemic since the data from NTNU was collected before the pandemic broke out.

4 RESULTS

From the thematic analysis, we identified results along two central dimensions corresponding to the posed research questions: *main challenges* and *institutional similarities and differences*. The main challenges that were identified from the analysis are presented in Figure 1 along with all sub-themes corresponding to each main challenge.

4.1 Main Challenges

Each main challenge represents a theme describing an area TAs found challenging. Each theme consists of a set of sub-themes, aggregated from the codes and represent recurring topics in the TAs reflections across the three included institutions.[1] Although the main challenges are separate issues, TAs would often mention several main challenges in one example, illustrating that there are also complex interconnections present between the identified themes.

[1] A complete overview of themes, codes, and exemplary quotes can be found at https://doi.org/10.18710/O8FCIK.

4.1.1 Becoming a professional TA. Being a professional was a topic many TAs reflected on as important but challenging. They described a mismatch of expectations between course teachers and students, especially in combination with unclear instructions. For example, course teachers would intend TAs to mainly facilitate group work, while the students expected debugging help and technical support. The TA community was seen as an important support network, but discovering unethical behavior of other TAs were sometimes mentioned as well. Multiple TAs noted areas they once found challenging but have since overcome, such as public speaking and personal interactions.

4.1.2 Student focused challenges. TAs reported on challenging experiences handling the vast diversity of students. From students with special needs, students dealing with personal problems to those unhappy about their assessment, or students working in groups. The level of content knowledge could also differ, from students who were very advanced to weaker students who were behind in the course. It was also described as challenging to meet unmotivated, passive, and unprepared students, as well as students who did not want help or were only focused on end results. The TAs also reflected on the uncomfortable interaction in situations with overly emotional, stressed, or upset students.

Multiple TAs reported that their own relationships with particular students sometimes made the interactions even more complicated. These included pre-existing friendly and romantic relationships, as well as the development of the TA-student relationship over the semester. Although many TAs highlighted the pedagogic benefits of having or developing strong relationships with their students, they also found it challenging to set boundaries and be professional, as is illustrated by the quote below.

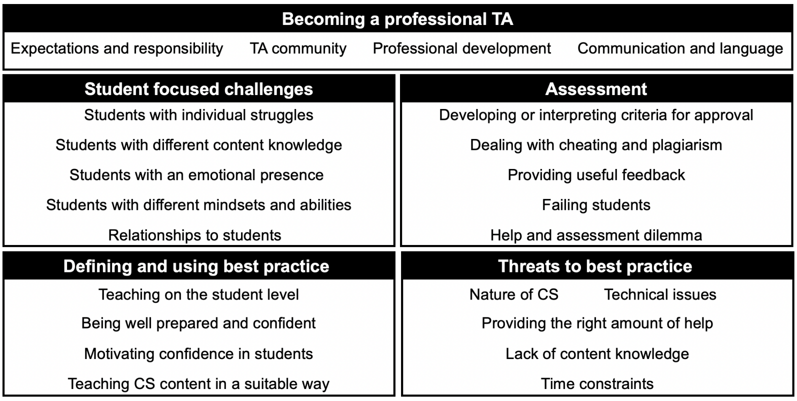


Figure 1: Main challenges identified in the thematic analysis

> *"It is not always so easy to grade their [close friends'] papers and I feel a conflict of interest while doing so. [...] It does not always feel fair that I spend so much time guiding my friends and not the rest of the students. It seems that the threshold for asking a friend through a FB-message is much lower than sending an email to an unknown teaching assistant."* [TA 48, NTNU]

4.1.3 Assessment. TAs found it challenging to develop and/or interpret passing/grading criteria, specifically mentioning determining what is "good enough", treating all the students fairly, and assessing effort and group work. Not surprisingly, failing a student was considered challenging: TAs reported feeling pressure from the students and uncomfortable announcing that the student failed, particularly in face-to-face interactions within hearing distance of other students. A further challenging aspect was cheating. Both determining if the submitted work was, in fact, plagiarized and acting on suspicions of unethical behavior were described as common challenges. In the formative paradigm, TAs reported that providing useful feedback was challenging, especially to students who were solely focused on the end result. Furthermore, TAs often described the challenge of giving both formative and summative feedback to the same students, sometimes even in the same session (guiding a struggling student and then immediately assessing if the submission was adequate).

4.1.4 Defining and using best practice. Overall, many TAs found it challenging to identify good practice for efficient teaching and learning CS and in applying it in practice. The topics mentioned included, for instance, visualizing code flow, writing pseudo-code, being creative, problem decomposition, and planning before coding. As a prominent sub-theme, the TAs found it challenging to teach at the students' level of understanding. First, the TAs needed to map the students' knowledge and understanding of the topic and then try to support them from there. Specifically, it was challenging for the TAs to formulate good questions and work with the student. Helping students reflect on their work and supporting good study habits was also brought up as wanted best practices but experienced as challenging to put into practice. TAs also mentioned to struggle with how to properly prepare for tutorials and lab sessions, feeling confident in teaching, making students feel confident and motivating them, and handling arising conflicts.

4.1.5 Threats to best practice. Although TAs reported on a plethora of effective pedagogical and didactic strategies to help students learn, they often went hand in hand with a challenge. For example, giving feedback to students' code, debugging code, and using pair programming. Several TAs reflected on the fact that CS was a new and difficult topic for many students, especially in the introductory courses. The following quote illustrated another aspect tied to the CS content: There are often multiple ways to solve an assignment, making it difficult to assess.

> *"As a TA I have met students who solve lab assignments in a very different way than the course teacher's solution. [...] It is a lot harder for a TA to assess these kinds of solutions. First the TA must interpret what the goal of the assignment was and what the central aspects in functionality and interaction were."* [TA 19, KTH]

In project-based courses, the programming language and technology used is sometimes up to the students to decide, resulting in TAs having to support topics outside their expertise. Furthermore, TAs also reported technical issues that stand in the way of learning the content (e.g., IDEs, operating system, version control).

Furthermore, TAs described being insecure about their lack of content knowledge, especially with new material, and, in general, just being worried about giving out the wrong information to their students. Providing the right amount of help was a commonly mentioned challenge as well. Concretely, resisting the urge to take over the student's keyboard, not pushing your own solutions, and balancing help, guidance, and teaching.

The threat to adopting best practice that most TAs reported on was, however, in-class time constraints. The time challenge involved dividing time evenly, prioritizing students who needed help, assessing students who like to present their solutions, and giving time to advanced students. The time predicament was visible throughout many identified themes, as illustrated by the quote below.

> *"In stressful situations, it can be easy to forget to take it slow to make sure the student understands fully what you are trying to help with. [...] In addition, I have a tendency to take over the students' keyboard when I feel the time pressure. One should always take the time to make sure the student has understood the problem you have been helping them with and adding some constructive feedback."* [TA 6, NTNU]

4.2 Institutional Similarities and Differences

In order to address the second research question, we need to examine the similarities and differences across institutions and countries. As described in Table 1, the way TAs work differs somewhat at MUNI compared to the other two institutions. At NTNU and KTH, TAs mostly help and support the students in open labs and determine if assignments are passed or failed. At MUNI, however, TAs have a more formal role, mostly conducting tutorials, assigning homework, and grading. Additionally, the TAs at KTH and MUNI plan and conduct tutorials, while TAs at NTNU have less responsibility in the planning and are just there to answer questions and conduct pass/fail assessment.

However, neither of these differences was prevalent in the main challenges presented in the previous section. While we found that the different structures lead to different specific situations, we also found that the core challenges remained the same across the studied institutions. TAs at NTNU and KTH would describe time management issues with students one-to-one or in the queue, while TAs at MUNI would discuss how to divide the time during a tutorial. An example becomes apparent with this reflection from a TA at MUNI:

> *"I tried from the beginning to explain the most important things, so that most of them [students] at least had a chance to 'catch the train', but it was at the expense of the time spent working on exercises."* [TA 23, MUNI]

Comparing the statement above to the second quote from NTNU in Section 4.1.5 about time management, it is evident that even though the specific situations were different, the core challenge was time constraints. At NTNU, the challenge was how to manage the time when helping students individually and to use best practices under stress. At MUNI, the challenge was how to divide the time in a tutorial between revising information from the lecture and working on exercises. These are both examples of the time constraints theme, but with different specific situations in different educational structures at the two universities.

Similarly, TAs at NTNU and KTH who did not formally decide on the grading experienced similar challenges regarding assessment as the TAs at MUNI who have that responsibility. Passing/failing assessments were described as similarly challenging, regarding assessing friends, setting the standard, and giving feedback, as actually setting a grade. Therefore, it can be concluded that all main challenges described were found to be similar across the examined institutions and countries, regardless of the education structure.

5 DISCUSSION

We have identified five main challenges that the TAs at NTNU, KTH and MUNI face: *becoming a professional TA*, *student focused challenges,assessment*, *defining and using best practice*, and *threats to best practice*. In many regards, these results confirm previous findings about the TAs' experiences. To begin with, our results strengthen the claim that TAs need help and support to develop within their role, which has also previously been shown [17, 23]. The TA community and social environment were shown to play a key role in that, which is also aligns with previous findings [22]. Communicating with the students and tutoring them is also a big part of the TAs' work tasks [17, 26] and our analysis found that this also comes with a whole set of challenges. Conducting assessments have previously been reported as difficult for TAs [14, 17, 26]. Furthermore, it has been found that TAs are both the tutor and grader to the same students [27], two roles that are non-trivial to combine.

The fact that TAs experience that they are approachable to their students [25], also comes with the downside of being too close to their students. A previous study reported that the TAs could view themselves as friends to their students [27], and our results extend on that. Our results show that personal relationships between TAs and their students exist and could be challenging for the TAs to handle. Time constraints have also been found to be hindering and challenging in previous studies [17, 26], which was confirmed by this study. The major challenges that have not received the same focus in previous studies are defining and using best practice. Although there have been studies reporting on training initiatives for TAs in CS [6, 11, 12, 16, 34, 35], little emphasis has been put on the CS specific best practices that we found the TAs also face. It is not surprising since CS is considered hard to both teach and learn [13]. Our findings, point towards challenges that span through the whole TPACK (technological, pedagogical and content knowledge) framework [19], including content knowledge, pedagogical knowledge, technology knowledge, and where they intersect. It is noteworthy, that TAs who have all been studying CS themselves experienced technical issues, and found the use of new software challenging. This has not been reported before and shows that it can not be taken for granted that you are an expert on all new technology simply because you are a CS TA.

The previous studies to which we have compared our results have, however, been conducted in small scales, isolated to one institution or one course. With our findings regarding RQ2, we could see that the identified challenges are present at multiple institutions in multiple countries. It is also interesting to note that these challenges were found to be similar across the institutions, despite different organizational structures. We would like to emphasize that TAs who take an active role in assigning homework or designing course material need to be aware of the intended learning outcomes, in order to be able to achieve constructive alignment [3] in the courses. The same applies for TAs who are tasked with using and interpreting passing/grading criteria. In order for the TAs to be successful and follow the course requirements, they need to understand the aim of each assignment they grade and each tutorial they conduct.

5.1 Implications

Based on these findings, we would like to highlight some implications for educators involved in TA training and coordinators of CS courses with TAs. The presented recommendations can also work as pointers for future research studies since whether or not they do have a positive effect on the TA experience remains to be investigated. Since this study shows that the identified challenges are similar between the three studied institutions, we would also argue that smaller scaled studies to address these can be valuable for the CS education community.

5.1.1 Be aware that best practice needs to be defined and spread. We can not assume that TAs have all the necessary content, pedagogical and technological knowledge needed as soon as they start. However, we can help TAs define best practices in CS education and share examples of how that can be implemented. Facilitating a social and supportive community is also believed to aid the sharing of best practices among TAs. As shown, the TA community can sometimes lead to the reproduction of unethical or unproductive practices, and we would therefore argue that experience is not enough. Formal TA training that includes illustrative examples is also needed.

5.1.2 Acknowledge threats to best practice and address them. Being aware of the main threats to best practice could be seen as a first step in overcoming these. Course coordinators have the power to make informed decisions to minimize these threats. For instance, one of the identified threats was working under time constraints. If you expect your TAs to be able to give the students detailed feedback and carefully guide them through a difficult programming problem, the TAs need to have sufficient time to do so. Another example is that material and instructions for assessment need to be clear, and even if they are, the TAs might still need additional help interpreting and using them.

5.1.3 Dare to discuss ethical dilemmas and provide guidelines. Dealing with ethical dilemmas, such as suspicions of plagiarism or deadline extensions for desperate students, is something that needs to be addressed and discussed with the TAs. The TAs need to know how to handle such situations and take actions based on knowledge, not feelings. Even though this might seem trivial to an experienced course coordinator, it is not trivial to all TAs.

5.1.4 Recognize the student-TA relationship as unique. The social aspect of being a TA and the interaction with students are crucial parts of the TAs' work. Concurrently, the student-TA relation was experienced as a major challenge for many TAs in this study. The TAs need to be equipped with tools and techniques to be able to overcome these challenges sufficiently. Some of the described challenges were related to general pedagogic knowledge, such as motivating students and handling a diverse student group. Other challenges come from the fact that TAs have other types of relationships with their students (for instance, are friends with or even romantically involved with their students). These are believed to be specific to TAs and need to be addressed as such. To the faculty that train TAs, we would recommend addressing these and providing the TAs with a good foundation on how to handle specific situations.

5.2 Threats to Validity and Limitations

In this research, we have used a qualitative method, with a large sample size (180 participants). Nevertheless, it is important to note that the sample size between the three institutions differed, which could have had an impact when comparing the three data sets to each other. If a theme was not present in the data set for an institution, that does not necessarily imply that the TAs have not experienced that challenge since we did not ask the TAs to name all challenges they ever encountered. However, we did not find any major differences between the institutions. This finding both strengthens the claim that the identified themes were truly the main challenges across the institutions and that even the smaller sample sizes (29–32) were sufficient to capture these through asking open-ended questions. It should also be noted that the open-ended questions did not explicitly ask the TAs to name their main challenges but rather to describe a situation or interaction and reflect upon it. It is, of course, a possibility that the TAs would have written something else if asked explicitly, but this method was chosen to give us the teaching contexts and enable the TAs to describe a challenging situation without having to pinpoint a specific challenge. The data is also limited by what the TAs were comfortable sharing. All collected data are also self-reported by the TAs – asking the students and course coordinators or lecturers for their view on these challenges would be an interesting additional input and possibility to validate the results further.

A limitation of the setup of this study is that we only studied three institutions within Europe, and the generalizability to other institutions is not investigated. In this research, we did also not take into account how much experience the TAs have had prior to writing their reflections, which could impact what they wrote in their essays. The data were not collected with prior experience as a controlled variable. At KTH, a majority of the TAs were new TAs, writing these essays at the beginning of the semester, at NTNU the TAs wrote these essays at the end of their first semester, and at MUNI it was more scattered. The unified results do suggest that the identified challenges are found across institutions and among TAs with different long experience, but we can not make any claims on to which degree experience played a role from this study. The presented results should also not be seen as a complete list of challenges that TAs in CS face. That was also not the aim of this research, and the results should rather be seen as a list of main challenges identified across the three studied institutions. In this study, all steps in the thematic analysis were carried out by two researchers independently, followed by a discussion resulting in an agreement. This rigorous process strengthens the trustworthiness of the results.

6 CONCLUSIONS

In this study, we have identified five main challenges that TAs in CS face by analyzing 180 reflective essays from TAs from three different institutions in three different countries. We also found that the identified challenges (*becoming a professional TA*, *student focused challenges*, *assessment*, *defining and using best practice*, and *threats to best practice*) were present in the essays from all three intuitions. In fact, no major differences were found between the institutions, despite the different organizational setups. We conclude by emphasizing that TA training and support are needed in order to assist the TAs in overcoming these challenges.

REFERENCES

[1] Christine Alvarado, Mia Minnes, and Leo Porter. 2017. Micro-Classes: A Structure for Improving Student Experience in Large Classes. In *Proceedings of the 2017 ACM*

SIGCSE Technical Symposium on Computer Science Education (Seattle, Washington, USA) *(SIGCSE '17)*. ACM, New York, NY, USA, 21–26. https://doi.org/10.1145/3017680.3017727

[2] Maureen Biggers, Tuba Yilmaz, and Monica Sweat. 2009. Using Collaborative, Modified Peer Led Team Learning to Improve Student Success and Retention in Intro Cs. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (Chattanooga, TN, USA) *(SIGCSE '09)*. ACM, New York, NY, USA, 9–13. https://doi.org/10.1145/1508865.1508872

[3] John Biggs. 1996. Enhancing teaching through constructive alignment. *Higher education* 32, 3 (1996), 347–364.

[4] Ofra Brandes and Michal Armoni. 2019. Using Action Research to Distill Research-Based Segments of Pedagogical Content Knowledge of K-12 Computer Science Teachers. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* (Aberdeen, Scotland Uk) *(ITiCSE '19)*. ACM, New York, NY, USA, 485–491. https://doi.org/10.1145/3304221.3319773

[5] Virginia Braun and Victoria Clarke. 2006. Using Thematic Analysis in Psychology. *Qualitative Research in Psychology* 3, 2 (Jan. 2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[6] Rebecca Brent, Jason Maners, Dianne Raubenheimer, and Amy Craig. 2007. Preparing undergraduates to teach computer applications to engineering freshmen. *2007 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports* (2007), F1J–19–F1J–22.

[7] Malte Buchholz, Mara Saeli, and Carsten Schulte. 2013. PCK and Reflection in Computer Science Teacher Education. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education* (Aarhus, Denmark) *(WiPSE '13)*. ACM, New York, NY, USA, 8–16. https://doi.org/10.1145/2532748.2532752

[8] S. E. DeChenne, N. Koziol, M. Needham, and L. Enochs. 2015. Modeling Sources of Teaching Self-Efficacy for Science, Technology, Engineering, and Mathematics Graduate Teaching Assistants. *CBE Life Sci Education* 14, 3 (2015), 347–364. https://doi.org/10.1187/cbe.14-09-0153

[9] Adrienne Decker, Phil Ventura, and Christopher Egert. 2006. Through the Looking Glass: Reflections on Using Undergraduate Teaching Assistants in CS1. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA) *(SIGCSE '06)*. ACM, New York, NY, USA, 46–50. https://doi.org/10.1145/1121341.1121358

[10] Dante D. Dixson and Frank C. Worrell. 2016. Formative and Summative Assessment in the Classroom. *Theory Into Practice* 55, 2 (2016), 153–159. https://doi.org/10.1080/00405841.2016.1148989

[11] Francisco J. Estrada and Anya Tafliovich. 2017. Bridging the Gap Between Desired and Actual Qualifications of Teaching Assistants: An Experience Report. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education* (Bologna, Italy) *(ITiCSE '17)*. ACM, New York, NY, USA, 134–139. https://doi.org/10.1145/3059009.3059023

[12] Jeffrey Forbes, David J. Malan, Heather Pon-Barry, Stuart Reges, and Mehran Sahami. 2017. Scaling Introductory Courses Using Undergraduate Teaching Assistants. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. ACM, New York, NY, USA, 657–658. https://doi.org/10.1145/3017680.3017694

[13] Mark Guzdial. 2015. Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics* 8, 6 (2015), 1–165.

[14] Brian Harrington, Marzieh Ahmadzadeh, Nick Cheng, Eric Heqi Wang, and Vladimir Efimov. 2018. TA Marking Parties: Worth the Price of Pizza?. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (Espoo, Finland) *(ICER '18)*. ACM, New York, NY, USA, 232–240. https://doi.org/10.1145/3230977.3230997

[15] Dan Leyzberg, Jérémie Lumbroso, and Christopher Moretti. 2017. Nailing the TA Interview: Using a Rubric to Hire Teaching Assistants. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education* (Bologna, Italy) *(ITiCSE '17)*. ACM, New York, NY, USA, 128–133. https://doi.org/10.1145/3059009.3059057

[16] Madeleine Lorås. 2020. From teaching assistants to learning assistants – lessons learned from learning assistant training at Excited. *Læring om læring* 4, 1 (2020).

[17] Farshid Marbouti, Kelsey Joy Rodgers, Hyunyi Jung, Alena Moon, and Heidi A. Diefes-Dux. 2013. Factors That Help and Hinder Teaching Assistants' Ability to Execute Their Responsibilities. In *2013 ASEE Annual Conference & Exposition*. ASEE Conferences, Atlanta, Georgia, 23.588.1 – 23.588.18. https://doi.org/10.18260/1-2--19602

[18] Diba Mirza, Phillip T. Conrad, Christian Lloyd, Ziad Matni, and Arthur Gatin. 2019. Undergraduate Teaching Assistants in Computer Science: A Systematic Literature Review. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. ACM, New York, NY, USA, 31–40. https://doi.org/10.1145/3291279.3339422

[19] Punya Mishra and Matthew J Koehler. 2006. Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers college record* 108, 6 (2006), 1017–1054.

[20] Alena Moon, Hyunyi Jung, Farshid Marbouti, Kelsey Rodgers, and Heidi Diefes-Dux. 2013. Undergraduate and graduate teaching assistants' perceptions of their responsibilities - Factors that help or hinder. In *2013 IEEE Frontiers in Education Conference (FIE)*. IEEE Conferences, 1576–1578. https://doi.org/10.1109/FIE.2013.6685103

[21] Valbona Muzaka. 2009. The niche of Graduate Teaching Assistants (GTAs): perceptions and reflections. *Teaching in Higher Education* 14, 1 (2009), 1–12. https://doi.org/10.1080/13562510802602400

[22] Elizabeth Patitsas. 2012. A Case Study of Environmental Factors Influencing Teaching Assistant Job Satisfaction. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research* (Auckland, New Zealand) *(ICER '12)*. ACM, New York, NY, USA, 11–16. https://doi.org/10.1145/2361276.2361280

[23] Elizabeth Patitsas. 2013. A Case Study of the Development of CS Teaching Assistants and Their Experiences with Team Teaching. In *Proceedings of the 13th Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '13)*. ACM, New York, NY, USA, 115–124. https://doi.org/10.1145/2526968.2526981

[24] Stuart Reges, John McGrory, and Jeff Smith. 1988. The Effective Use of Undergraduates to Staff Large Introductory CS Courses. In *Proceedings of the Nineteenth SIGCSE Technical Symposium on Computer Science Education* (Atlanta, Georgia, USA) *(SIGCSE '88)*. ACM, New York, NY, USA, 22–25. https://doi.org/10.1145/52964.52971

[25] Yanyan Ren, Shriram Krishnamurthi, and Kathi Fisler. 2019. What Help Do Students Seek in TA Office Hours?. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. ACM, New York, NY, USA, 41–49. https://doi.org/10.1145/3291279.3339418

[26] Emma Riese. 2018. Teaching Assistants' Experiences of Lab Sessions in Introductory Computer Science Courses. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, San Jose, CA, USA, 1–5. https://doi.org/10.1109/FIE.2018.8659243

[27] Emma Riese and Viggo Kann. 2020. Teaching Assistants' Experiences of Tutoring and Assessing in Computer Science Education. In *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE, Uppsala, Sweden, 1–9. https://doi.org/10.1109/FIE44824.2020.9274245

[28] Emma Riese and Stefan Stenbom. 2020. Experiences of Assessment in Introductory Programming From the Perspective of NonComputer Science Majors. In *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE, Uppsala, Sweden, 1–9. https://doi.org/10.1109/FIE44824.2020.9274060

[29] Eric Roberts, John Lilly, and Bryan Rollins. 1995. Using Undergraduates as Teaching Assistants in Introductory Programming Courses: An Update on the Stanford Experience. In *Proceedings of the Twenty-Sixth SIGCSE Technical Symposium on Computer Science Education* (Nashville, Tennessee, USA) *(SIGCSE '95)*. ACM, New York, NY, USA, 48–52. https://doi.org/10.1145/199688.199716

[30] Linda J. Sax, Kathleen J. Lehman, and Christina Zavala. 2017. Examining the Enrollment Growth: Non-CS Majors in CS1 Courses. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. ACM, New York, NY, USA, 513–518. https://doi.org/10.1145/3017680.3017781

[31] Lee Shulman. 1987. Knowledge and teaching: Foundations of the new reform. *Harvard educational review* 57, 1 (1987), 1–23.

[32] Lee S Shulman. 1986. Those who understand: Knowledge growth in teaching. *Educational researcher* 15, 2 (1986), 4–14.

[33] Maddalena Taras. 2005. Assessment - Summative and Formative – Some Theoretical Reflections. *British Journal of Educational Studies* 53, 4 (2005), 466–478. https://doi.org/10.1111/j.1467-8527.2005.00307.x

[34] Martin Ukrop, Valdemar Švábensk; y, and Jan Nehyba. 2019. Reflective Diary for Professional Development of Novice Teachers. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. ACM, New York, NY, USA, 1088–1094. https://doi.org/10.1145/3287324.3287448

[35] Martin Ukrop, Valdemar Švábenský, and Imrich Nagy. 2020. Teaching Lab: Training Novice Computer Science Teachers. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (Trondheim, Norway) *(ITiCSE '20)*. ACM, New York, NY, USA, 561. https://doi.org/10.1145/3341525.3393967

[36] Arto Vihavainen, Thomas Vikberg, Matti Luukkainen, and Jaakko Kurhila. 2013. Massive Increase in Eager TAs: Experiences from Extreme Apprenticeship-Based CS1. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education* (Canterbury, England, UK) *(ITiCSE '13)*. ACM, New York, NY, USA, 123–128. https://doi.org/10.1145/2462476.2462508

[37] Rebecca Vivian and Katrina Falkner. 2019. Identifying Teachers' Technological Pedagogical Content Knowledge for Computer Science in the Primary Years. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. ACM, New York, NY, USA, 147–155. https://doi.org/10.1145/3291279.3339410