Doctoral theses at NTNU, 2024:126

Majid Rouhani

An Online Professional Development Framework for In-service Teachers

Challenges and Opportunities of Learning and Teaching Programming

Doctoral thesis

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

Majid Rouhani

An Online Professional Development Framework for In-service Teachers

Challenges and Opportunities of Learning and Teaching Programming

Thesis for the Degree of Philosophiae Doctor

Trondheim, May 2024

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 978-82-326-7846-4 (printed ver.)
ISBN 978-82-326-7845-7 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2024:126

Printed by NTNU Grafisk senter

Abstract

In this thesis and its related publications, we explore the problem of designing, implementing, and refining a fully online professional development framework. The work aims to meet the needs of in-service teachers who are required to learn programming to teach it to pupils in secondary schools. We carried out a systematic mapping of the research on the professional development of in-service teachers in CS, designed and evaluated an online flexible teachers' professional development program that promotes active learning and motivates teachers to learn and teach programming. Additionally, we investigated and reported on teachers' attitudes, perceived challenges, and opportunities. The study contributes to the field of pre-college CS education.

The program is designed to cater to in-service teachers from diverse backgrounds and equip them with programming skills to integrate into their courses. The study focuses on the learning process of the participants and how they plan to apply their newfound knowledge in their classrooms. The research aims to provide insights into practical strategies for teaching programming to teachers and the benefits and challenges of incorporating programming into secondary school curriculum.

In three implementation cycles, each spanning one academic year, we combined 26 interviews and analysis of 849 reflection notes using qualitative and mixed methods. The abstracts of 7 251 papers were analyzed, and 206 were included in the final literature mapping. More than 450 participants were involved in the program, with a completion rate of 95% in total. The number of participants increased from 87 in-service teachers in the first cycle to nearly 200 in the third run. The refinements implemented during each iteration were based on a continuous feedback loop and an active effort to create a feeling of belonging, social interaction, and peer engagement.

The research resulted in (1) the development of a professional development framework, (2) a bridge model for connecting training and practice by supporting in-service teachers in creating individualized lesson plans, (3) a set of challenges and opportunities related to the learning to program from an in-service teacher's perspective, (4) a set of stumbling and stepping stones (challenges and enablers) when learning to teach programming from an in-service teachers' perspective, and (5) a list of guidelines provided for the implementation purposes.

Sammendrag

I denne avhandlingen og dens tilnyttede publikasjoner utforsker vi problemet med å designe, gjennomføre og evaluere et nettbasert rammeverk for faglig utvikling. Arbeidet tar sikte på å møtekomme behovene til lærere som er i full job og som er pålagt å lære programmering for å lære det til elever i ungdom- og videregående skolen. Vi gjennomførte en systematisk kartlegging av forskningen på faglig utvikling av lærere innen informatikk, designet og evaluerte et fullstendig nettbasert og fleksibelt faglig utviklingsprogram som kan fremme aktiv læring og motivere lærere til å lære og undervise i programmering. I tillegg undersøkte vi lærernes holdninger, utfordringer og muligheter programmering kan gi dem. Studien bidrar til feltet for informatikkutdanning på videregående nivå.

Programmet er utformet for å imøtekomme yrkesaktive lærere med ulike bakgrunner og utruste dem med programmeringsferdigheter som kan integreres i deres undervisning. Studien fokuserer på deltakernes læringsprosess og hvordan de planlegger å anvende den nye kunnskapen i klasserommet. Forskningen tar sikte på å gi innsikt i effektive strategier for å undervise programmering til lærere og fordelene og utfordringene med å integrere programmering i læreplanene i skolen.

I tre implementeringssykluser, som hver spenner seg over ett akademisk år, kombinerte vi 26 intervjuer og analyser av 849 refleksjonsnotater ved bruk av kvalitative og kombinerte metoder. Abstraktene fra 7 251 artikler ble analysert, og 206 ble inkludert i litteraturkartleggingen. Mer enn 450 deltakere var involvert i programmet, med en gjennomføringsgrad på totalt 95%. Antall deltakere økte fra 87 lærere i første syklus til nesten 200 i tredje syklus. Forbedringene som ble implementert under hver iterasjon var basert på en kontinuerlig tilbakemeldingssløyfe og en aktiv innsats for å skape en følelse av tilhørighet, sosial interaksjon og kollega samarbeid.

Forskningen resulterte i (1) utviklingen av et opplæringsrammeverk, (2) en bromodell for å koble opplæring og praksis ved å støtte etterutdannede lærere i å lage individuelle undervisningsopplegg, (3) et sett med utfordringer og muligheter når de lærer å programmere fra et lærersperspektiv, (4) et sett med utfordringer og muligheter når man lærer å undervise i programmering fra et lærersperspektiv, og (5) en liste over retningslinjer gitt for implementeringsformålene.

Preface

As the need for computational thinking and programming skills grows across industries, there is an increasing demand to integrate computer science education into school curricula worldwide. However, many secondary school teachers currently teaching these subjects have not received proper training in computer science pedagogy. This presents a significant challenge as unprepared educators may struggle to effectively engage students and teach technical concepts.

Quality professional development for in-service teachers is crucial to address this issue. Traditional one-size-fits-all training models are often ineffective due to the diverse needs of practicing educators. There is a clear need for flexible, personalized programs that can be easily accessed. Advances in online learning technologies now make it possible to design online education tailored to individual teachers' circumstances.

This thesis documents a design science research study that aimed to develop and evaluate an online flexible professional development framework for in-service computer science teachers. Through iterative design, implementation and mixed methods evaluation with over 450 participating teachers, key elements for an impactful program were identified.

The findings offer valuable guidance for stakeholders seeking to advance computer science education through long-term teacher development. By incorporating project-based learning, collaboration, flexibility, scaffolding and ongoing support, online programs can actively engage educators and motivate skills acquisition. This equips teachers to confidently bring programming into diverse classrooms through effective pedagogical approaches.

While the research project has some limitations, such as the focus on a single country, it provides valuable insights into the challenges and opportunities associated with teaching programming in secondary schools. The research project also suggests directions for future research, such as exploring the effectiveness of the training framework in other contexts and with larger groups of participants.

I started working on this project in partnership with KOMPiS[1] and Excited[2] (Centre for Excellence IT Education). KOMPiS receives public funding through UDIR[3], the Directorate of Education, which is the executive agency for the Ministry of Education and Research. Excited receives public funding through HK-dir[4], Norwegian Agency for International Cooperation and Quality Enhancement in Higher Education.

[1] https://www.ntnu.no/su/kompis
[2] https://www.ntnu.edu/excited
[3] https://www.udir.no/in-english/
[4] https://hkdir.no/

Acknowledgements

I am incredibly grateful for the opportunity to conduct this study, and I must express my deep appreciation for all the individuals who have supported me throughout this journey. It has been an exhilarating experience working with the program participants and my colleagues, and I am truly thankful for the unwavering support I have received. Working intensively on a project over an extended period requires ample attention from both professional and family life. Therefore, I am immensely grateful for the understanding and encouragement I have received from my wife, Shohreh, and the rest of my family - Shirin, Shervin, Nadia, Adrian, Hailey, Mia, and Noah. Their patience and unwavering support have been a constant source of inspiration for me, and I am truly thankful for their presence in my life.

I would also like to extend my gratitude to my colleagues, Monica Divitini and Ali Alsam, for their invaluable support during the project. Additionally, I am deeply appreciative of the contributions made by my co-authors - Vojislav Vujosevic, Sondre Stai, Hege Anette Olstad, Jørgen Thorsnes, Veronica Farshchian, Victor Jørgensen, Atle Olsø, Miriam Lillebo, and Amir Massoud Hashemi. Their expertise and collaboration have been instrumental in the success of this project. Furthermore, thanks to Elise Klæbo Vonstad for being an instructor in the program. I also want to express my gratitude to all my colleagues from the Department of Computer Science, the AIT group, and the participants, teacher assistants, and master students involved in the program. Thanks to the AIT group leaders, Monica Storvik and Geir Ove Rosvold, for all their support and excellent leadership skills. Their contributions and support have been invaluable.

I must also acknowledge the significant role played by NTNU Videre (NTNU's Centre for Continuing Education and Professional Development) in the development of the program. I am particularly grateful to Gunvor Viker Singsaas for the outstanding support and assistance with the program's administrative tasks. Additionally, I would like to express my appreciation for the exceptional collaboration with the KOMPiS group, led by Elin Bø Morud and coordinated by Marianne Leirvik Bragstad. I am also thankful for the support and encouragement I have received from everyone at NTNU whom I have had the pleasure of meeting over the years. The NTNU system is filled with individuals who are not only supportive but also encouraging, and I am truly grateful for their presence in my professional journey.

Contents

Figures

Tables

Code Listings

Acronyms

CAS computing at school. 81

CHAT Cultural-Historical Activity Theory. 91

CK content knowledge. 17

CoI community of inquiry. 20, 21, 42

CoP community of practice. 91

COVID-19 coronavirus disease of 2019. 86

CS computer science. iii, xv, 2, 4, 6, 9, 11, 13, 38, 69, 70, 77

CT computational thinking. 70, 78

DigComp digital dompetence. 1

DS design science. 37

DSRM design science research methodology. 37, 42, 49, 51

OLT Online learning theory. 5

OPDF online professional development framework. xii, xv, 27–33, 42, 87

PCK pedagogical content knowledge. 17

PD professional development. iii, 1, 4–6, 8–10, 13, 52, 54, 69, 70, 73, 77

PjBL project-based learning. xxiii, 76

PK pedagogical knowledge. 17

RQ research question. xi, 5–11, 72, 88

SM systematic mapping. 54

STEM science, technology, engineering and math. 36

TA teaching assistant. 60, 76, 96

TCK technological content knowledge. 18

TK technological knowledge. 17, 18

TPACK technological, pedagogical, and content knowledge. 17, 18, 26, 42

TPD teacher professional development. xi, 1, 2, 7

Glossary

active learning is a type of education that engages students with course content via conversations (asking/answering questions), conducting lab experiments, case studies, problem-solving, and other activities [1].. 95

artifact This study's artifact corresponds to the training program designed and evaluated for in-service teachers.. 37, 64, 76, 85, 86

bridge model The bridge model [2] use PjBL in a flexible learning trajectory course to create an artifact (lesson plan) that can be used directly in teachers' classes to teach programming. The model consists of four phases: (1) preparation, (2) specialization, (3) realization, and (4) rectification.. 51

cognitive presence is the capacity of students to build and validate views via persistent thinking and debate [3].. 19, 20

digital content creation Defined by Carretero *et al.* [4] and consists of several competency areas: developing digital content, integrating and re-elaborating digital content, copyright and licences, and programming.. 1

digital transformation Digital transformation in education is about incorporating new technology in the classroom, changing assessment forms, and following up on the student's development. The changes aim to improve general learning outcomes.. 1

educator A participant of the training program or a person teaching at secondary school.. 4, 5, 8, 10, 11, 13

flexible learning trajectory In this context, a flexible learning trajectory program is defined as a program where participants select parts of the syllabus most relevant to their practice [5].. xv, 7, 13, 57, 60, 61, 92

in-service teacher Post-graduate teachers who teach in primary- and secondary schools. iii, 1, 2, 5, 6, 8, 10–13

instructor The term "instructor" refers to the professor(s) of the training program being discussed in this thesis. When referring to participants of the program, the terms "teacher," "educator," "in-service teacher," "pre-college teacher," "participants," and "students" are used. Further, we use the term "pupils" when discussing pre-college students.. 60, 76, 77, 85, 95, 96

knowledge promotion The Norwegian school curriculum reform effort The Knowledge Promotion, implemented in 2006, indicated a move from a content-oriented curriculum to a competency-oriented curriculum [6].. 1

lesson plan A lesson plan in this thesis is to be understood as a lesson in programming created by a teacher customized for his/her class.. xxiii, 5, 76, 88

participant A student in the training program.. iii, 9, 13

pupil A learner at pre-college educational institution.. 9, 11, 13

self-efficacy refers to confidence in one's ability to plan and execute the actions necessary to achieve the desired outcome [7].. 5, 10, 13, 81, 82, 84

social presence is "the ability of participants to identify with the community (e.g., course of study), communicate purposefully in a trusting environment, and develop interpersonal relationships by way of projecting their individual personalities" [8, p. 252]. 19–21

student A learner/participant in the training program.. 1–3, 7, 9, 13

teaching presence is the process of planning and assisting cognitive and social processes in achieving meaningful and educationally worthwhile learning results [3].. 19, 21

webinar The term "webinar" is a combination of "web" and "seminar." A webinar is a digital meeting that is watched only by online participants[5].. 35, 53

[5]https://www.webinar.nl/en/webinars/what-is-a-webinar/

Part I

Synopsis

Chapter 1

Introduction

Digital transformation is one of the most significant social shifts in modern times [9]. This transformation causes many occupations to disappear, new jobs and sectors to emerge, and work requirements to change. As part of its digital transformation policy, the Norwegian government's vision is that everyone, regardless of geographical location, age, and background, should have access to education at schools, colleges, and universities. To meet its vision, the government seeks to enhance access to high-quality educational programs that are flexible, decentralized and tailored to the diverse needs of the population [10]. Similarly, governments around the globe are preoccupied with enhancing educational quality and professional development to meet the challenges imposed by digital transformation [11, 12].

The 'knowledge promotion' is a Norwegian school reform that covered the entire primary and secondary school and was launched by the government in 2006. It represented a shift from a content-oriented curriculum to a competence-oriented curriculum. In contrast to primarily concentrating on what students should learn, a competency-based curriculum [6] emphasizes the multifaceted qualities of a learning process (knowledge, skills, and attitudes) to be applied by students. An essential change in this reform was the inclusion of digital competency. Like the foundational skills of reading, writing, calculating, and oral communication, digital competencies were incorporated as integral components of education [13]. The DigComp Framework [4] defines five dimensions of digital competence: Information and data literacy, Communication and collaboration, Digital content creation, Safety, and Problem-solving. The DigComp Framework [4] defines programming as a sub-area of "digital content creation."

Norwegian schools adopted the new reform in 2020 and integrated programming into existing subjects such as mathematics and science. Consequently, a demand for professional development (PD) of in-service teachers of programming arose. To meet this demand, it is necessary to strengthen the teachers' programming ability so that they can include it in their teaching practice.

Teacher professional development (TPD) plays a crucial role in equipping educators with the knowledge and abilities required in a rapidly changing educational landscape. TPD programs aim to enhance teachers' content knowledge and pedagogical approaches to keep pace with emerging technologies and skills demanded by 21st century learners. Next section (1.1) aims to introduce and explore the concept of TPD more deeply, with a focus on its significance for computer science (CS) education and some of the challenges involved in learning to teach programming. It provides important background and context related to the topic of the thesis.

1.1 Teacher Professional Development (TPD)

The development of flexible, high-quality educational programs is constrained by human resources, time, cost, and geographical location [14]. To overcome these restrictions, governments seek to provide flexible, cost-effective digital alternatives to various educational programs. A shortage of teachers exemplifies the human resource restriction with the required technical and pedagogical competencies in CS. It is critical to prepare teachers in the area of CS, specifically equipping them with the skills to teach programming.

The scientific community agrees that learning programming may be problematic [15, 16], and even more complicated when it is done through online educational environments [17]. Haagensen [18] conducted interviews with in-service teachers who learned to program and found obstacles such as lack of time, teacher discipline and attitude, political issues, varying school priorities, and available programming resources. In addition, teachers are often unable to gain proper training, integrate the learned information into their educational settings, and are frequently required to teach subjects in which they are uncomfortable [19]. Programming can be an intricate topic that demands ongoing effort, a specific way of thinking, and multifaceted knowledge [20]. It is commonly accepted that it takes around a decade to change a novice programmer into an expert [21]. Therefore, flexible programs must consider these challenges and facilitate the best learning environment.

Many professional development organizations worldwide [22–25] provide different types of PD to increase teachers' skills, such as summer programs, workshops, in-school activities, and courses [26]. There are reports of varying degrees of the effect these training programs have on teachers' digital competency [26, 27]. Creating a sustainable computer science education in schools has several obstacles that require a comprehensive approach from the involvement of policymakers, school administrations, higher education, students, and parents [28]. The development of such programs requires the application of effective PD principles.

Darling-Hammond *et al.* [29] outlines seven criteria of successful teacher professional development that contribute to improvements in teacher behavior and enhanced student learning outcomes. These criteria are listed in Table 1.1. Effective professional growth includes most or all these elements.

Another well-known set of criteria for effective professional development is Garet *et al.* [30]'s suggestion of five essential factors: duration (sustained over time), collective participation, disciplinary content focus, promoting active learning, and fostering coherence (the extent to which teachers perceive professional development activities to be a part of a coherent program of teacher learning). Hunzicker [31] suggest a similar list of criteria for effective professional development: supportive (supports teacher motivation and commitment to the learning process), job-embedded (makes it both relevant and authentic), instructional-focus (emphasizes subject area content and pedagogy as well as student learning outcomes), and

Table 1.1: Darling-Hammond *et al.* [29]' seven criteria of successful teacher professional development

No	Criteria
1	Focus on specific content: discipline-specific curriculum development and pedagogy.
2	Incorporation of active learning: interactive activities and other strategies to provide deeply embedded, highly contextualized professional learning.
3	Support for collaboration: space for teachers to share ideas and collaborate in learning.
4	Use of effective practice models: models that include lesson plans, unit plans, examples of student work, peer teacher observations, and video or written instances of teaching.
5	Providing coaching and expert support: sharing expertise on content and evidence-based practice, directly focused on teachers' individual needs.
6	Offering feedback and facilitating reflection: reflect, receive input about and make changes to their practice by facilitating reflection and asking for feedback.
7	Sustained duration: learn, practice, implement, and reflect on new strategies that facilitate changes in their practice.

collaborative (emphasizes both active and interactive learning experiences, often through participation in learning communities)

In their assessment of the literature on successful PD, Gibson and Brooks [32, p. 1065] finds the reoccurring aspects listed below:

> (1) ongoing and intensive; (2) coherent and connected to broader school goals and other professional development opportunities; (3) content and curriculum-focused; (4) based on teachers' needs; (5) delivered in ways that were meaningful and relevant through active learning; (6) collaboration, modeling, and opportunities for practice and feedback; and (7) teacher-controlled and administration-supported.

These are essential, acknowledged, and timeless qualities of professional development, yet creating good PDs continues to be a difficulty [33]. The following section (1.2) will set the context for the study by examining the deficiencies found in current literature regarding professional development for in-service computer science teachers, and by explaining how the suggested framework intends to address some of these gaps.

1.2 Situating the Study Within the Literature and Identifying Gaps

PD for in-service CS teachers is a critical area that requires further exploration and research [29, 33]. The existing literature has highlighted the necessity for a more comprehensive understanding of the challenges and opportunities in this domain [28, 34]. This study seeks to position a proposed teaching framework within the existing literature on PD for in-service CS teachers, emphasizing the need for a structured approach to professional development. Existing literature on PD programs for teachers has outlined criteria for effective PD [29, 32]. However, the creation of successful PD programs remains a significant challenge [35].

Issues may, for example, be related to PDs being unfocused, disconnected from the realities of the classroom [36], and not reflecting adult learning preferences [37]. Another concern has been taking a training model and assuming a solution to a known, common problem (one size fits all) [38]. In his research, Gibson and Brooks [32] identifies as a widespread problem of educators the perception that professional developers did not hear their concerns (including their current knowledge, prior experiences, and unique interests). These challenges underscore the necessity for a more nuanced approach to PD program design and implementation.

While the criteria for effective PD are well-documented, the creation of sustainable CS education in schools faces numerous obstacles that demand a comprehensive approach involving multiple stakeholders, as mentioned in section 1.1. This highlights the need for a holistic perspective that considers the broader educational ecosystem when designing and implementing PD programs for in-service CS teachers. Furthermore, there is a lack of research on designing and evaluating online/flexible PD frameworks specifically targeting in-service CS teachers as a diverse group [39]. This gap presents a significant opportunity for further exploration and development. The thesis aims to address this gap by developing and evaluating a framework tailored to the unique needs of in-service CS teachers, thereby contributing to the advancement of PD in this domain.

1.3 Aim and Research Questions

This research aims to develop a professional development framework that prepares in-service teachers to acquire digital skills and critical competencies related to learning and teaching programming. The following goals guide the research:

G1 Explore the research inside in-service teachers' professional development to build an understanding of existing work and topics that require further examination.

G2 Design a professional development framework for in-service teachers that increases their general interest in learning and teaching programming by (a) making programming relevant to the individual teacher so they can apply it in their subject areas, (b) enhancing collaborative learning and online activities using a communication platform, (c) using project-based learning to create lesson plans that lend themselves to direct use in classroom settings by in-service teachers, and (d) highlighting the advantages and shortcomings of the PD program offered to in-service teachers for programming instruction.

G3 Explore the challenges of mastering programming from an in-service teacher's viewpoint by (a) determining teachers' attitudes towards programming for all and (b) identifying the level of complexity when learning to program from a teacher's perspective.

G4 Explore the challenges and opportunities of teaching programming from an in-service teacher's perspective by: (a) identifying challenges teachers face that may negatively impact their learning and elements that positively promote their future role as programming educators, and (b) exploring the long-term effects of the PD program on employed school educators, focusing on teachers' self-efficacy in instructing how to program.

1.3.1 Main RQ

Research emphasizes the need for high-quality teacher professional development (PD) to support the effective integration of computer science and programming education in schools [28, 29]. However, traditional "one-size-fits-all" models struggle to meet the diverse needs of in-service teachers [38]. Online learning theory (OLT) promotes flexible, learner-centered designs that leverage technology to provide personalized learning experiences [40, 41]. Gibson and Brooks [32]'s research underscores the critical need for a more effective and empathetic approach to professional development in education. It highlights the necessity for professional developers to actively listen to educators, acknowledge their expertise, and tailor development initiatives to align with their unique needs and aspirations. This study aims to address gaps by drawing on OLT to design an online framework evaluated through iterative cycles by asking this main research question:

> *RQ: What are the elements of an online professional development program that prepares in-service teachers to acquire the skills needed to teach programming and incorporate it into their subjects?*

This research employs a combination of exploratory, descriptive, and explanatory approaches to achieve a comprehensive understanding of program design and its impact on learning and motivation. The exploratory aspect of this research allows for the exploration of new facets of program design without predefined hypotheses, fostering the discovery of elements that promote learning and motivation with each iteration. The descriptive approach systematically describes the developed artifact and maps the existing literature to portray the current state of the field. The explanatory approach goes beyond mere description; it seeks to explain the relationships between program design elements and learning outcomes. By understanding why certain design aspects were more or less effective, this analysis helps to uncover the underlying mechanisms driving program success. This comprehensive approach can provide a deeper understanding of program design and its impact on learning and motivation, and may contribute to a more robust and insightful analysis. To answer the main question, we ask several research questions.

1.3.2 RQ-1

Effective teacher PD requires understanding the current state and evolution of research in the field [42]. However, few attempts have been made to systematically map the literature and examine its development over time [43]. Mapping studies are important to link work in a discipline and provide overviews to guide further research [43]. The Community of Inquiry framework emphasizes the role of cognitive, social and teaching presences in online learning communities [41]. Connectivism highlights the value of networks in the digital age [44]. Understanding how research networks and knowledge sharing among stakeholders like researchers, policymakers and teachers have developed can provide insights for strengthening communities of inquiry. By conducting a systematic literature mapping of research from 2010-2020, this study aims to address gaps in understanding the evolution of PD research for in-service CS teachers. Mapping the field through the lenses of communities of inquiry [41] and connectivism [44] can reveal new directions for collaborative knowledge building in this area.

> *RQ-1: How has the research on the professional development of in-service teachers in CS evolved?*

As this question is focused on systematically describing how the research area has changed over time based on existing literature, without exploring new aspects or testing hypotheses, it would be classified as a descriptive research question aimed at portraying the evolution of the field.

1.3.3 RQ-2

Research emphasizes the need for flexible, personalized online PD programs to meet diverse teacher needs [29].

Ni *et al.* [43]' study identifies several areas on TPD that needs further research, e.g., research on models that provide more sustained support for teachers allow deeper development of content and pedagogical knowledge, and examining factors like administrative support and resources is needed to better understand what influences the sustainability of PD efforts. Active learning pedagogies like problem-based and project-based learning have also shown promise for engaging teachers in programming education [30]. RQ2.1-RQ2.4 all focus on specific aspects of designing the online PD framework, and therefore help address the identified gap around developing and evaluating such frameworks.

> *RQ-2: What are the elements of an online flexible teachers' professional development program that promotes active learning and motivates teachers in learning and teaching programming?*

This research question has elements of both exploratory and descriptive research: Exploratory in its open-ended exploration of program elements without preconceptions; descriptive in its objective to systematically describe and identify those elements based on data collected.

By grounding the design in theories of multimodal online education [41] and communities of inquiry [41], this study aims to address gaps in understanding how to develop flexible online courses for a diverse group of adult learners [45]. To answer RQ-2, we have defined several sub research questions.

> *RQ-2.1: How can we design an online flexible learning-trajectory course targeting a diverse group of teachers?*

This research question has descriptive elements as a major focus and describes the case. It also contains exploratory aspects in seeking to understand lessons from this case.

The literature demonstrates [46] that a greater degree of involvement improves attendance and motivation and facilitates the learning process. Students may participate in debates, exercises, brainstorming, simulations, games, or quizzes. Slack as a communication platform was used in the program and we want to investigate how it supports the learning process in a flexible online programming course for in-service teachers.

> *RQ-2.2: How can communication tools like Slack enhance students' learning processes in a flexible learning trajectory environment?*

Study type is mixed methods but can be categorized as primarily descriptive and exploratory in research type based on its stated aims and methodology.

Research has shown that project-based learning (PjBL) can increase student motivation by allowing them to work on authentic projects that apply their learning [47]. However, the effectiveness of PjBL implementation is heavily dependent on teachers' understanding of the strategy. This understanding ultimately impacts students' content knowledge and skill development [48]. PjBL is an educational approach that involves students in examining and solving real-world problems. The in-service teachers were tasked with creating teaching materials for their classrooms as part of a program that implemented a PjBL approach. Specifically, the participants were asked to develop a teaching plan for incorporating programming into their classes. Our investigation focused on identifying the motivational factors that arose from this question.

> *RQ-2.3: What motivational factors are achieved by introducing project-based learning as a bridge between training and practice?*

Study type combines exploratory and descriptive research. It aims to gain new insights into organizing project-based learning for in-service teacher training in programming, testing the "bridge model." Quantitative and qualitative data are collected to characterize experiences with the model.

Effective PD programs are content-focused, incorporate active learning and collaboration, and are of sustained duration [29]. However, few studies have evaluated PD programs from the perspective of in-service CS teachers themselves. Training in-service teachers to become programming educators is complex and intricate. Firstly, the duration of the training program is limited to one academic year during which the teachers are in-service. Secondly, the teachers' backgrounds vary greatly regarding interests, abilities, teaching styles, and experiences. Therefore, it is vital to know the teachers' viewpoints to map their requirements and expectations for professional growth as programming educators.

> *RQ-2.4: In the in-service teachers' experience, what are the strengths and weaknesses of the PD program and what mechanisms can be offered to support this long learning process beyond the duration of this PD?*

Mainly descriptive but also includes an exploratory aspect in qualitatively examining teachers' perspectives. Additionally, it suggests ideas for enhancing similar professional development initiatives based on its findings, thus encompassing elements of normative/prescriptive.

1.3.4 RQ-3

Research emphasizes the important role teachers play in integrating computer science, yet their perspectives remain underexplored [49]. The Community of Inquiry framework highlights the importance of cognitive presence for meaningful online learning through sustained reflection and discourse [41]. Social cognitive theories emphasize how self-efficacy, outcome expectations and affective states like anxiety influence learning [45]. By applying the Community of Inquiry [41] and social cognitive theories [45], this study aims to address gaps by investigating teachers' attitudes, perceived difficulties and how this shape cognitive presence in an online context. Findings can provide insights into scaffolding techniques and resources needed to support teachers' programming skills based on their cognitive needs and perspectives.

We examine teachers' attitudes towards learning to program, what they perceive as complex, how this affects the perception of challenges the students will face, and how their perception of the students' challenges affects the planning of the teaching, by investigating the following research questions:

> *RQ-3: What are teachers' attitudes towards learning to program, and what do they perceive as problematic?*

The study type is mainly focused on exploring and describing teachers' perspectives and experiences through qualitative methods. Its aim is to gain insights and contribute to future research. To answer this research question, several sub-research questions have been investigated (RQ3.1-RQ3.2). Both research questions connect directly to the gap identified around needing more research to understand teacher perspectives and needs better [49] when it comes to programming education and PD programs.

The beginning of this century gave rise to a movement called "CS for all." Communities and organizations worldwide recognize computer literacy's significant role, including familiarity with programming [50]. Programming as part of a teacher's PD has grown in importance in pre-college education. However, debates exist around universal coding education [51]. We sought to gain insights into teachers' perspectives on computational literacy by posing the following research question:

> *RQ-3.1: Should everyone learn to code to be a fully literate participant in our future society?*

The type of study is mainly a descriptive and exploratory qualitative study, with some normative elements in the discussion of implications. The focus is on discovering rather than proving established ideas.

Previous research has identified common difficulties novice programmers face with acquiring fundamental concepts such as iteration, variables, and program design [16]. Additionally, mastering a programming language does not automatically translate into addressing new programming problems. Gaps identified in the literature [16] involves conducting longitudinal studies, creating materials that emphasize practical skills development, and bridging the gap between student and teacher perspectives on challenges. These actions have the potential to advance the literature on supporting novice programmers. We investigated teachers' perspective by asking the following research question:

> *RQ-3.2: What core programming concepts do teachers perceive as problematic? Further, which concepts do they anticipate being challenging for their pupils?*

The study type is primarily a descriptive and exploratory qualitative study that seeks to understand and describe teachers' views and experiences. The goal is to gain insights and inform further research rather than prove or disprove specific theories.

1.3.5 RQ-4

Research emphasizes the importance of teacher self-efficacy for improved student and teacher performance [52]. Measuring teachers' self-efficacy is crucial for developing effective PD frameworks. However, few studies have examined factors that influence self-efficacy in teaching programming from teachers' perspectives [53]. Active learning theories highlight the benefits of problem-based and project-based learning for engaging teachers [41, 44]. The Community of Inquiry framework emphasizes the role of cognitive, social and teaching presences for meaningful online learning [41]. By applying social cognitive [45] and active learning theories [41, 44], as well as the Community of Inquiry framework [41], this study aims to address gaps by investigating elements that promote teaching programming and increase self-efficacy from teachers' experiences. Findings can provide insights into effective online program design features to support the development of teachers' programming self-efficacy.

RQ4 also relates to the gap around understanding teacher perspectives and needs. It aims to gain insights from teacher experiences in the PD program. I look at teachers' perceptions of challenges and opportunities, their self-efficacy in teaching programming, and how this might impact their future role as programming educators. To answer this research question, several sub-research questions have been investigated (RQ4.1-RQ4.2). Both RQ4.1 and RQ4.2 directly relate to the identified gaps around understanding teacher needs/perspectives better and addressing challenges in PD programs. They do so by exploring teacher experiences after and within the PD program respectively. Exploring the following research question can provide insights into the effectiveness of professional development programs in meeting the needs of teachers and addressing their challenges, ultimately contributing to the enhancement of educational practices and teacher support:

> *RQ-4: What elements in a professional development program, in the experience of in-service teachers, promote teaching programming and increase their self-efficacy?*

To explorer this question, qualitative descriptive and exploratory research methods can be employed to help answer the research question. Interviews with teachers can be carried out to descriptively detail their experiences after the training program, exploring impacts on self-efficacy without predetermined hypotheses. This helps identify elements of the program that increased self-efficacy. Similarly, analyzes of teacher reflection notes through a qualitative approach can be applied to descriptively present themes around challenges and opportunities discovered in teaching programming, taking an exploratory approach without fixed expectations. This aids in understanding elements that promote teaching programming from the teacher perspective. This helps answer the research question by understanding the teacher experience of elements that increase self-efficacy and promote teaching programming. To answer RQ-4, we have defined several sub-research questions.

Teacher self-efficacy impacts motivation and classroom practices [52]. We interviewed in-service teachers who had completed the PD program. Specifically, we explored their sense of ease and self-efficacy in teaching programming by asking the following question:

> *RQ-4.1: Can you share your experience in teaching programming on the back of the PD program?*

The study type encompasses a mixed methods approach, incorporating elements of descriptive and exploratory research. The descriptive aspect focuses on detailing the in-service teacher training program and presenting the outcomes of interviews regarding teachers' attitudes, self-efficacy, and perceptions. The exploratory component is evident in the research question, which seeks to understand teachers' long-term perceptions of the training's impact on their self-efficacy without testing a specific hypothesis. This multifaceted approach can provide a comprehensive understanding of the in-service teacher training program and its implications for teacher development.

The computer education literature frequently states that learning and teaching programming are difficult [49, 54]. To get insights into the experience of in-service teachers during their training to teach programming to their pupils, we investigated the following research question:

> *RQ-4.2: Which are, in the experience of in-service teachers, the elements that promote positive learning and the challenges they face that might negatively impact their future role as programming educators?*

The study has a primarily descriptive and exploratory aim/approach in seeking to understand this context based on teacher perspectives, though it also tentatively explores some implications in a less prescriptive manner.

In this section, we established the purpose and scope of the research by stating its aim to develop a PD framework and outlining the specific goals and research questions that will direct the study. It frames the objectives that will be investigated. Next section discusses the key contributions the research has made to advancing knowledge and understanding of professional development for in-service CS teachers. It outlines the original contributions to both research and practice based on the findings.

1.4 Research Contribution

The primary contribution of this research was the development and evaluation of an online professional development framework for in-service computer science teachers. Through three iterative design cycles involving over 450 teacher participants, the framework was refined to meet the specific needs and expectations of in-service educators. It provided teachers with flexible learning opportunities to gain both the programming skills and pedagogical knowledge required to teach computer science concepts. The evaluation of the framework yielded valuable insights into effective models for professional development in this domain. It offered guidelines and lessons that can help inform the design of future online teacher training programs. The study contributed to the field of pre-college CS education by:

- C1 identifying the main directions along which research is evolving and the questions that require further attention. This contributes to the research field by providing an overview of where more work is needed, such as developing standardized abstract structures. It helps guide future research agendas.

C2 suggesting and evaluating a professional development framework for a diverse group of in-service teachers that provides flexibility in meeting the learning objectives. This contributes a model for practitioners (teacher educators, schools) to reference when designing their own online/blended teacher PD programs. It also provides lessons for researchers evaluating such programs.

C3 emphasizing the capacity to master programming through sufficient teacher training and learning time, while recognizing persistent challenges, and demonstrating a positive shift in in-service teachers' attitudes toward incorporating programming and coding education in school curricula, despite recognizing potential student difficulties. This contributes to educators, policymakers, and researchers interested in the integration of programming and coding education into school curricula.

C4 providing a teacher-centered "mindset" (Understanding the challenges and opportunities of integrating programming into the classroom from the teacher's perspective) on instructing programming in schools, focusing on teaching and integrating programming in various subjects. This contributes a new perspective for both researchers and practitioners. Researchers now have a lens for how to approach their work. Practitioners can apply this mindset to make training/resources more relevant and useful for teachers.

1.5 Purpose and Structure of the Thesis

The purpose of this thesis is to present a comprehensive research project that serves as the foundation for this doctoral work. The primary objective is to introduce the findings and proposed professional development framework, which are intended to be valuable for several key stakeholders, including computer science education researchers, policymakers, university faculty, and K-12 computer science teachers within the field of computer science education.

For computer science education researchers, this thesis offers insights into effective models for the professional development of in-service teachers. By providing a detailed analysis of these models, the research aims to contribute to the advancement of understanding in this area, thereby facilitating the development of more effective training programs for educators.

At the ministry and school district levels, policymakers are constantly making decisions about support for in-service training programs. This thesis aims to provide valuable data and recommendations that can inform these decisions, ultimately contributing to the improvement of teacher preparation and training initiatives within the field of computer science education.

University faculty members involved in designing and delivering in-service and continuing education courses focused on pedagogical content knowledge for teaching K-12 computer science will find this research beneficial. The insights and proposed professional development framework can serve as valuable resources for enhancing the quality of educational programs, thereby improving the preparation of future computer science educators.

Current and prospective K-12 computer science teachers seeking to enhance their subject matter and pedagogical skills through flexible, online learning opportunities will benefit from the findings and proposed professional development framework outlined in this thesis. The research aims to provide practical guidance and resources that can support the professional development of educators within the K-12 computer science domain.

1.5.1 Structure of the Thesis

Part I: Synopsis

- Chapter 1 describes this thesis's background, motivation, and aims and outlines the approach, results, and contributions.
- Chapter 2 presents the theories and definitions that underpin this research.
- Chapter 3 presents the case and the research method.
- Chapter 4 summarizes the results.
- Chapter 5 discusses the research results concerning the research questions, contributions, and implications.
- Chapter 6 includes final remarks and suggestions for future work.

Part II: Papers

- P1 Design of a programming course for teachers supporting flexible learning trajectories.
- P2 In-service teacher training and self-efficacy.
- P3 Utilizing slack as a communication platform in a flexible learning trajectory course: supporting the learning process.
- P4 Teaching programming in secondary schools: Stepping and stumbling stones.
- P5 In-service teachers' Attitude Towards Programming for All.
- P6 Project-based learning and training of in-service teachers in programming: projects as a bridge between training and practice.
- P7 PD for In-service teachers of Programming: Evaluation of a University-Level Program.
- P8 Learning to Program: an In-service teachers' Perspective.
- P9 CS in Schools: A Literature Mapping of PD for In-service teachers.

1.6 A Note About the Use of Terms in This Thesis

The term "instructor" refers to the professor(s) of the training program being discussed in this thesis. When referring to participants of the program, the terms "teacher," "educator," "in-service teacher," "pre-college teacher," "participants," and "students" are used. Further, we use the term "pupils" when discussing pre-college students.

Chapter 2

Theoretical Grounding

The current research is positioned at the intersection between computer science education and online learning. By combining these research fields, this work aims to contribute to a more holistic understanding of how a diverse group of in-service teachers learns computer science in online environments. This research adopts online learning itself as the overarching theoretical framework. Given the focus on developing and evaluating an online professional development program for teachers, conceptualizing teacher learning through the lens of online learning provides the most direct relevance. Several existing online learning models are incorporated, such as the Community of Inquiry frameworks, Multimodal- , and Anderson's online learning models. This allows for a coherent analysis of how online features like interactions, presences, technologies and other online design elements influence teacher learning outcomes in a digital environment.

The study draws from established learning theories to inform its analysis while also considering the unique challenges and opportunities of online learning. Ultimately, the goal is to identify effective strategies for teaching computer science in digital contexts and to shed light on how technology can be leveraged to enhance student learning outcomes. By engaging with these complex and multifaceted topics, this thesis hopes to make a meaningful contribution to the ongoing conversation around computer science education and online learning.

Section 2.1 offers a comprehensive definition of online learning derived from existing literature. Moving on to Section 2.2, the focus is on presenting several significant theoretical frameworks and models relevant to online education. The discussion delves into the key concepts and components of each framework/model, with a specific emphasis on how they conceptualize aspects like knowledge domains, interactions, presences, and elements necessary for a unified theory of online education. Section 2.3 examines literature on the inherent difficulties in learning to program, highlighting the cognitive load, complexity across multiple knowledge domains, and progression of challenges from basic to applied levels. It provides context on the difficulties teachers face in learning programming. Section 2.4 discusses the pedagogy of teaching programming. It outlines that programming pedagogy involves mastering a hierarchy of skills simultaneously. It also notes that active learning pedagogies like project-based and problem-based learning have shown promise for engaging teachers in programming education. Section 2.5 provides an overview of the various online learning theories, models, and frameworks that inform the design of the Online Professional Development Framework for in-service teachers. Additionally, it offers insights into programming pedagogy derived from the collection of papers presented, highlighting the significance of integrating pedagogical factors into programming instruction and developing teachers' competencies to effectively teach programming.

2.1 Online Learning

Online learning, online education, e-learning, distance learning are all terms used to express the nontraditional method of teaching and learning. Singh and Thurman [55] conducted a systematic literature review to collect definitions of online learning from peer-reviewed journal articles published between 1988-2018. They collected 46 unique definitions of online learning from 37 different resources/references. While there is no single agreed upon definition, a definition that captures the common elements found across the literature could be: Online learning is education facilitated and delivered using internet and digital technologies to provide asynchronous and/or synchronous access to learning, which may involve interaction between learners and instructors as well as resources [55].

The integration of technology is a fundamental component within the realm of online learning, encompassing the use of internet and digital tools to facilitate and deliver education. Interactivity is also a critical focus in many definitions, emphasizing the significance of engagement between students and instructors, as well as peer-to-peer interaction through online resources. Additionally, definitions frequently differentiate between synchronous and asynchronous learning, specifying whether online learning includes real-time engagement, such as live online classes, or non-real time interaction, where students access materials at their own pace [55].

Accessibility is a key advantage of online learning, as it removes physical location and time constraints, allowing learners to participate from anywhere with an internet connection. Moreover, the flexibility of online learning modalities, both in terms of location and pace of learning, is underscored in certain definitions, providing learners with greater control over their schedules [56]. The use of online technologies also opens up new avenues for delivering educational content, including multimedia and interactive activities. In addition, the significance of fostering an online learning community and promoting diverse interactions to mitigate potential isolation is increasingly recognized in later definitions. Early definitions also acknowledge the increased workload for instructors in transitioning content online, emphasizing their pivotal role in facilitating online interactions [55].

Providing online education to our 21st-century learners has been essential to the growth of higher education. But building an engaging online learning environment that improves understanding of course material can be demanding for a student group with varying demographics due to different circumstances. The shift towards online professional development (PD) has emerged as a viable solution for educators facing geographical or time-related barriers to attending in-person workshops. This alternative offers increased flexibility and accessibility, enabling teachers to engage in meaningful learning experiences. However, the effectiveness of online PD depends on the quality of instructional design. Meaningful contexts, collaborative activities, and higher-order thinking must be thoughtfully integrated to ensure a rich and engaging learning environment. Establishing social presence, characterized by trust, belonging, and interaction, is paramount. Additionally, cognitive presence, involving sustained communication and reflection, is equally important and should be seamlessly integrated into the design [55].

Online learning theory has been a subject of in-depth analysis, with ontological assumptions that knowledge exists objectively and can be transferred through online means, and that learners and teachers are separate entities connected through technology. Epistemological assumptions include the transmission of knowledge from experts to learners and the individual acquisition and application of knowledge. The analysis of online learning theory has highlighted the importance of incorporating social constructivist views and emphasizing the community and interaction in online learning. These developments have led to a more comprehensive understanding of online learning and its implications for education.

2.2 General Learning Models and Frameworks for Online Education

This section gives an overview of the TPACK framework, CoI framework, Anderson's model and Picciano's multimodal model. It discusses how these theories conceptualize key aspects of online learning like the knowledge domains, presences, interactions, and components required for a unified theory of online education. These frameworks inform the theoretical grounding and approach taken in the study.

2.2.1 The TPACK Framework

Teachers must consider how their knowledge domains cross to effectively teach and engage students with technology. Technological and Pedagogical Content Knowledge (TPACK) examines what they know, how they teach, and how technology might improve students' learning. TPACK is an application of Shulman [57]'s Pedagogical Content Knowledge (PCK) work. By merging technology with the notion of PCK, Mishra and Koehler [58] advanced the concept of TPACK as a "lens" for analyzing the development of teachers' capabilities. In Figure 2.1, TPACK is illustrated, demonstrating that technical knowledge (TK) must integrate with the content (CK) and pedagogy (PK) that are specific to each topic area.

Content knowledge (CK) are topics in the curriculum that provide understanding and skills in the subject matter. This may include, for example, physics, maths, social studies, and other courses that impart knowledge and abilities. The experts' understanding of the art and science of teaching, from learning theories to instructional design, is known as pedagogical knowledge (PK). Project-based learning and instructional tactics like think-pair-share [59] are examples of teaching and assessment practices included in PK. Understanding these concepts enables the teacher to create practical learning experiences for every learner. The pedagogical and content domains are intersected by pedagogical content knowledge (PCK). It is the best way to get students interested in learning new ideas and abilities. This expertise covers strategies for adapting content to suit various learning preferences and structuring it for greater comprehension. Knowledge of the tools, including how to choose, apply, and incorporate technology into the curriculum, is called technological knowledge (TK). Students gain fresh perspectives and possibilities to study technological topic knowledge when technology is included in pedagogical content knowledge (TCK). TCK uses technology in a subject area to promote profound and long-lasting learning.

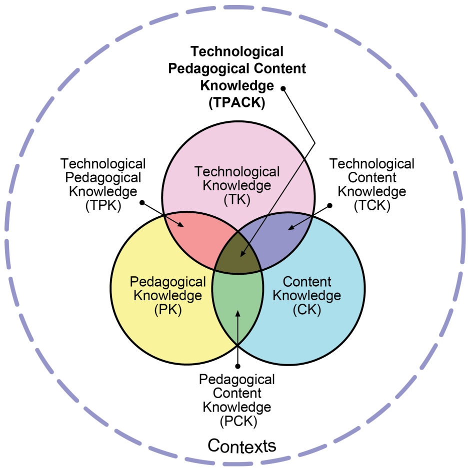

Figure 2.1: TPACK Framework. Reproduced by permission of the publisher, © 2012 by tpack.org [1].

The TPACK framework is one of the essential theories of online education due to its multidimensional perspective on the overlapping abilities necessary to educate online. However, the TPACK paradigm focuses on the cognitive domain and excludes attitudes, beliefs, motivation, and other emotional domain components [60].

The TPACK framework is a valuable tool for educators seeking to integrate technology into their teaching practices. This framework emphasizes the importance of understanding the interaction between technology, pedagogy, and content in order to effectively teach students. It is based on the pragmatic view that knowledge is gained through practical application and is consistent with theories that emphasize the practical integration of knowledge domains. Additionally, the TPACK framework is compatible with theories that focus on teacher knowledge, making it a versatile and valuable resource for educators looking to enhance their teaching practices through the integration of technology.

2.2.2 Community of Inquiry Framework (CoI) As a Social Constructivist Learning Model [61]

A fundamental shortcoming in the online educational community is human connection and the social realm, which needs to be purposely addressed and planned for. Social constructivism places emphasis on culture and setting as essential elements for understanding social development and generating knowledge based on this understanding [62, 63]. According to social constructivism, [61], all cognitive processes, including learning, depend on interactions with others. In other words, when someone else helps us, our cognitive processes change. The Community of Inquiry model considers learning as being facilitated by three interacting entities: 1) cognitive presence (the degree to which learners may generate and validate meaning in a virtual community of inquiry by exchanging and connecting ideas and addressing critical thinking across the board), 2) social presence (the capacity of participants in an online class to project themselves socially and emotionally and, as a result, regard one another as "real."), and 3) teaching presence (designing and organizing online courses, facilitating learning, and providing direct teaching inside them). Figure 2.2 illustrates the connection between these entities.

Grounded in social constructivism, this framework recognizes that knowledge is co-created through interaction and collaboration within a community. It is based on the ontological assumption that learning is a result of discourse and interaction, and the epistemological assumption that individuals personalize information through reflection and discourse. By recognizing the importance of interaction and collaboration in the learning process, the Community of Inquiry Framework provides a valuable perspective for educators and learners alike.

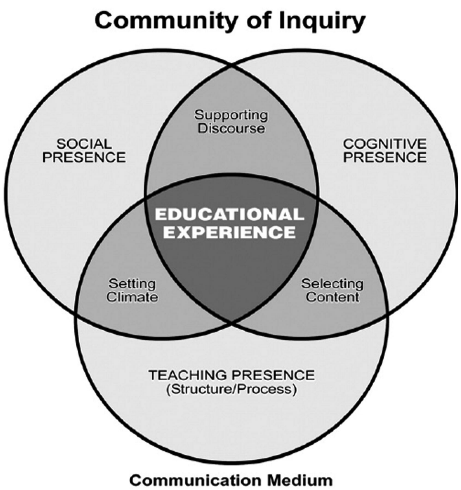

Figure 2.2: The Community of Inquiry model, Shea *et al.* [64].

Cognitive Presence

To better comprehend the topic, students must generate knowledge and understanding individually and collaboratively and use sustained communication. Inquiry or reflective thinking is crucial to teachers' and students' learning [65]. Reflective cognition is a cycle that starts with a problem, goes through five stages (claims, problems, hypotheses, reasoning, and testing), and ends with a solution [66]. This idea served as the inspiration for the Practical Inquiry Model [61], which was created to explain the cognitive presence in the CoI framework [67]. It promotes greater awareness of critical thinking and discourse and guides students in constructing personal meaning and sharing understanding. Figure 2.3 shows the model and its four phases:

1. Triggering event: This is frequently a well-planned activity in a classroom setting, often a problem or an issue that the students can relate to from prior experiences or studies that the teacher has prepared.
2. Exploration: The students must first comprehend the issue before seeking relevant data and potential solutions, preferably through group brainstorming or independent literature searches.
3. Integration: The integration phase will emphasize reflexive actions in the person's knowledge-building through conversation with others.
4. Application/Resolution: Clarification occurs during the resolution process. The resolution is rarely totally attained, but this stage frequently "triggers" new issues and themes to be investigated.

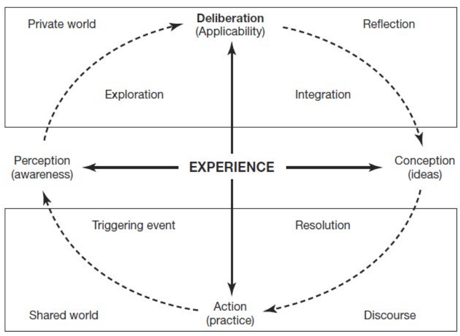

Figure 2.3: Practical inquiry model, Olpak *et al.* [68].

Social Presence

The term "social presence" is linked to the capacity of various media to transmit signals (verbal and visual), which is an essential part of in-person communication [69]. Social presence is defined under the CoI paradigm as the participant's capacity to establish oneself socially and emotionally in an online class and to view other participants as "real" in that class [70]. Social presence has great significance for students learning. It has much to do with what they're using to communicate. Garrison and his colleagues [71] have shown that you can establish a high social presence using text. But it could be easier to do that using video or audio or things like that.

Within the CoI paradigm, social presence is conceived as three behaviors: emotional expression, group cohesiveness, and open communication. The affective or emotional expression uses personal terminology for emotions, sentiments, ideas, and values to convey one's presence. Group cohesion is established and maintained via interpersonal communication. Open communication is acknowledging, complementing, and reacting to others in a manner that encourages dialogue and critical thought. Thus, these three behaviors are seen as mutually reinforcing to produce an atmosphere that facilitates the social production of knowledge [61].

Teaching Presence

Teaching presence in online courses deals with design and organization, facilitation of discourse, and direct instruction [67]. Teaching presence manifests itself via the teacher's capacity to recognize relevant information, develop learning activities that foster reflection and conversation, and evaluate students' learning outcomes. Facilitation for learning focuses mainly on online conversation facilitation, where it is essential to be helpful and present. However, it is crucial to foster joint activities and individual student development. Direct instruction comprises any lesson content in online courses and directions in students' responses to assignments [61]. The community of inquiry idea presented here is embedded in Anderson [40] 's Online Learning Model introduced in the next session.

2.2.3 Anderson's Online Learning Model

According to Anderson [40], many theorists see online learning as a subcategory of 'general learning' and 'distance learning' -a perspective that complicates the construction of a standard online education theory [41]. Anderson [40] nevertheless discusses the possibility of developing an inclusive theory of online learning based on the CoI model. He builds on Bransford *et al.* [72]'s work which defines effective learning environments characterized by the intersection of four overlapping lenses: knowledge-, learner-, community-, and assessment-centeredness. Considering this and the fact that the Internet has evolved and supports many different media types, Anderson [40] proposes a new model for online learning depicted in Figure 2.4. In the model, we see the actors (students and teachers), the content, and the interactions between them. Students can access content or collaborate with teachers to get assistance through synchronous or asynchronous activities and structured learning tools.

A variant of Anderson [40] 's approach, known as 'the Multimodal Model for Online Education', proposed by Picciano [41], is a recent effort at integrating online learning. This paradigm offered a more practical foundation for online training and will be explained in the next section.

The ontological assumptions of the model posit that knowledge exists in a distributed manner across a network of actors, content, and interactions/tools. It assumes an objective view of knowledge that can be accessed and shared through the online network. In terms of epistemological assumptions, the model suggests that knowledge is constructed through active engagement and participation within the network, and takes a pragmatic, applied view that knowledge is demonstrated and applied in practice. The model is consistent with social

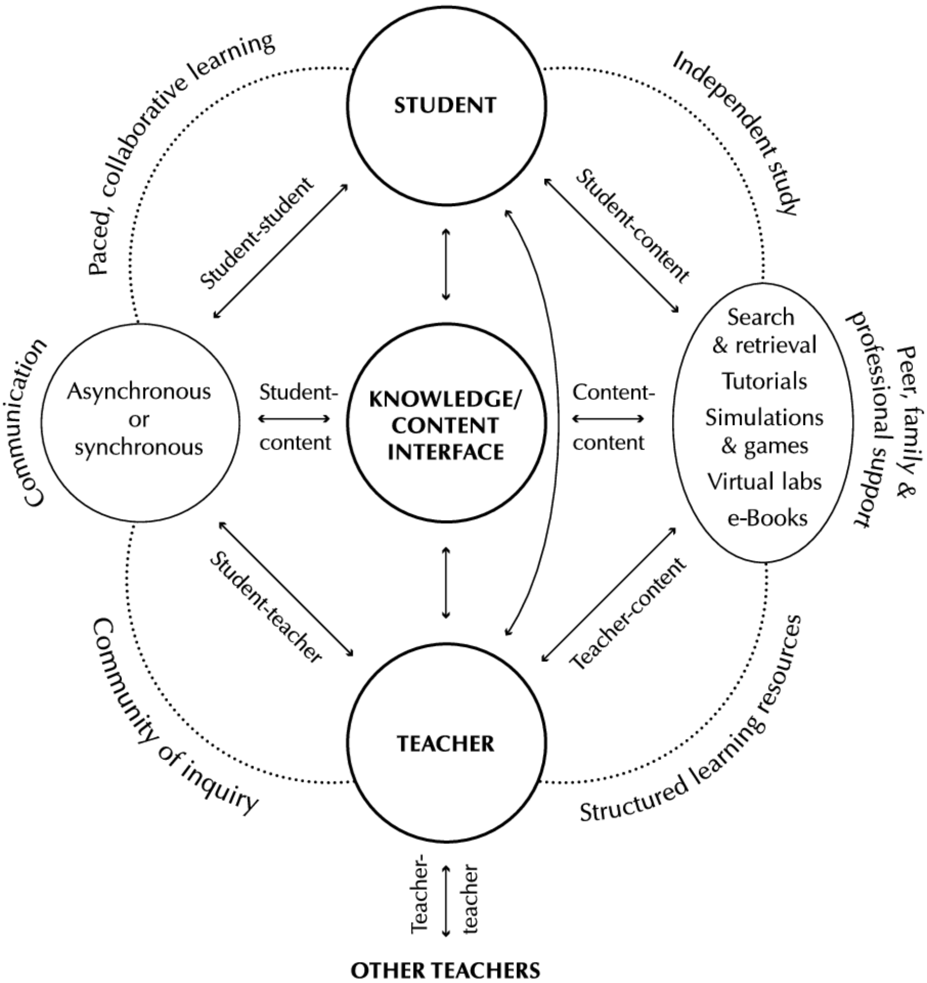

Figure 2.4: Anderson [40]'s Online Learning Model. The theory and practice of online learning.

.

constructivist theories such as CoI that view learning as socially constructed through interaction and discourse. Additionally, it is compatible with theories emphasizing blended/multimodal learning environments, such as the Multimodal Model.

2.2.4 Multimodal Model for Online Education

According to Anderson [40], a theory or model based on online learning should include all other modes except for "*rich face-to-face interaction in formal classrooms*" [40, p.67]. However, a common theory for online teaching becomes problematic if it does not support personal, face-to-face activity assuming that online teaching is a subcategory of education. Picciano [41], in his study, looked for a blended model for online teaching by approaching a face-to-face education perspective. Based on the 'Blending with Pedagogical Purpose' model from Bosch [73] and the incorporation of several elements from various theories and models, Picciano [41] suggested a new model for online education called the "Multimodal Model for Online Education" (see Figure 2.5).

The multimodal model for online education is a contemporary approach that integrates elements of social constructivism with a blend of online and face-to-face learning. This model acknowledges that learning takes place through a variety of modes, including content delivery, reflection, and collaboration. It also emphasizes the idea that knowledge is constructed through different modalities within a community. This approach is consistent with social constructivism and other theories that highlight the importance of blended and multimodal learning experiences. By incorporating a range of learning methods and technologies, the multimodal model aims to create a dynamic and interactive educational environment that caters to diverse learning styles and preferences.

This model considers a course or an academic program as a learning community where interaction is one of its fundamental characteristics. Self-study and autonomous learning are included in the paradigm, which Anderson [40] highlighted as incompatible with community-based models. This model seeks to address Anderson [40]'s concerns on the components necessary for a coherent and integrated theory for the online education model. Figure 2.6 shows a teacher-led, fully online course. A brief explanation of the components applicable to online education follows:

- Content - a platform system provides the course content.
- Dialectic or questioning - examining what the students know is important through dialectics and questions.
- Reflection - can be a personal exercise and allows sharing of one's insights. Enhancing and extending reflection are pedagogical practices that require students to think about their learning and share their thoughts with teachers and classmates.
- Collaborative learning - an approach for solving problems in groups. Email, wikis, and other forms of digital communication are examples of technologies that promote online cooperation.
- Evaluation - assessment of student learning is an essential component of the paradigm, and techniques such as tests, assignments, and portfolios may be conveniently administered online.

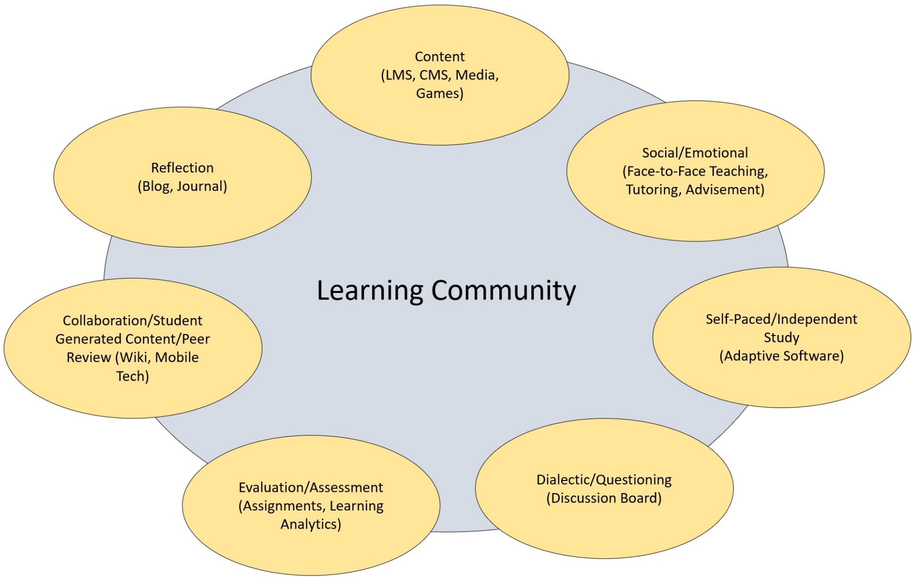

Figure 2.5: Multimodal Model for Online Education, adapted from Picciano [41]
.

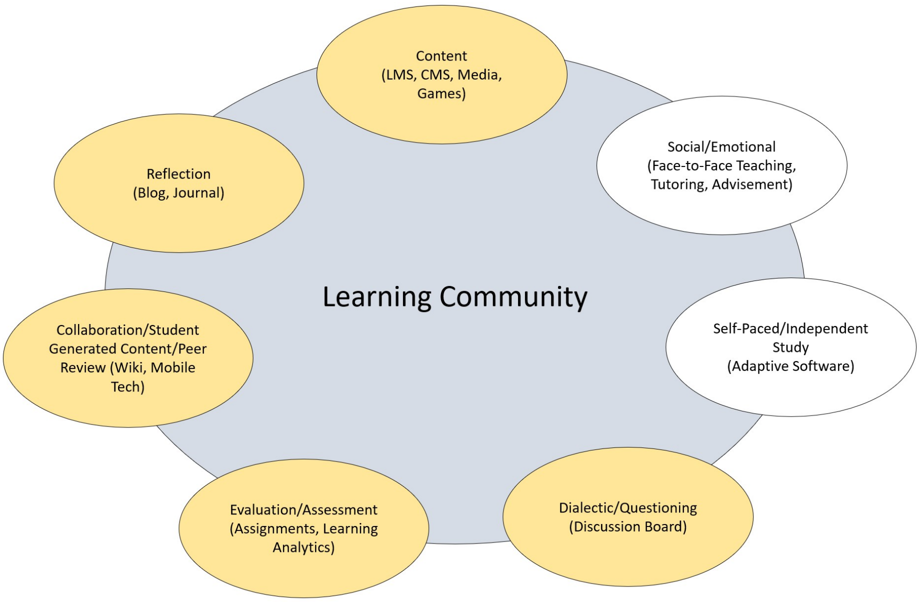

Figure 2.6: Teacher-Led Fully Online Course, adapted from Picciano [41]
.

2.3 Understanding the Challenges of Teaching and Learning Programming

The acquisition of programming skills, particularly in text-based programming languages, is often portrayed as complex and demanding [15, 16]. High dropout rates in programming courses reflect the difficulty involved, and mastery of a programming language's constructs does not ensure the ability to solve new programming problems. Programming encompasses various cognitive activities and mental representations, including program design, understanding, modification, debugging, and documentation. Even at the level of computer literacy, it requires conceptual knowledge and the ability to structure basic operations into schemas and plans, as well as the development of flexible strategies to derive benefits from the programming environment and methods [74].

This complexity is further highlighted by the elements involved in programming, as outlined by Juárez-Ramırez *et al.* [75]: (a) Problem domain; (b) Programming language domain; (c) Programming paradigm domain; and (d) Solution orchestration. Consequently, programming comprises a hierarchy of skills, with programmers utilizing many of these skills simultaneously. It is not merely a single skill but involves multiple distinct processes.

Previous research [76] has shown that novice students may encounter difficulties in learning programming, particularly in understanding programming concepts and program design. Novice developers face a wide array of challenges and deficiencies, with mastering syntax being one of the initial hurdles. Writing syntactically correct code is a recurring challenge for novice programmers.

Learning programming involves not only acquiring knowledge and skills but also developing a sense of mastery and self-efficacy in learning and teaching programming. The success of a professional development program for teachers can be measured by the change in participants' confidence in their programming ability and their capacity to teach programming and computation to others. Teachers' self-efficacy is crucial in teaching, as low self-efficacy can have a detrimental impact on teaching effectiveness and performance.

TPACK encompasses knowledge about the intricate relationships between technology, pedagogy, and content, enabling teachers to develop appropriate and context-specific teaching approaches. In the context of programming, TPACK includes dimensions such as content knowledge (CK), pedagogical knowledge (PK), and pedagogical content knowledge (PCK). For programming, CK represents knowledge of concepts and practices in programming, while PK refers to general pedagogical knowledge. PCK addresses fundamental questions related to teaching programming, such as why to teach programming, what should be taught, the learning difficulties, and how to teach programming.

Teachers may find it challenging to separate the process of learning programming from the task of teaching it. Addressing pedagogical challenges related to learning programming concepts can enhance the relevance of the course, which is crucial in professional development. It is essential to identify topics that are more challenging to grasp to provide high-quality in-service teachers' professional development.

Teachers often learn to program not with the goal of becoming programmers themselves, but with the intention of teaching programming to enhance students' understanding in their field of study. This additional responsibility must be managed alongside regular teaching, adding to the pressure of time. This study seeks to understand the challenges highlighted by teachers, how these challenges affect their perceptions of what their students will find challenging, and whether the teachers' challenges and beliefs about their students impact their teaching. To our knowledge, there is limited research on teacher training programs for in-service teachers where both learning and teaching programming are contextualized.

In the exploration of some of the educational theories in chapter 2, it becomes evident that the theories discussed here are harmonious with other theories presented throughout the chapter. The TPACK framework is relevant for online education due to its emphasis on teachers' technological, pedagogical, and content knowledge domains. Moreover, the self-efficacy theory is a crucial theory applied in the OPDF, pertaining to the measurement of teacher confidence. The emphasis on active, collaborative approaches in section 2.4 aligns with the active learning theory, discussed elsewhere. Moreover, the emphasis on developing skills and competencies through social interactions aligns with the social constructivism perspective. Furthermore, the integration of various pedagogical factors as discussed in section 2.4 is consistent with the multimodal model's emphasis on blended learning. This coherence among the theories discussed throughout the chapter underscores the interconnectedness and applicability of these theories in the realm of education.

2.4 Pedagogy of Teaching Programming

The collection of papers presented here offers a comprehensive overview of the intersection between programming pedagogy and practical teaching experiences. While each paper may not explicitly center its discussion on pedagogical theory, they collectively provide insights into the design decisions and considerations that underpin effective programming instruction. The emphasis on tailoring learning experiences to different learners and teaching levels/subjects reflects a nuanced understanding of pedagogical principles and the diverse needs of students and educators (papers P1 [5], P4 [77]). Furthermore, the papers touch upon various aspects of programming pedagogy, including teachers' self-efficacy, instructional strategies, assessment, and lesson planning (papers P2-P5 [77–80]). By delving into teachers' perceptions of their capabilities and competencies in teaching programming, these papers shed light on the challenges and opportunities inherent in pedagogical approaches to programming instruction.

The incorporation of active, collaborative, constructivist, and feedback-based learning approaches in course and tool design demonstrates a commitment to developing pedagogical programming skills in teachers. The evidence presented in the evaluation of these techniques further reinforces the importance of skill development for effective programming instruction (papers P1 [5], P3 [79]). Additionally, the papers analyze pedagogical dimensions of learning to teach programming based on teacher experiences, encompassing considerations of motivation, assessment, integration across subjects, and adapting instruction based on student needs (papers P2 [78], P4 [77], and P5 [80]). This multifaceted exploration provides insights into the complexities of programming pedagogy and offers practices for educators to consider.

While some papers may not extensively focus on specific pedagogical theories or methods, the conceptual discussion of programming pedagogy as a crucial aim and element of teacher training models is noteworthy (paper P4 [77]). The emphasis on developing teachers' pedagogical competencies to effectively teach programming aligns with the broader goal of enhancing educational practices. Moreover, the inclusion of pedagogical considerations in evaluating professional development programs and analyzing teachers' learning experiences and perspectives further underscores the significance of integrating pedagogical factors into programming instruction (paper P2 [78], P4 [77]). The discussions around linking content to contexts, flexible/personalized approaches, and the evolving role of the teacher highlight the evolving landscape of programming pedagogy and the need for responsive and adaptable teaching practices (paper P1 [5], P5 [80]).

2.5 Online Professional Development Framework (OPDF) for In-Service Teachers

Knowles *et al.* [45] differentiated adult learning (andragogy) from child learning (pedagogy). Adults learn differently than children, whether they are attempting to further their careers or fulfill their sense of curiosity. Adults' social settings and life experiences should inform the design of courses for them. In higher education, adult students often use online technologies in continuing education programs. Therefore, Knowles *et al.* [45]' theory is vital in designing and implementing programming courses for in-service teachers [41].

First, I provide an overview of various online learning theories, models and frameworks that inform the design of OPDF for in-service teachers (Section 2.5.1). These include the Multimodal Model for Online Education, Community of Inquiry framework, TPACK framework, learner-centered design, social cognitive theory of self-efficacy, active learning theory and others. Next, I discuss how the theoretical perspectives on online learning inform the formulation of the research question(s) and methodology used to design and evaluate the OPDF that is the focus of this study (Section Section 2.5.2). It establishes the theoretical grounding for the research approach.

2.5.1 Theories, Models and Frameworks

Table 2.1 provides an overview of the various online learning theories, models and frameworks that are applied in OPDF developed for in-service teachers in this study. It first lists the Multimodal Model for Online Education. This model is said to identify elements needed for an integrated online education theory. It considers a course as a learning community centered around interaction. Papers 3 and 6 in the study relate to how this model's elements like social, cognitive and teaching presences were supported through the use of Slack to enhance online learning outcomes. Next, the Community of Inquiry (CoI) framework is discussed. CoI proposes that learning is facilitated through cognitive, social and teaching presences. Papers 1, 3, 4, 6, 7, 8 and 9 show some conceptual relevance to CoI, though it is not always explicitly evaluated. The papers provide evidence for how the three presences interacted to enhance online learning experiences.

The TPACK framework examines teachers' technological, pedagogical and content knowledge domains. It is relevant as the online environment incorporates technology, content and pedagogy. Learner-centered design constructs learning based on learners' needs and interests. Paper 1 applied this approach. Bandura's social cognitive theory of self-efficacy, connected to outcomes, is used to measure teacher self-efficacy dimensions in papers 2, 4, 5, 6, 7 and 8. Active learning theory and concepts like cooperative/collaborative learning, flexible trajectories and social learning theory informed aspects of Paper 3's course design. Other frameworks discussed include programming pedagogy challenges, teacher professional development characteristics, computational thinking concepts, motivational theories, project-based learning approaches, connectivism, constructivism, communities of practice, and reflective practice.

Table 2.1: Overview of online learning model applied for the professional development framework of in-service teachers

Theories, models, and frameworks	It's relevancy to OPDF for in-service teachers.
Multimodal Model for Online Education	Is a component model that identifies "elements that might be needed for an integrated or unified theory or model for online education. It considers a course or an academic program as a learning community where interaction is one of its fundamental characteristics. The study design and findings of **paper 3** do relate to how Slack supported the social, cognitive, and teaching presences proposed by the model to enhance online learning outcomes. The tool helped achieve an effective multimodal learning experience.

Continuation of Table 2.1	
Theories, models, and frameworks	It's relevancy to OPDF for in-service teachers.
CoI	Learning is facilitated by three interacting entities: cognitive presence, teaching presence and social presence. **Paper 1** relates to some aspects of the course design rationale and goals to constructs within the Community of Inquiry framework, even if not explicitly evaluating it through that theoretical lens. CoI provides some implicit conceptual relevance. **Paper 3** does not directly reference CoI, but the paper's approach and conclusions do align with and provide evidence for how the three core presences interacted through the use of Slack to enhance the online learning experience. **Paper 4** does not directly apply the CoI framework, the paper's analysis of teacher experiences does have some relevance for understanding how to foster a community of inquiry, especially among in-service educators. There are conceptual connections between the approach studied in **paper 6** and CoI principles, even if CoI is not directly referenced in the paper. The paper is relevant for those interested in applying the CoI lens to online project-based teacher education. While CoI is not explicitly referenced in **paper 7**, some of the key findings around the value teachers placed on collaboration, community, and ongoing interaction/presence mirror aspects of the CoI framework for online learning environments. The paper implicitly points to the relevance of social, cognitive, and teaching presence over the long term for effective PD. **Paper 8** touches upon important elements like online collaboration, reflection, course design and facilitation that are key components of the Community of Inquiry approach to online learning. The findings could potentially inform strengthening the three presences defined in the CoI model. **Paper 9**'s analysis does engage to some extent with concepts proposed in the Community of Inquiry framework for online education.
TPACK	**Paper 8** discuss the TPACK framework which examines what teachers know, how they teach, and how technology might improve students' learning; TPACK paradigm focuses on the cognitive domain and excludes attitudes, beliefs, motivation, and other emotional domain components. Online learning environment where technology, content and pedagogy are important elements.

Continuation of Table 2.1	
Theories, models, and frameworks	It's relevancy to OPDF for in-service teachers.
Learner-centered design	Constructing learning opportunities based on who the learner is and wants to be, rather than what experts want them to be. **Paper 1** describes applying a learner-centered approach to allow students to identify and follow learning paths that fit their competencies, interests and needs. Literature citing benefits of combining web-based learning with a learner-centered structure.
Bandura's social cognitive theory of self-efficacy	In the context of teaching, teacher self-efficacy refers to a teacher's belief in their own ability to teach a subject effectively and help students achieve learning outcomes. **Papers 2, 4, 5, 6, 7, 8** discuss and cites research on self-efficacy which is connected to increased student and teacher outcomes, as well as teacher well-being. It discusses how previous studies have found many teachers lack confidence in teaching programming (i.e. have low programming teaching self-efficacy). The interview guide and analysis are grounded in self-efficacy theory, drawing on constructs like the validated Teachers' Sense of Efficacy Scale to measure different dimensions of self-efficacy.
Active learning theory	**Paper 3** discusses active learning as an alternative to traditional lectures where students are more actively engaged through discussions, exercises, brainstorming etc. It cites research showing active learning increases engagement and motivation.
Cooperative and collaborative learning	**Paper 3** discusses the differences between cooperative (working together towards individual goals) and collaborative learning (working together towards shared goals). It cites social constructivist perspectives that learning develops through social interactions.
Flexible learning trajectories	**Paper 3** discuss the course design which is based on the concept of flexible learning trajectories which allow for adaptability and customization of learning paths to meet individual needs and goals.
Social learning theory	Learning occurs through social interactions and knowledge sharing. **Paper 3** discusses how tools like Slack can support social learning by enabling knowledge sharing and construction through social interactions, similar to social media environments students are familiar with.
Constructive feedback theory	**Paper 3** discuss the importance of timely, meaningful feedback to support learning is discussed, citing research showing its role in successful teaching and learning.

Continuation of Table 2.1	
Theories, models, and frameworks	It's relevancy to OPDF for in-service teachers.
Programming pedagogy	**Paper 4** reviews previous studies that have identified challenges and issues related to programming pedagogy, such as the cognitive difficulties students face in learning programming concepts and the skills needed for effective programming instruction. The paper analyzes teachers' experiences through the lens of identifying "stumbling stones" (challenges) and "stepping stones" (enablers), drawing on previous studies that have used similar frameworks to understand issues in computing education. **Paper 8** acknowledges the common narrative that programming is a difficult subject to learn, especially for beginners. However, it aims to challenge this narrative based on the results. It draws on literature that has identified common difficulties faced by novice programmers when learning programming concepts like variables, loops, functions etc.
Teacher professional development	**Paper 4** situates its focus on in-service teacher training within the broader context of research on how to best support teachers in developing their competencies, especially for new topics like programming. The paper also considers how the educational system, curriculum policies, school resources, and other contextual factors shape teachers' experiences based on literature looking at these external influences. **Paper 7** cites research that has identified key characteristics of effective PD based on empirical studies. These include being content-focused, incorporating active learning, supporting collaboration, providing coaching/feedback, and being of sustained duration. **Paper 9** examines research on professional development (PD) interventions and programs for in-service teachers.
Computer literacy and computational thinking	**Paper 5** discusses the concepts of computer literacy, which refers to basic skills and knowledge of using computers, and computational thinking, which involves solving problems, designing systems and understanding human behavior by drawing on computer science concepts
Motivational theories regarding beliefs and attitudes	**Paper 5** discusses research on motivational beliefs and attitudes, which are important factors that influence students' learning and performance. Positive motivational beliefs and attitudes are associated with better academic achievement.
Project-based learning (PjBL)	PjBL is presented as the core learning approach used in the proposed "bridge model" in **paper 6**. The paper cites literature defining PjBL as organizing learning around projects and being based on constructivism.

Continuation of Table 2.1	
Theories, models, and frameworks	It's relevancy to OPDF for in-service teachers.
Connectivism learning theory	Connectivism is briefly mentioned as a theory that may apply to aspects of the course involving an online learning network in **paper 6**.
Constructivism	PjBL is said to be based on constructivist learning principles, with learners constructing knowledge through projects in **paper 6**.
Communities of practice	**Paper 7** references literature highlighting the importance of collaborative learning environments and communities of practice for informal workplace learning.
Reflective practice	**Paper 8** discuss the use of open-ended reflection allows applying the concept of reflective practice, where teachers reflect on their own learning and teaching challenges.
Literature mapping/review methodology	**Paper 9** follows the methodology of systematic mapping/literature mapping studies. It cites frameworks and guidelines from other studies that have conducted systematic mappings to structure its own mapping process.

2.5.2 Designing and Evaluating the Online Professional Development Framework: A Theoretical Perspective

The first research question (RQ-1) seeks to examine the evolution of research in the area of online learning theory. Online learning theory serves as a conceptual framework for analyzing the development of the field and identifying avenues for further exploration in online teacher education. By understanding the historical progression of research in this domain, we can gain insights into the trajectory of online professional development and the gaps that exist in current literature.

RQ-2 focuses on the design of engaging and active learning environments using project-based learning. Online learning theory provides valuable guidance on how to integrate interactive features, collaborative activities, and other design principles to optimize the learning experience in an online setting. By aligning the framework with online learning theory, we can ensure that the design of the professional development program is rooted in pedagogical best practices and tailored to the needs of online educators.

By understanding teacher attitudes and perceptions of challenges, RQ-3 connects to theories around how attitudes/beliefs impact behavior and self-efficacy, the role of experiences in adult learning, and the need to understand teacher-specific TPACK when designing effective technology-focused professional development. The insights gained can then be used to better design the PD program.

Finally, RQ-4 focuses on evaluating teacher perspectives, aligning with online learning theory's emphasis on learner experiences and feedback to enhance program design. By gathering insights directly from teachers engaged in the professional development program, we can gain a nuanced understanding of their experiences, challenges, and suggestions for improvement. This aligns with the learner-centered approach advocated by online learning theory and ensures that the framework is responsive to the needs and perspectives of its primary beneficiaries.

By addressing each research question through the theoretical lens of online learning, the framework achieves stronger conceptual coherence with the methodology. This alignment enhances the trustworthiness of knowledge claims derived from the findings, as the theoretical underpinnings provide a robust foundation for the design, implementation, and evaluation of the online professional development framework. The theoretical perspective plays a pivotal role in shaping the design and evaluation of an online professional development framework. By grounding the research questions in online learning theory, we can ensure that the framework is informed by established principles of effective online education and responsive to the needs of online educators. This approach not only enhances the rigor of the research but also contributes to the advancement of knowledge in the field of online teacher education.

The coherence of theories discussed in this chapter, lies in their emphasis on the interplay between social, cognitive, and teaching presences/knowledge during online learning. The CoI, TPACK, and Multimodal Model all align with this focus, providing a coherent foundation for pedagogical approaches. Additionally, the constructivist perspectives across these theories, such as social constructivism and constructivism, contribute to their coherence. Consistency is also evident across these theories, particularly in their shared emphasis on knowledge construction through social and collaborative means. Compatibility can be observed between TPACK and the Multimodal Model, as both analyze the knowledge domains and elements necessary for effective online learning. Similarly, the social-cognitive units of analysis in CoI and self-efficacy theory make them compatible. However, some inconsistencies exist between the theories. TPACK's transmission view of knowledge acquisition contrasts with the constructivist perspectives' emphasis on knowledge being actively constructed. Additionally, reconciling the individual focus of self-efficacy theory with the community-orientation of other theories may be required for a fully integrated understanding of online learning.

Chapter 3

The Case and Research Methodology

The professional development of in-service teachers is an essential component in enhancing the quality of education. This study focuses on the design of a professional development program aimed at equipping teachers with the necessary skills to incorporate programming into their teaching practices. The research seeks to gain insights into teachers' experiences and perspectives regarding the integration of programming in their pedagogical approach. The following sections describes the case and the research methodology that form the basis of this thesis.

3.1 The Case

The program is web-based with no physical gatherings and is aimed at secondary (8th to 13th grade) school teachers who are already employed. It consists of two courses. The first course covers programming fundamentals and uses the Gaddis [81] textbook. The students are activated through compulsory assignments. In addition, optional activities are provided for those who want to challenge themselves even further. The lectures are regularly broadcasted via webinars during the semester. The students are tested via a mini-project toward the end of the course.

The second course follows a similar structure to the first, except that the curriculum is significantly more extensive, and participants must pick their own learning path. Figure 5.2 shows the initial structure of the course. The portions of this course that change considerably from the previous one are highlighted in grey. The freedom to choose an individual learning path necessitates those assignments be tailored to specific students. Another distinction between the two courses is the assessment, where the second course emphasizes project-based learning. PjBL is an instructional approach that engages students in solving an authentic problem or creating an artifact that is meaningful to them. Some key elements of PjBL incorporated in the second course include:

- Authentic projects: Participants work on projects that mirror real-world problems and situations, not just textbook exercises.
- Participants voice and choice: Participants have voice and choice in their projects, allowing them to pursue their personal interests.

- 21st century skills: Projects require collaboration, communication skills, critical thinking skills, creativity, etc.
- Artifact/production: Participants create a tangible product or presentation as a result of their project work. In this case, an artifact like a lesson plan for teaching programming.
- Reflection: Participants reflect on their project work to reinforce learning.
- Presentations: Participants present their project work, processes and learning to audiences (their own classroom) for feedback.

By incorporating these key elements of PjBL, the goal was to enhance motivation and create a bridge between the training and teachers' practice. The projects resulted in artifacts like lesson plans that could be directly applied.

The majority of participants are in-service teachers from upper secondary schools (this matches to the latter three years of the K-12[1] educational system). In addition, some participants are elementary and lower secondary school teachers. Most of them teach STEM-related subjects, while the rest teach in a variety of areas (e.g., language, arts and crafts, music). Participants range in teaching experience from those who have just begun their careers to those who have spent decades in the field. In addition, participants come from a variety of schools and regions in Norway, where all schools adhere to a unified national curriculum. Teachers engage in the course with the assistance of their school, which releases 20% of the teachers' obligations (one day per week) so that they may complete the course. A national program of the Directorate of Education[2] partly covers the additional costs for the schools.

Table 3.1 shows the three cohorts involved in this research. The first cohort started in the fall of 2018 with a 100-person admittance restriction (capacity). Due to the tremendous demand for skill development, we extended the admissions limit to 200 in 2019 and 2020 at the request of UDIR[3]. Column A (candidates) of Table 3.1 shows the number of participants who showed up at the start of the course. Column B (took exam) indicates the number who took the exam, and column C shows the completion rate. This figure (column C) shows the ratio between those who showed up for the exam and those who started the program. The table shows a failure rate between 0-13% for the three cohorts.

Figure 3.1 depicts the different stakeholders involved in computer science education for in-service teachers. It identifies the main groups/individuals that have a role or interest in the CS education of in-service/practicing teachers. Stakeholders include the teachers themselves, students/pupils, school administrators, parents, government/policymakers, teacher education institutions/universities, and CS/industry professionals. All of these parties have some level of investment in or influence over how CS education is designed and delivered to teachers. For example, teachers need support from administrators, parents influence student choices, governments set curriculum standards, etc. Showing these stakeholders provides context on the complex ecosystem of actors involved beyond just the teachers themselves. Effective CS teacher education needs to consider the needs and perspectives of multiple

[1] https://en.wikipedia.org/wiki/K-12
[2] https://www.udir.no/in-english/
[3] https://www.udir.no/in-english/

Table 3.1: Cohorts, admission limit, candidates (A), participants registered for exam (B), completion rate (C)). The group starting in 2021 is not included in this research.

Cohort	Year	Admission limit	Autumn [82]			Spring [83]		
			A	B	C	A	B	C
	2021	300	285	271	95%	194	169	87%
III	2020	200	192	180	94%	173	170	98%
II	2019	200	194	182	94%	80	73	91%
I	2018	100	86	86	100%			

groups.

3.2 Design Research Approach

A mixed methods design science research approach [84] was adopted to address the main research question (Section 1.3.1). Design science research entails a rigorous process of developing products (artifacts) to solve problems, contributing to research, assessing the designs, and disseminating the findings to the relevant audiences. Some examples of artifacts are constructs, models, methods [85], or informational resources [86]. This study conceptualizes the professional development framework as the design artifact, which was developed through an iterative design process to address the problem of preparing in-service teachers for computer science education.

The DS process includes six steps: "problem identification and motivation, the definition of the objectives for a solution, design and development, demonstration, evaluation, and communication" [87, p. 46]. This procedure is constructed in sequential order; nevertheless, it is not expected that researchers will always continue from activity 1 through activity 6 in sequential order. In truth, they might begin at almost any level and expand outward [87]. In the first stage of the development of the professional development framework, our approach was problem-centered (see Figure 3.3). All steps were followed to design, evaluate, and communicate the first version of the program. In the second and third stages, the process started with an existing artifact (program design) (see Figure 3.4) and followed activities 1 to 6 to improve the program design.

The research examines teachers' experiences and perspectives during the development process, with the understanding that these represent individual perceptions rather than objective realities. An interpretive approach was taken to analyze qualitative data from teachers' points of view. A mixed methods approach combining qualitative and quantitative data was employed to gain a rich understanding of teachers' experiences while also capturing measurable impacts. Qualitative data in the form of interviews and reflection notes provided insights into teachers' subjective realities, while quantitative surveys facilitated comparisons and statistical analysis. An iterative DSRM process allowed for continuous refinement of the framework based on emerging insights.

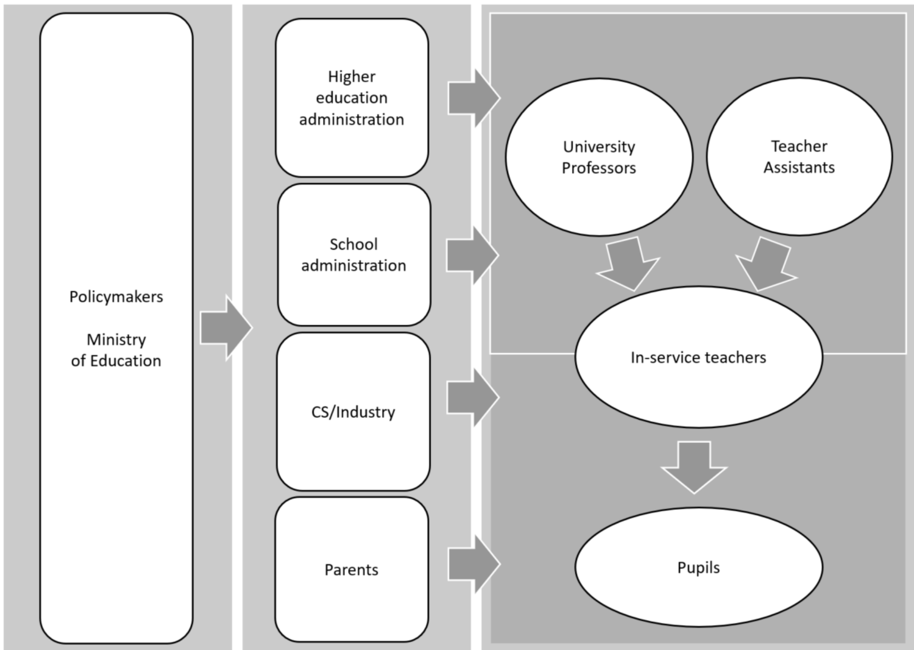

Figure 3.1: Stakeholder in CS education of in-service teachers and their interrelationships.

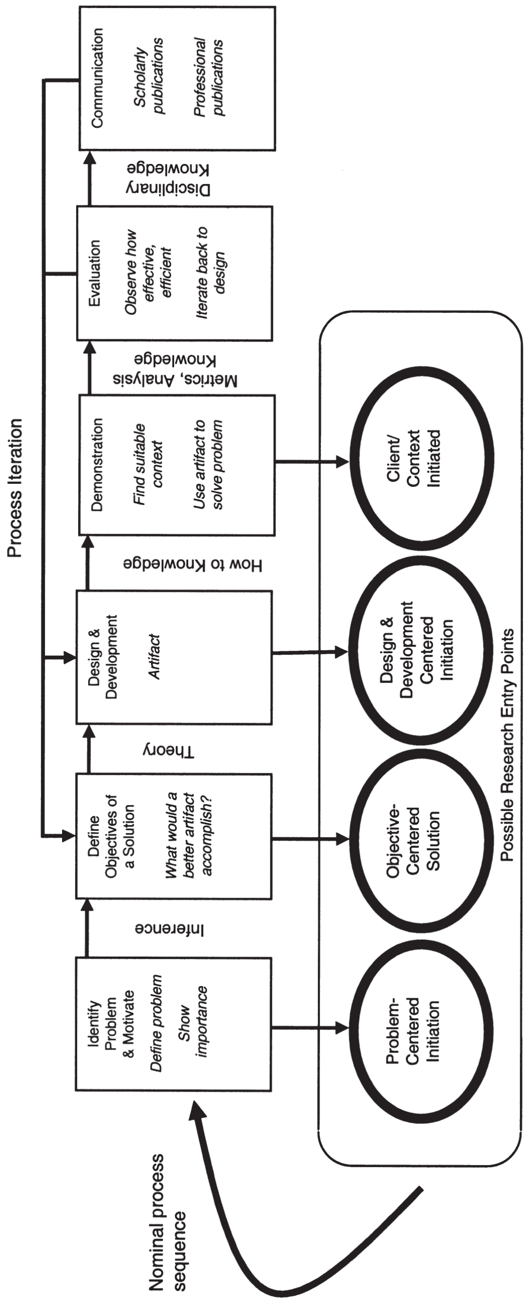

Figure 3.2: Design Science Research Process Model adapted from Peffers *et al.* [87].

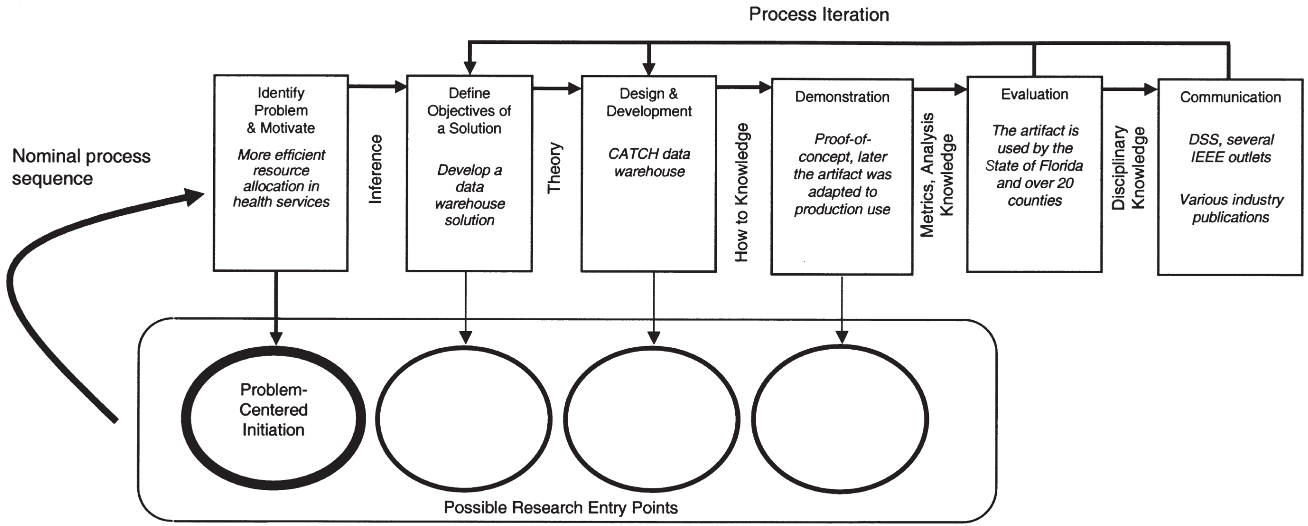

Figure 3.3: Design Science Research Process Model: Problem-Centered Approach. Adapted from Peffers *et al.* [87]. Iteration 1.

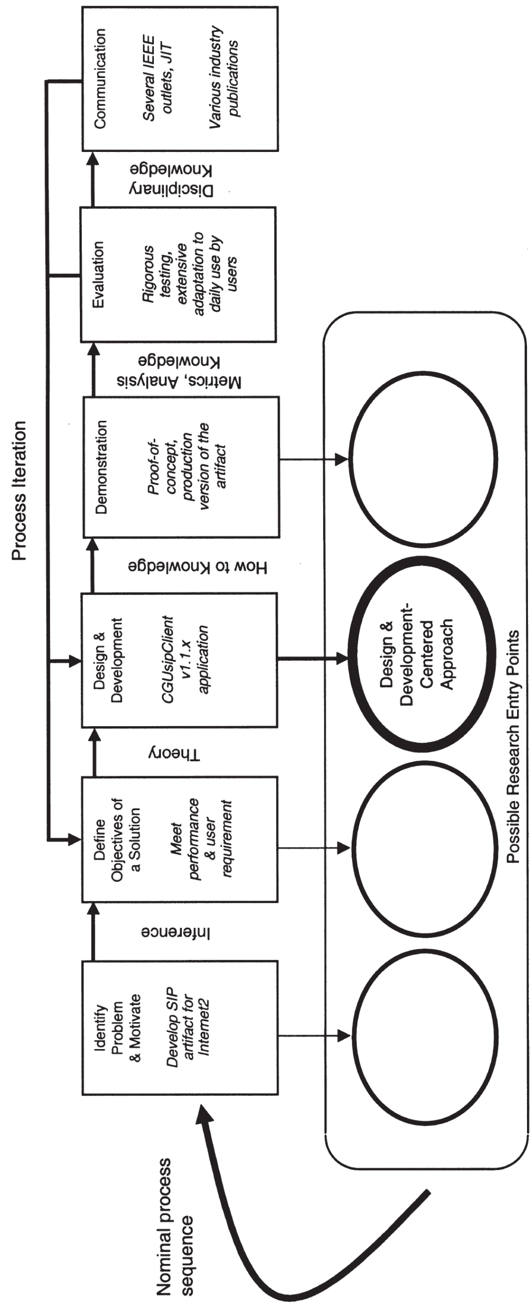

Figure 3.4: Design Science Research Process Model: Design- and Development-Centered Approach. Adapted from Peffers *et al.* [87]. Iterations 2 & 3.

3.3 Theoretical Starting Points

DSRM as a guiding framework, provides a structured process for developing and evaluating innovative artifacts to solve practical problems [85]. The iterative processes involved in this framework aligns well with the goal of iteratively developing and assessing the professional development framework.

In Section 2.1, online learning is defined, and its key characteristics explored. This provided an important theoretical foundation for conceptualizing the online professional development framework (OPDF) designed for in-service teachers in the study. One influence was the emphasis placed on flexibility and accessibility. Recognizing the diverse geographical locations and time constraints of teachers, the OPDF was designed as an entirely online program. This aligned with online learning's removal of physical and temporal barriers. It also allowed for asynchronous participation, appealing to teachers' scheduling needs. The OPDF also incorporated flexibility in pacing by offering both synchronous and asynchronous course components. Teachers could engage at their own rhythm through recorded lectures and self-paced study. This self-directed element respected the autonomy and control over scheduling that online learning aims to provide. By leveraging technology, the OPDF could deliver content through rich multimedia and interactive activities as online learning does. This variety in presentation methods helped cater to different learning preferences among teachers. Finally, the OPDF incorporated collaborative project-based work and online community elements. This aimed to facilitate the social connectivity emphasized in online learning to mitigate isolation and foster knowledge sharing. In these ways, directly applying the principles around flexibility, pacing, engaging delivery, and social learning outlined in Section 2.1 resulted in an OPDF design aligned with established online pedagogical best practices from the start.

Additionally, Anderson's online learning model [40] emphasizes interaction and active learning through collaborative activities and projects. The community of inquiry (CoI) framework also informed the methodology. As described in Section 2.2.2, the CoI model proposes that learning occurs through cognitive, social and teaching presences. Papers 1, 3, 4, 6, 7, 8 and 9 demonstrated conceptual relevance to these presences, showing how they influenced factors like collaborative project-based activities and instructor facilitation. TPACK underscores the need for teachers to develop integrated knowledge of technology, pedagogy, and content. Similarly, the Multimodal Model for Online Education presented in Section 2.2.4 identified elements needed for integrated virtual education, including the three presences. Papers 3 and 6 directly related aspects of their methodology, like using collaborative tools, to this model. Active learning theory, outlined in Section 2.2.3, also provided a starting point through its emphasis on engaged, collaborative experiences. This impacted choices like incorporating project-based learning approaches in Papers 3, 4, 6 and 7 to promote participation and knowledge application.

Theories of teacher professional development also provided guidance. Darling-Hammond *et al.* [29] outlines seven criteria of successful teacher professional development that contribute to improvements in teacher behavior and enhanced student learning outcomes. These criteria informed the design of project-based learning activities and mechanisms for feedback and reflection. Principles from programming pedagogy research were considered [75]. Literature on common difficulties faced by novice programmers [15, 16] helped shape the content and sequencing of topics.

Together, these learning theories provided a strong theoretical foundation for conceptualizing the research methodology in a way that addressed both the practical goal of developing an effective training program and the academic aim of contributing new knowledge to the field. The study was thus firmly grounded in relevant online, collaborative and active learning principles.

3.4 Research Design Methodology

The project was implemented in three iterations, as depicted in Figure 3.5. The initial version of the program was deployed in 2019, with further revisions in 2020-2021 based on lessons learned. Section 3.6 describes the three iterations of studies (Sections 3.6.1 to 3.6.3) that were conducted as part of developing and evaluating the professional development framework for teachers. Each iteration built upon the previous one to further refine the framework based on the findings.

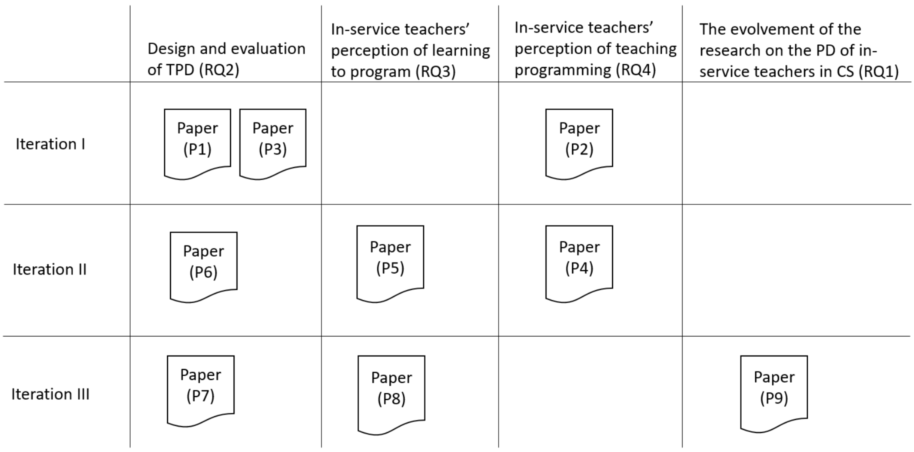

Figure 3.5: Overview of the research process and the corresponding papers.

Figure 3.6 presents a timeline that outlines the planning, data collection, and analysis processes undertaken across the three iterations of this design science research study. It illustrates the structured methodology employed to systematically develop and evaluate the professional development framework. In Iteration 1, planning for the initial program and first study spanned several months. This involved developing research instruments and protocols. Data was then gathered during and after delivering the program to the first cohort of teachers. Once collection was complete, analysis and writing commenced to code and identify themes in the data, with findings documented in a report. Iteration 2 followed a similar process, building on the results of the previous iteration. Planning incorporated refinements to address areas for improvement identified previously. A new cohort provided data that was again analyzed and written into a report. For Iteration 3, the timeline indicates planning was informed by the two prior iterations. Data collection involved an even larger cohort. Analysis and writing of the final report integrated all findings across the iterations.

Figure 3.7 provides a visual representation of the three iterations that were conducted as part of the design science research methodology used in this study. The figure illustrates how each iteration built upon the previous one to further develop and refine the professional development framework for in-service teachers. Iteration 1 focused on designing an initial flexible learning trajectory program. This involved establishing the foundational structure and content of the framework. Through evaluation, lessons were learned about areas for improvement. Iteration 2 delved deeper into specific challenges and opportunities related to teaching programming from the teacher's perspective. A project-based learning environment was introduced to engage participants. Feedback from this iteration informed modifications to better support the teachers. Iteration 3 centered around evaluating the experience of teachers with the professional development program. Core programming concepts that proved difficult were also identified. Larger cohorts of over 180 teachers provided insights in the third iteration.

Table 3.3 provides a comprehensive overview of the evaluation methods used in each study across the nine research papers that comprised this design science research project. The table synthesizes the mixed methods approach employed to systematically develop and assess the professional development framework over multiple iterations. The first column lists each paper numerically for easy reference. The second column then indicates the specific evaluation method utilized in that paper, such as questionnaires, interviews, or literature reviews. The third column gives a brief description of how the method was operationalized. For example, it may specify what types of questions were included in a questionnaire or the number of participants interviewed. These descriptions provide context around the application of each method. A range of quantitative and qualitative techniques are represented, from surveys using Likert scales to thematic analysis of reflection notes and transcripts. The mixed methods triangulate data sources and allow comprehensive evaluation from different perspectives. It conveys at a glance the breadth of techniques used to gather feedback over the course of framework development.

Table 3.2 provides an overview of the nine papers included in the thesis. It summarizes the alignment between each paper and the research questions, methods, contributions, and iterations that were central to the design science research process undertaken. The table first lists the nine papers from P1 to P9. For each paper, it then indicates which of the four main research questions the paper addresses. This helps to map how the different studies relate to the core areas of inquiry that guided the overall research. The methods column describes the methodology used in each paper. A variety of qualitative and mixed methods were employed, including questionnaires, interviews and reflection note analysis. This diversity of approaches allowed comprehensive investigation of the research questions. The contributions section outlines the key findings and outcomes generated by each paper. Finally, the iterations column links each paper to the specific iteration of the design science research process it relates to.

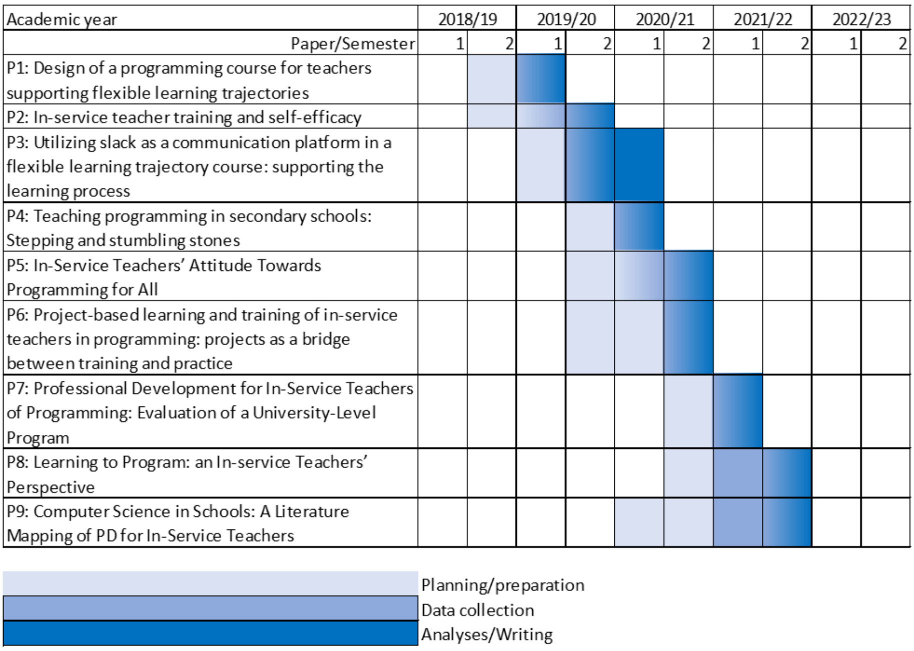

Figure 3.6: Timeline of planning, data collection, and analysis processes

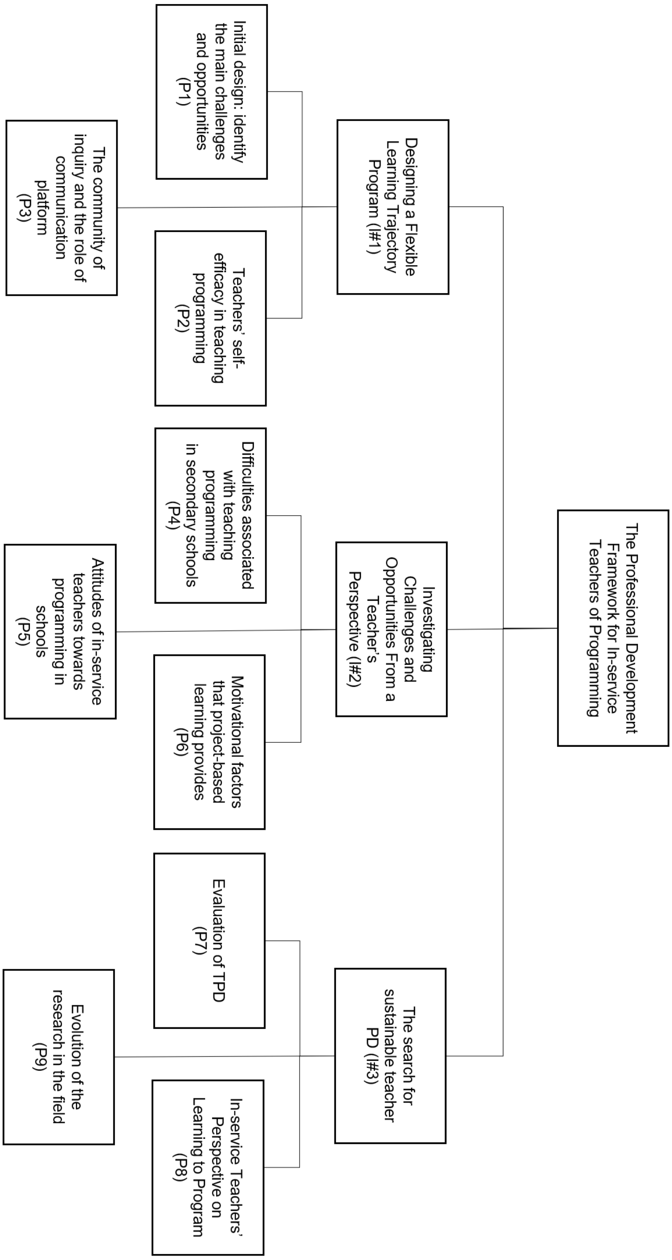

Figure 3.7: The three iterations in developing the professional development framework

Table 3.2: Overview of papers, contributions, research questions, methods, and iterations

	P1	P2	P3	P4	P5	P6	P7	P8	P9
Research questions									
RQ-1	x			x	x	x	x	x	x
RQ-2	x		x			x	x		
RQ-2.1	x								
RQ-2.2			x						
RQ-2.3						x			
RQ-2.4							x		
RQ-3					x			x	
RQ-3.1					x				
RQ-3.2								x	
RQ-4		x		x					
RQ-4.1		x							
RQ-4.2				x					
Contributions									
C1									x
C2	x		x			x	x		
C3					x			x	
C4		x		x					
Goals									
G1									x
G2	x		x			x	x		
G3					x			x	
G4		x		x					
Methods									
Qualitative		x		x	x		x		
Mixed	x		x			x		x	
Literature Mapping									x
Iterations									
iteration 1	x	x	x						
iteration 2				x	x	x			
iteration 3							x	x	x

3.5 Evaluation Methodology

Evaluation methodologies play a crucial role in shaping the quality and depth of research outcomes. In this thesis, a mixed methods approach was adopted, incorporating both qualitative and quantitative data collection techniques to provide a holistic understanding of the subject matter. This approach facilitated the validation of findings through triangulation, thereby enhancing the overall credibility of the study.

Qualitative data, comprising reflection notes, interviews, and communication platform messages, was instrumental in capturing rich experiences and diverse perspectives. The use of thematic analysis allowed for a systematic exploration of this data, with an emphasis on identifying recurring patterns and developing a narrative that encapsulated the essence of the information gathered. The process of coding and theme refinement was essential in ensuring that the final themes reflected the data and were substantiated by relevant quotes, thereby enriching the qualitative findings with depth and context.

On the other hand, quantitative data, obtained through surveys featuring closed and open-ended questions, enabled the application of statistical analysis to identify trends and patterns. Descriptive statistics were employed to calculate frequencies and percentages for closed survey questions, providing valuable insights into levels of satisfaction, confidence, and other measurable variables. Furthermore, the thematic analysis of open-ended survey responses complemented the quantitative findings, offering a more nuanced understanding of the data and enriching the overall analysis. The integration of qualitative and quantitative data was facilitated through an embedded mixed methods design. This strategic approach allowed for the seamless alignment of quantitative findings with qualitative insights, thereby reinforcing the robustness of the research outcomes.

The units of analysis, including reflection notes, surveys, and interviews collected at different time points, allowed for a comprehensive evaluation of the research questions, addressing both the effectiveness of the program and the factors influencing its outcomes. The integration of qualitative and quantitative findings through mixed methods further enriched the analysis, providing a comprehensive understanding of the research outcomes.

3.5.1 Data Collection and Analysis Process

The data collection process encompassed three iterations, each utilizing distinct methods to gather insights. The data collection process in iteration 1 (Section 3.6.1) utilized a mixed methods approach across three papers to gather both quantitative and qualitative data. In Paper P1, questionnaires were distributed to collect quantitative feedback on teacher satisfaction, while reflection notes were analyzed to provide qualitative insights into their experiences. Paper P2 involved qualitative semi-structured interviews conducted approximately one year after the initial training to evaluate the long-term impact on teachers' self-efficacy. Paper P3 employed a mixed survey research design combining quantitative Likert scale questions with qualitative open responses and analysis of messages from the communication platform used during the course.

The data collection process in iteration 2 (Section 3.6.2) utilized qualitative methods to gain deeper insights into challenges and opportunities from the teachers' perspective. Paper

P4 gathered reflection notes to identify difficulties related to teaching programming, while Paper P5 collected reflection notes to understand teachers' attitudes. Paper P6 employed a mixed methods approach, utilizing a questionnaire with Likert scale statements to collect quantitative data as well as an open-ended question for qualitative feedback on the new project-based learning environment. Insights from analyzing these qualitative data sources informed enhancements to the professional development framework for the next iteration.

The data collection process in iteration 3 (Section 3.6.3) utilized qualitative and mixed methods approaches in papers P7 and P8, utilizing interviews and surveys. P9 conducted a literature mapping. The focus of Iteration 3 was a comprehensive evaluation of the PD program and identification of areas for further refinement.

The transcription of interviews facilitated understanding of participants' narratives, followed by familiarization with the data through reading and re-reading to identify recurring patterns and themes. The initial coding process, incorporating both deductive and inductive approaches, allowed for a comprehensive exploration of the data. Subsequently, the collation of codes into potential themes and their review and refinement against the coded data ensured the emergence of robust and meaningful themes. The final step involved the selection of quotes to illustrate the identified themes, adding depth and context to the analysis.

In parallel, the quantitative data analysis process involved the entry of survey responses into Excel, enabling systematic organization and manipulation of the data. Descriptive statistics, including frequencies and percentages for closed questions, as well as measures of central tendency, were calculated to provide a quantitative understanding of the participants' responses.

The integration of qualitative themes and quantitative results facilitated a comprehensive and nuanced analysis, allowing for the identification of contrasts, comparisons, and discrepancies. This mixed methods approach enabled the drawing of integrated conclusions, presenting a holistic view of the program's impact. The presentation of findings in joint display tables further enhanced the clarity and coherence of the results.

Rigor was ensured throughout the data collection and analysis processes through an iterative coding process, researcher triangulation, and participant validation. These measures safeguarded the credibility and trustworthiness of the findings, enhancing the overall robustness of the evaluation. Furthermore, data anonymization was rigorously implemented to uphold participant confidentiality and privacy, adhering to ethical standards and regulations.

Overall, the majority of papers integrated both deductive and inductive elements for a comprehensive exploration of teachers' experiences, allowing confirmation of existing ideas alongside generation of novel insights.

3.6 Studies on Teachers' Attitudes and Development

3.6.1 Iteration #1: Designing a Flexible Learning Trajectory Program

In the first iteration, we began with the "Problem Identification and Motivation" activity in design science research methodology (problem-centered approach), in which we formulated research questions based on the program's requirements (activity 1). The primary need was to provide a completely online course for secondary school teachers who were already employed. Using the problem statement from the first activity, we defined the solution's goals (activity 2). We aimed to establish a flexible course where researchers and practitioners collaboratively find priority areas to focus on to raise the program's quality. An initial version of the program was developed and executed for the program's first cohort of teachers (activity 3). This was demonstrated through implementation with the first teacher cohort in activity 4. Evaluation methods like questionnaires and reflection notes were used to assess the design in activity 5. Findings were communicated through P1-P3 in activity 6.

A mixed methods approach using questionnaires and analysis of reflection notes was employed to evaluate the effectiveness and outcomes of the initial program design. Questionnaires gathered quantitative data on teacher satisfaction, while reflection notes provided qualitative insights into their experiences. This mixed methods evaluation assessed how well the program achieved its intended goals and contributed to teachers' learning. It also identified areas for refinement.

In iteration 1, the evaluation process of the teacher training program involved collecting participant feedback, publishing experience reports on the initial design, evaluating self-efficacy impacts, examining the learning platform, and disseminating the findings. This provided data to identify challenges and priority areas to refine the program for the next iteration. Main steps were:

- P1: Published an experience report explaining the overall program structure and flexible learning trajectory approach. It provided an overview of the intended design and aims to establish priority areas for improvement.
- P2: Focused on evaluating the long-term impact on teachers' self-efficacy in teaching programming concepts learned in the program. Interviews were conducted with teachers about one year after completion.
- P3: Investigated the use of the communication platform Slack to support collaboration and community-building. It analyzed messages and survey data to evaluate how well Slack facilitated the learning process.
- Dissemination: The evaluation results from Papers 1-3 were presented at conferences to get feedback from the research community.

In paper 1 (P1), qualitative analysis of teachers' reflection notes and quantitative data from questionnaires was used to understand teachers' perspectives on various aspects of the program, such as structure and usefulness. The methodology in paper 2 (P2) involved qualitative semi-structured interviews with teachers approximately 1 year post-training, analyzed thematically with a focus on self-efficacy constructs to understand impacts on attitudes and development over time. Paper 3 (P3) used a mixed survey research approach combining quantitative Likert scale data with qualitative open responses, activity tracking, and direct quotes to develop a holistic understanding of teachers' experiences, attitudes, levels of confidence and development throughout the course.

Units of analysis were "reflection notes from teachers", messages in the communication platform (Slack), and "questionnaire responses". These data provided valuable insights into the strengths and areas for improvement of our initial program. We used this feedback and the insights gained from these submissions to refine the professional development program.

The data were coded and analyzed thematically to identify common themes, patterns, and insights. This involved reading the data closely, assigning codes line-by-line, grouping codes into potential themes and reviewing themes against the data. Responses to closed-ended questions were analyzed statistically using descriptive methods like frequencies and percentages to quantify feedback. Open-ended questions were also analyzed thematically. For details of how this process was carried out in each paper, see [5, 78, 79] and appendices [A, B].

3.6.2 Iteration #2: Investigating Challenges and Opportunities From a Teacher's Perspective

Iteration 2 began with the existing artifact from Iteration 1 (see Figure 3.4 (DSRM)). Activity 1 saw areas for deeper investigation identified through evaluation, leading to formulation of new research questions. Activity 3 involved qualitative data collection methods like interviews and reflections to explore challenges and opportunities from the teacher perspective. Insights informed improvements to the framework design for Activity 4.

This iteration focused on preparing teachers to teach programming by exploring challenges and opportunities from a teacher's perspective in a project-based learning environment. The first research objective was identifying opportunities and difficulties associated with teaching programming in secondary schools (P4). We took a comprehensive approach and considered their impact on policy and teacher education. This strategy highlights the significance of seeing teachers as individuals and components of a complex ecosystem that influences their performance. The second research (P5) investigated the advantages of computational thinking, coding, and programming in the classroom, as well as the attitudes of in-service teachers about programming in schools. The third article (P6) investigated the motivational factors that project-based learning provides. We proposed a new "bridge model of programming for in-service teachers," which aimed to integrate theory and practice by letting them create an artifact (lesson plan) during their project in the course.

Papers 4-6 used qualitative data collection methods, employing interviews and analysis of reflection notes. This allowed an in-depth investigation of teachers' experiences with project-based learning approaches and their impact on motivation and learning. In iteration 2, three research papers (P4, P5, and P6) provided insights into teaching programming. Qualitative data collection involved gathering reflection notes from teachers, which were analyzed thematically to identify challenges and opportunities. The program was refined based on these findings to address challenges for the next iteration. This process represented a significant step towards enhancing programming education by engaging with the experiences and perspectives of teachers, exemplifying a robust and systematic approach to program development.

Qualitative research methods, including interviews and reflection note analysis, were applied in Papers 4-6. This facilitated an understanding of teachers' perspectives on challenges, opportunities, and factors influencing their professional development when engaging in project-based learning activities.

"Reflection notes" collected from teachers in the revised program and "Interview transcripts" from in-depth interviews conducted with teachers to understand their views.

Thematic analysis of reflection notes and interviews: This followed an iterative process of coding, theme identification and reviewing. Both deductive coding based on prior research and inductive coding of emergent ideas were used. Codes were grouped into overarching themes. For details of how this process was carried out in each paper, see [2, 77, 80] and appendices [A, B].

3.6.3 Iteration #3: The search for a sustainable teacher professional development program

Iteration 3 again began with the existing artifact, with activity 1 focusing on evaluating strengths, weaknesses and support mechanisms based on teacher experiences. Activity 3 involved quantitative and qualitative data collection. Findings revealed core concepts teachers struggled with, aiding framework refinements in activity 4.

The first study (P7) was about in-service teachers' experiences of what they experience as the strengths and weaknesses of this PD program and what mechanisms can be offered to support the long learning process beyond the duration of this PD. The second study (P8) investigated the core programming concepts teachers perceive as problematic to learn and teach. Additionally, we wanted to know which concepts they anticipate challenging for their pupils. Throughout the entire project, a comprehensive literature review was conducted. However, in order to solidify the literature mapping, the third study (P9) focused on examining the evolution of research in the field. Specifically, we delved into the various types of interventions described in the literature and identified the key factors that influence the outcomes of these interventions.

Mixed methods using questionnaires and interviews were utilized in Papers 7-8 to evaluate various aspects of the program, including strengths/weaknesses and impact on teacher development. The evaluation process in iteration 3 of the educational program encompassed a comprehensive array of methodologies, yielding a nuanced understanding of the program's impact and informing future research directions. By integrating diverse methods of data collection, analysis, and reflection, the evaluation process served as a robust foundation for ongoing improvement and innovation within the educational program.

Both quantitative questionnaires and qualitative interviews were employed in Papers 7-8 to gather data on teachers' experiences, views on challenges, and attitudes towards the program. This provided insights into factors shaping professional growth.

"Questionnaire responses" collected from teacher feedback surveys and "Interview transcripts" from interviews. The unit of analysis in the literature mapping study is the individual piece of scientific articles that is being examined and analyzed.

The questionnaire analysis involved the statistical analysis of quantitative data. This included entering the responses to closed-ended and rating scale questions into a Excel. Descriptive statistics were then used to identify patterns in the responses, which involved calculating frequencies, percentages, and measures of central tendency.

Thematic analysis was conducted for the interviews. This involved an iterative coding and theme development process, where the interview transcripts were thoroughly read and re-read to gain familiarity with the data. Initial codes were assigned to segments of text related to the research questions, and these codes were then grouped into potential themes through an iterative process of reviewing, refining, and organizing them. The final themes were defined and refined by identifying the story they captured about the data, and representative excerpts were reported to support these themes.

We also employed mixed methods analysis to integrate the qualitative interview findings with the quantitative survey results. This allowed us to obtain a holistic understanding of the data by leveraging the strengths of both analytical approaches. The integration of results from the quantitative and qualitative strands of analysis during interpretation provided an overview as well as depth, and contrasts and comparisons between the two datasets were made to triangulate and validate the findings. Any discrepancies were explored to provide a more complete picture of the teachers' experiences, and integrated conclusions were drawn to summarize the findings.

3.7 Overview of the methods used in each paper

Table 3.3 provide a comprehensive overview of various papers and the methods employed in each. The columns include the paper id, method, and a detailed explanation of the method utilized.

Table 3.3: Overview of the methods used in each study (Paper, method, and description)

Paper	Method	Description
P1	Mixed	After the course, we asked participants to answer a questionnaire to evaluate the program and informed them that anonymized data would be used in the research. Statements in the questionnaire were (1) course modules give me a good overview of learning objectives, (2) webinars were helpful, and I feel I increased my knowledge, (3) I find that topics covered in the modules were relevant to my practice, and (4) exercises had relevant content and was not too hard to resolve. In addition to the quantitative data, we gathered and examined reflection notes (qualitative data) from each student who passed the course.
P2	Qualitative	Ten participants in the 2018/2019 cohort were interviewed.
P3	Mixed	We surveyed with close- and open-ended questions before the start of the course *Applied programming for teachers* [83] where 186 candidates answered the survey.

Continuation of Table 3.3		
Paper	Method	Description
P4	Qualitative	We performed a thematic analysis on the data. Since we previously conducted an interview study with teachers who had completed the program once [78], we based our code and themes on this data set. Although the research questions in the two studies were different and participants were not in the same cohort, they shared the overall goal of learning about challenges related to instructing programming. New codes were added during the analysis [88]. This study was, therefore, a combination of a deductive and inductive theme analysis.
P5	Qualitative	This research collected reflections from the classes of 2019 and 2020. We analyzed 22 reflection notes from 2019 and 306 from 2020 (169 from the start and the rest when they completed the program).
P6	Mixed	For the quantitative data collection, we used a questionnaire containing the following statements by the end of the course [2, pp. 265–266]. For the qualitative part of the study, we defined an open-ended question, asked teachers to deliberate on the entire course, and received 17 responses. The reflection notes of the 173 participating teachers provided the supplemental data set for the investigation described in this paper. We did not perform a comprehensive study of the reflection notes; we relied upon a few quotations to confirm the quantitative findings.
P7	Qualitative	This study used interviews in the semi-structured form to analyze the study inquiries and capture teachers' viewpoints on their PD program. The interview guide (see Appendix A) was created based on the underlying factors [27, p. 126].
P8	Mixed	We used a questionnaire with both open- and closed-ended questions. We formulated 16 statements based on what we found in the literature as common challenges for beginners to learn to program. Statements are shown in the paper 'Learning to Program: an In-service Teachers' Perspective,' p. 125.
P9	Literature Mapping	This investigation used systematic mapping (SM) to collect, organize, and classify all accessible data on a particular subject. SM is helpful as a brainstorming and scoping tool at the beginning of an investigation.

3.8 Summary

This chapter discusses the case study that was conducted as part of the research project and the research methodology that was employed. It provides an overview of the Design Science Research Methodology (DSRM) process and explains how it was applied in the development of a professional development framework for in-service teachers. Additionally, the chapter presents an overview of the three iterations that were completed as part of the research project.

Chapter 4

Results

This section presents the results achieved in this PhD project. The papers presented in this thesis consist of nine peer-reviewed articles: eight conference papers, and one journal article. The papers are attached as appendices.

For each paper, there will be a presentation of:

1. Title
2. Authors' names
3. Authors' contributions
4. The name of the publisher
5. Abstract of the paper
6. A short description of the main findings
7. Relation to the research questions.

4.1 Paper I (P1)

Title: Design of a programming course for teachers supporting flexible learning trajectories

Authors: Majid Rouhani, Monica Divitini, Vojislav Vujosevic, Sondre Stai, Hege Anette Olstad

Authors' contribution: Rouhani had the primary responsibility for the idea, writing, and presenting the findings at the conference.

Published: In Proceedings of the 8th Computer Science Education Research Conference (pp. 33-38).

Abstract: How to design an online flexible learning trajectory course where students are in-service teachers with varied level of programming knowledge, interests, and different application need? This paper presents the design of such a course for teachers on applied programming. The main learning objective of the course is to provide in-service teachers with insight into how programming can be used to create digital solutions. The course is practically directed and emphasizes programming as a constructive and creative tool. The course is aimed at

> teachers in secondary schools. The paper describes the main design choices of the course. Based on the experience with the course, the paper reflects on the challenges to design courses that do not support a single learning path for all the students, but rather aims at providing a context where students can identify and follow the learning path that is best fitting for their competencies, interests, and needs of the local practices.

A description of main findings:

The study revealed that 73 out of 80 students completed the course, resulting in a retention rate of approximately 91%. Furthermore, the utilization of programming using micro:bit, Arduino, and LEGO Mindstorms, with over half of the projects employing block-based programming, indicates a practical and hands-on approach to learning.

The wide array of topics covered in student projects demonstrates the students' ability to select relevant and diverse subject matter. The identification of different learning trajectories based on individual subject areas, skills, and interests underscores the flexibility and adaptability of the course, catering to the unique needs of each student. The use of communication channels such as Slack to facilitate interactions and knowledge sharing among students of varying skill levels exemplifies the effectiveness of fostering collaborative learning environments.

The study's findings on student reflections indicate a notable increase in engagement, motivation, and understanding of the practical applications of programming in teaching. The insights gleaned from these reflections shed light on the challenges encountered by teachers in implementing programming activities in their own classes, offering valuable considerations for future course enhancements. The course evaluation survey predominantly yielded positive feedback on the flexibility and relevance of the course for teaching practice, affirming its value to the participants.

The critical areas discussed in this study, centered around ongoing refinement of student guidance, scaling up efficiently, content focus, learning support structures, and continuous evaluation and improvement through a design-based research lens. The assessment process emerged as a key challenge, particularly in light of increasing student numbers, necessitating a restructuring to ensure scalability and fairness. Moving forward, addressing these identified challenges will be crucial in enhancing the course's overall effectiveness and ensuring its continued relevance and impact in the realm of applied programming in education.

The relation to research questions: Paper I directly responds to RQ2.1 by presenting the flexible course design and evaluating challenges/opportunities of that design based on the first iteration of the program. It addresses the question of how to design an online flexible learning course for diverse teachers.

4.2 Paper II (P2)

Title: In-Service Teacher Training and Self-efficacy

Authors: Jørgen Thorsnes, Majid Rouhani, Monica Divitini

Authors' contribution: Data analysis for this study is related to programming courses I have developed and implemented for in-service teachers. Additionally, I contributed in writing the paper and presented the results at the conference.

Published: In International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (pp. 158-169). Springer, Cham.

Abstract: Programming is increasingly introduced in secondary schools, both as a stand-alone subject or integrated into other subjects, leading to growing attention to the training of in-service teachers. Teachers need to learn both (a) how to program and (b) how to teach programming, often in the context of different disciplines. The paper explores the impact of a university-level training program offered to in-service teachers, with a focus on teachers' self-efficacy in teaching programming. The paper reports the interviews with ten teachers after about one year they have completed the program. The results indicate that the training has improved teachers' self-efficacy, and the impact is lasting in time. Also, some teachers expressed concerns about their skill level in programming, but this does not necessarily associate with lower self-efficacy in teaching programming. The paper presents the results from the study and some implications for the design of training of in-service programming teachers.

A description of main findings: The findings reveal a generally positive attitude among teachers towards teaching programming and utilizing it across various disciplines. However, concerns were raised regarding the time-consuming process of implementing programming widely due to the lack of expertise among teachers. The importance of teacher collaboration emerged as a significant factor, with those having colleagues to work with reporting beneficial outcomes. Conversely, some teachers reported encountering hostility from peers and expressed concerns about gender stereotypes in the context of programming education.

Challenges related to assessment were identified by some teachers. Despite this, the majority felt capable of motivating students to learn programming and believed they could effectively explain concepts and provide alternative explanations. Additionally, most teachers expressed confidence in developing lesson plans, with some relying more on adapting existing resources.

The study revealed that the majority of teachers transitioned from feeling unable to teach programming to feeling capable after receiving training. Additionally, some teachers who already felt capable reported feeling more secure in their abilities after training. The training had a positive impact on teachers' attitudes towards teaching programming.

Programming skills tended to diminish over time without regular use, although they could be easily refreshed. In contrast, self-efficacy for teaching programming was found to be enduring, showing minimal decline over time. Moreover, the experience of teaching programming further bolstered teachers' teaching self-efficacy.

The study's implications emphasize the need for teacher training programs to support ongoing development. It underscores the importance of addressing concerns related to the implementation of programming education, fostering collaborative environments, and providing support for teachers to enhance their programming skills over time. Additionally, the findings underscore the enduring impact of training on teachers' self-efficacy and the need for continuous reinforcement of programming skills.

The relation to research questions: This study is related to RQ4.1, where we investigate the program's impact on in-service teachers, focusing on teachers' self-efficacy in instructing programming.

4.3 Paper III (P3)

Title: Utilizing Slack as a communication platform in a flexible learning trajectory course: supporting the learning process

Authors: Majid Rouhani

Authors' contribution: Single author

Published: In Proceedings of the 9th Computer Science Education Research Conference (pp. 1-7).

> *Abstract*: Online and flexible programming courses for in-service teachers with varied level of programming knowledge, interests, and different application need might, on the one hand, be challenging. On the other hand, this flexibility might provide convenience and promote learning. With the advancement of technology and the change of habits for the use of the traditional communication platform, educators need to explore practical ways to communicate. This paper presents the results of an investigation into Slack's use as a communication platform to enhance collaborative learning and online activities. We use messages from Slack channels and survey data used in the course to investigate the effectiveness of the communication process between students, instructors, and TAs and to what extent students believe the tool supports reaching their goals. The most common form of online collaboration was the need for clarifications, a deeper understanding of course topics, assessments, and sharing their programming knowledge. Teachers worked towards achieving their goals through collaborative development of programming skills by asking questions and sharing competence. A smaller number of teachers were involved in contribution, which was then used by the majority. This paper summarizes the author's findings on Slack's use as a communication platform in an online flexible learning trajectory course.

A description of main findings: The integration of Slack into a flexible programming course for teachers has been highly effective, with a 98.8% adoption rate among students. Registrations occurred swiftly after the course began, demonstrating eagerness to engage with the course and its communication tools. Over 8,500 messages were exchanged, indicating Slack's success in fostering interaction and collaboration.

Weekly active users averaged between 86 and 116 across the two courses, showing consistent involvement. Student feedback praised Slack for its communication efficiency, quick teacher responses, and contribution to a dynamic learning environment. The course structure, with its mix of module types, was well-received for meeting diverse learning preferences. Assessments were deemed appropriately challenging, and the pilot project was seen as practically applicable to the students' teaching. Workload balance was mostly considered 'just right,' with a significant increase in students' confidence in teaching programming noted. While most feedback was positive, some students suggested the need for deeper learning materials, clearer relevance of content, and more teaching examples. These points highlight areas for course improvement. In summary, Slack has significantly supported and enhanced the online learning experience, with its high engagement and positive impact on the course structure and assessments. Despite some areas for improvement, Slack has proven to be an effective tool for educators learning programming.

The relation to research questions: This study relates to RQ2.2, where we wanted to investigate how communication platforms such as Slack can support students' learning processes in a flexible learning trajectory teaching environment.

4.4 Paper IV (P4)

Title: Teaching Programming in Secondary Schools: Stepping and Stumbling Stones

Authors: Majid Rouhani, Veronica Farshchian, Monica Divitini

Authors' contribution: Rouhani had the primary responsibility for the idea, writing, and presenting the findings at the conference.

Published: Interaction Design and Architecture(s) Journal - IxD&A, N.47, 2020-21, pp. 48 - 68

Abstract: Programming is introduced in secondary education in a growing number of subjects. This results in an increasing number of teachers teaching programming in their classes, often without proper training. Learning programming might be complicated, even more so is teaching it. In this context, there is a need to understand teachers' perspectives on teaching programming. This paper aims to identify challenges that teachers in secondary schools face and might negatively impact their teaching, i.e., stumbling stones, as well as elements that promote teaching and give motivation, i.e., stepping stones. The paper is based on the analysis of reflection notes delivered by in-service teachers attending a university-level course on teaching programming. The teachers compile the reflection notes after they complete their final project. Projects are centred around the definition of teaching plans to be tried out in class. The reflection notes of 173 students are analysed to identify issues related to: programming; teaching programming; recurrent didactic issues; and external challenges. The analysis is then summarised in a set of stumbling and stepping stones. For

example, time is identified as one of the main stumbling stones by teachers. On the other side, motivation is one of the central stepping stones that we can identify in the data, often connected to the excitement of teaching something that was not previously taught in schools or that teachers perceive as highly relevant for society and the future job market. Implications for teacher training are also identified.

A description of main findings:

This study aimed to understand the experiences of teachers learning to teach programming through an analysis of their reflection notes. The goal was to identify elements that both promoted and challenged their teaching efforts. Upon analysis, several recurring themes emerged that could be categorized as either obstacles (stumbling stones) or facilitators (stepping stones).

One of the most prominent obstacles was a lack of time. Teachers expressed insufficient time to develop their own programming skills, plan engaging lessons, and provide ongoing formative assessment and support to students. The intrinsic complexity of programming also posed challenges, as did adapting lessons to students with varying skill levels. The disruption of Covid-19 further exacerbated time constraints and ability to implement plans as intended. Integrating programming across disciplines also introduced difficulties aligning content to standards and student understanding in different subject areas.

However, several facilitators emerged that promoted teaching efforts. A primary factor among these was motivation. Teachers were excited about teaching new and relevant content they saw benefiting students' futures. Collaboration also strongly supported learning and planning. Working with peers provided invaluable feedback and shared resources. Being able to customize projects through flexible learning pathways further motivated teachers. Involvement in collaborative communities aided skills development and lesson planning. Autonomy over their learning trajectory seemed to boost self-efficacy and engagement.

This study highlights practical considerations that can enable or inhibit computing education integration. With awareness of known obstacles and support for facilitators like motivation and collaboration, more teachers may be empowered to bring programming into diverse classrooms.

The relation to research questions: This paper is related to RQ4.2, where we aim to identify challenges that secondary school teachers face that might negatively impact their teaching (stumbling stones) and elements that promote education and motivate them (stepping stones).

4.5 Paper V (P5)

Title: In-Service Teachers' Attitude Towards Programming for All

Authors: Majid Rouhani, Victor Jørgensen

Authors' contribution: Rouhani had the primary responsibility for the idea, writing, and presenting the findings at the conference.

Published: In Ludic, Co-design and Tools Supporting Smart Learning Ecosystems and Smart Education (pp. 149-162). Springer, Singapore.

Abstract: Coding instruction has increased widely throughout primary and secondary education in many countries, and educators are just beginning to understand the complexities of teaching students to code. A question raised is whether everyone should learn to code to be fully literate participants in our future society? What are the teachers' main arguments concerning the concept of programming for all? We investigate these questions from a teacher's perspective and aim to determine what attitudes teachers have towards programming for all. We gave teachers a task to describe their thoughts and perceptions on programming for all and collected data during a programming course for in-service teachers. This paper reports on preliminary findings. Although the vast majority in this study have positive attitudes towards programming, we can also see negative attitudes. These concerns are mainly related to lack of time. In-service teachers in this study believe that programming can be fun and engaging. They come with many arguments on reasons why they should include programming in school.

A description of main findings: Teachers generally view programming positively, recognizing its role in enhancing understanding across subjects, developing important cognitive skills, and preparing students for a job market that values programming expertise. Programming is also seen as a tool to engage and inspire students, fostering enjoyment of and passion for learning. However, some educators express concerns about the feasibility of integrating programming into the current curriculum, noting time constraints and the risk of neglecting other subjects. Questions about the relevance of programming for non-technical career paths and the costs associated with teacher training and equipment are also raised. Despite initial reservations, teachers tend to develop a more favorable opinion of programming education after participating in relevant courses. They show interest in using programming to complement their teaching in other areas. Nonetheless, the potential over-extension of the curriculum remains a worry.

The research underscores the importance of computational literacy and equitable access to digital education. It suggests that with increased confidence and proper motivation, teachers can effectively teach programming, which has a positive impact on student outcomes. In summary, the research advocates for programming as a vital component of modern education, with the potential to enhance interdisciplinary learning and equip students with key digital skills. While challenges exist, they may be addressed through targeted professional development and curriculum planning. The ongoing debate highlights programming's significant role in preparing students for the digital future.

The relation to research questions: This study is related to RQ3.1, where we investigate teachers' perspectives on programming for all. Should everyone learn to code to be fully literate participants in our future society?

4.6 Paper VI (P6)

Title: Project-based learning and training of in-service teachers in programming: Projects as a bridge between training and practice

Authors: Majid Rouhani, Monica Divitini, Atle Olsø

Authors' contribution: Rouhani had the primary responsibility for writing the paper and presenting the results at the conference. Initial structure of the paper was defined by Monica. Section V restructured later by Rouhani. ABSTRACT: Initial draft was written by Divitini and updated later by Rouhani. CONCLUSIONS AND FUTURE WORK: initial draft written by Monica and updated by Rouhani. All other sections written by Rouhani.

Published: In 2021 IEEE Global Engineering Education Conference (EDUCON) (pp. 262-271). IEEE.

> *Abstract*: This paper presents and evaluates a model of project-based learning (PjBL) to train in-service teachers of programming. The paper contributes to the body of knowledge around in-service teachers' training to meet the growing demand for teachers being prepared to teach programming in different subjects and school levels. The aim is to design a course that prepares teachers to teach programming, is relevant, and increases their self-efficacy. To contribute to this, we suggest a PjBL approach where teachers create a teaching plan during a project. Teachers may bring in their disciplinary knowledge and pedagogy combined with traditional and instructional methods to design an artifact that can be used directly in their practice. In the proposed model, teachers' projects act as a bridge between training and practice. We implemented the first version of the model in 2019 and used this experience to revise the model described in this paper. The results show that teachers are mainly satisfied with this form of PjBL. Many are favorable to this approach and express joy in teaching programming as they feel prepared and, at the same time, understand that programming can be demanding to learn and teach. Therefore, they are also mentally prepared for what is to come.

A description of main findings: The study presented a novel approach to organizing project-based learning for in-service programming teacher training through the "bridge model". This model aimed to make the training relevant, increase competencies, and boost self-efficacy among participants. By implementing real teaching plan projects, teachers could directly apply their new skills in a meaningful way. Quantitative survey data revealed high levels of satisfaction across different aspects of the course. Qualitative reflections from teachers corroborated this positive reception and perceived value. A key finding was that the model appeared to successfully prepare teachers to feel more confident and ready to teach programming. Having a concrete teaching plan project to complete seemed to give educators tangible skills and resources to draw on in their future practice.

Analysis of the distinct phases of the bridge model yielded further useful insights. In the specialization phase, the flexible structure allowed teachers to focus on personally relevant topics, and workload was deemed appropriate by most. During realization, problem identification was considered helpful by the vast majority for developing their teaching plans. Many anticipated implementing these plans once schools reopened. While the COVID-19 pandemic disrupted the final rectification stage for testing plans in classrooms, it did spur some collaboration between teachers. The bridge model through project-based learning seemed to significantly contribute to the professional development of these in-service educators. The study provides valuable guidance on integrating programming into diverse educational contexts.

The relation to research questions: This study deals with project-based learning, where the project results in an artefact that becomes the link between training and practice. The research relates to RQ2.3, where we evaluate the program's second iteration.

4.7 Paper VII (P7)

Title: Professional Development for In-Service Teachers of Programming: Evaluation of a University-Level Program

Authors: Majid Rouhani, Miriam Lillebo, Veronica Farshchian, Monica Divitini

Authors' contribution: Rouhani had the primary responsibility for the idea, writing, and presenting the findings at the conference.

Published: In International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (pp. 123-134). Springer, Cham.

Abstract: Professional Development (PD) organizations provide training programs for computer science teachers through teacher PD. Programming as part of a teacher's PD has grown in importance in K-12 education. As a result, an increasing number of teachers, often without sufficient training, are teaching programming in their schools. It's challenging to learn to program, and it's even more challenging to teach it. Teachers must be able to program and teach programming, which is usually done in the context of multiple disciplines. In this study, we take a closer look at a teacher's perspective on teachers' PD in programming. We focus on the PD program offered by our university, composed of two courses over one academic year, and define the following research questions: Which strengths and weaknesses of the PD program provided are suited for in-service teachers when learning to program? Learning programming takes time. How could this long learning process be supported? This is an interview study of sixteen in-service teachers who joined the program. The main findings seem to be that teachers are comfortable with how the program is set up. They prefer flexible courses so that they can adapt the implementation with their regular teaching. They also emphasize the need for a continuous learning process and a community of practice (CoP) to continue developing. The main contribution of this study is the evaluation of the PD program at the university.

A description of main findings: The PD program consisted of two sequential courses, starting with programming fundamentals and progressing to pedagogical methods. The second course's adaptability was appreciated, but some teachers felt overwhelmed, especially those with weaker programming backgrounds or lacking guidance. Opinions on the workload varied, with some finding it excessive. The option for flexible deadlines helped mitigate stress. The use of subject-specific content was a key strength, enhancing engagement. While the flexibility was meant to personalize learning, it had the potential downside of creating uncertainty and stress for teachers who needed more structure or lacked confidence in selecting content independently. Additional support was suggested to improve the program. Teachers expressed a desire for more collaborative opportunities, such as communities of practice, and further learning through courses and seminars. Time constraints due to teaching duties were a significant challenge, indicating a need for schools to better support PD activities. Collaborative learning and hands-on project-based experiences were favored as effective methods for learning programming. Teachers might need to adopt a facilitative approach, learning with their students. Active, collaborative, and personalized strategies are crucial for engaging programming education.

The relation to research questions: The study relates to RQ2.4, where we investigate our PD program to identify its strengths and weaknesses.

4.8 Paper VIII (P8)

Title: Learning to Program: an In-service Teachers' Perspective

Authors: Majid Rouhani, Miriam Lillebo, Veronica Farshchian, Monica Divitini

Authors' contribution: Rouhani had the primary responsibility for the idea, writing, and presenting the findings at the conference.

Published: In 2022 IEEE Global Engineering Education Conference (EDUCON) (pp. 123-132). IEEE.

Abstract: Learning to program is often reported as challenging. Difficulties might be connected to, e.g., acquisition of core concepts like iteration, specific language constructs, and program design. Additionally, mastery of a programming language's constructs does not consequently translate into solving new programming problems. These challenges have to be taken into account in the context of in-service teachers' professional development (PD). In this paper, we address challenges teachers face when learning to program, considering these closely related questions: What do teachers perceive as difficult and how does this impact the perception of challenges their students will face? How does this influence the perception of teaching programming? The paper is based on the analysis of 178 reflection notes delivered by in-service teachers attending a university level course on programming to identify issues related to the research questions. Out of the topics that we have selected from the literature as challenging

> for newcomers, the most difficult ones for teachers seem to be writing code and pseudocode to solve a specific problem. The easiest ones are understanding variable initialization and if statements. Also, teachers express that pupils will struggle with many aspects of programming and point to the complexity and students' misconceptions inherent in several programming concepts such as loops. Our results suggest that it is not necessarily difficult to learn to program if in-service teachers are given paid time to learn.

A description of main findings: The study aimed to gain insights into the challenges teachers face when learning to program, and how this impacts their views on teaching programming. It analyzed reflection notes from 178 in-service teachers who took an introductory programming course. The results challenged the common notion that programming is inherently difficult. Teachers generally did not find it too challenging, as long as they had adequate training and time to learn. However, they did identify increasing levels of complexity - from understanding basic concepts to applying them and developing full programs.

When reflecting on pupils' challenges, teachers expected difficulties at a lower conceptual level compared to their own learning. This indicates teachers may underestimate pupils' capabilities. Common struggles for teachers included connecting concepts, writing code to solve problems, and applying concepts due to lack of experience. Teachers were concerned pupils would struggle with loops, writing code, new ways of thinking, functions/libraries and lack of prior knowledge. They recognized teaching programming brings challenges like relating it to other subjects and varying student motivation levels. Time constraints in schools also worried teachers, who need practice but must fit programming into packed schedules. Most felt confident designing lessons after the course, though some wanted more preparation. Feedback implied avoiding steep learning curves and better linking learning to teaching needs.

Overall, the research helped dispel the myth that programming is too hard to learn. With proper support and time, teachers can gain confidence in both learning and teaching this important subject. Tailored development addressing identified needs may help expand programming education.

The relation to research questions: This study is related to RQ3.2, where we look closely at teachers' challenges of learning to program and how this impacts the perception of challenges their students will face. How does this influence the perception of teaching programming?

4.9 Paper IX (P9)

Title: Computer Science in Schools: A Literature Mapping of Professional Development for In-Service Teachers

Authors: Majid Rouhani, Monica Divitini, Amir Massoud Hashemi

Authors' contribution: Rouhani had the primary responsibility for the idea, writing, and presenting the findings at the conference.

Published: In 2022 IEEE Global Engineering Education Conference (EDUCON) (pp. 164-173). IEEE.

Abstract: During the last few decades, there has been a growing need for Professional Development of in-service teachers in Computer Science. This paper presents a systematic mapping of the research in this area published between 2010 and 2020. The study's goal is to map existing literature, understanding how research is evolving and identifying gaps that can prompt new research. The literature mapping is based on the analysis of 206 articles collected from various online databases, then selected according to defined inclusion/exclusion criteria. The study investigates the type of intervention/study, school level, geographical location, connection to curricula, learning objectives, phase of the research, size of the study, and type of collaboration. From the literature mapping, Professional Development of in-service teachers of Computer Science emerges as a rapidly growing and dynamic research area. However, some threats are connected to fragmentation of the research and the need for more cooperation to increase inclusiveness and international collaboration.

A description of main findings: The literature mapping reveals that the professional development of in-service computer science educators is a fast-expanding and active study field. North America leads in PD research output, with Europe as the second most productive region. Other areas like Asia, the Middle East, and Australia/New Zealand contribute modestly, while Africa remains underrepresented, highlighting regional disparities in CS teacher PD resources and research. PD interventions are mainly connected to STEM and CS/ICT curricula, with some studies exploring interdisciplinary applications, such as incorporating CS into English education. The development of computational thinking is a primary objective of PD interventions, alongside programming knowledge and self-efficacy. Equity in CS education is also emerging as a significant focus.

Most studies center on the design and evaluation of PD interventions, with a considerable number informing design and a smaller portion dedicated to evaluation only, reflecting the importance of both creating and assessing PD programs. Research varies in scale, with many studies involving large cohorts of over 100 teachers and others examining smaller groups, indicating diverse research contexts in CS teacher PD. Collaboration within institutions and across organizations is frequently discussed, underscoring the importance of collaborative efforts in developing effective PD programs.

The relation to research questions: This paper addresses RQ1. It is a systematic literature mapping (2010-2020) of the interventions and the factors influencing the professional development of in-service computer science educators.

Chapter 5

Discussions

This chapter delves into the comprehensive analysis of the research findings by examining each research question in detail. Section 5.1 presents a summary of the literature mapping findings on professional development for computer science teachers, highlighting the key trends and research gaps in this area. Section 5.2 evaluates the design and evolution of the professional development framework, focusing on its effectiveness in addressing factors that impact teacher engagement, motivation, and support for student learning. In Section 5.3, the discussion centers on the challenges identified in learning to program from the perspective of teachers, including their perceptions and attitudes towards this subject. Section 5.4 explores the obstacles and facilitators reported by teachers in teaching programming, shedding light on issues such as self-efficacy. Section 5.5 synthesizes the overall contributions of the research, providing a cohesive understanding of the findings in relation to each research question. In Sections 5.6-5.7, the implications of the findings for online education for practice and theory are discussed, including the proposal of an adapted model of online teacher-led learning. Section 5.8 evaluates the research methodology, validity, credibility and limitations.

5.1 RQ-1: The Evolvement of Research in the Area of In-Service Teachers' PD in CS

Study P9 investigated the research within the PD of in-service teachers. The findings for RQ-1 centered around describing the evolving and expanding yet fragmented nature of the research field, key trends identified, and the need for more collaboration internationally to help connect the different pieces of work being done. The fragmentation and lack of standardization were seen as gaps that future research could help address

After identifying relevant publications using the search parameters listed in P9, we learned throughout the screening process that the area lacks a method for structured abstracts [90]. Structured abstracts can help develop high-quality literature mappings to construct field overviews. This is essential, particularly in light of the relative novelty and quick evolution of research on PD of in-service teachers of CS. This challenge is shared with other computing education research communities [90].

The recent review by Yadav et al. [42] analyzed international models of CS teacher education. It identified key themes such as pre-service vs in-service programs, standalone vs integrated CS courses, and face-to-face vs online delivery formats. While Paper 9 focused specifically on in-service teacher PD interventions, some overlapping themes emerged around fragmented research, lack of standardized structures, and need for more inclusive and international work. Both highlight the evolving and multidisciplinary nature of the field.

The articles we studied reveal a complex and diverse ecosystem in which PD meets numerous demands, ranging from integrating fundamental abilities into several disciplines to developing in-depth CS material and pedagogical expertise. Although this diversity is good, there is a barrier associated with the absence of uniform vocabulary, which may contribute to the field's fragmentation, such as the numerous ways in which nations refer to educational levels [91]. There is no standardized method to define the intervention type. For instance, the word 'workshop' is used for various interventions based on their form and length. Although there is no simple answer, raising community awareness of this issue is necessary. Community efforts to develop agreed terms would also be beneficial.

Despite the tremendous expansion of the area, international collaboration remains low. Few studies attempt to comprehend issues across educational systems by establishing joint research or duplicating treatments in various countries. Multinational studies in the education sector are difficult but necessary for expanding knowledge.

Figure 5.1 shows the development of some patterns emerged from the analysis (CT, equity in CS education, self-efficacy in teachers, and sustainability patterns) during 2010-2020. We can see that research has exploded over time. Between 2010 and 2014, 41 qualifying research papers were published, compared to 165 between 2015 and 2020. Research discussion related to equity-oriented approaches for broadening computing participation increased to 4 in 2018 and 8 in 2020. Other trends are computational thinking and a greater emphasis on teachers' self-efficacy development. From 2018 forward, we have seen a clear trend toward providing teachers with an engaging and long-lasting professional development model.

5.1.1 Contribution #1 (1.4)

The study (P9) contributed (C1) to the field of pre-college CS education by identifying the main directions along which research is evolving and the questions that require further attention. We can summarize the directions as follows: (1) few papers meet the requirements for structure abstracts [90], (2) the discipline is both reach and fragmented, and the absence of standard terminology represents a barrier for the international research community, (3) more inclusive research, e.g., researchers should corporate with schools, and (4) more international cooperation.

Literature mappings aim to link work in a particular discipline in a comprehensible fashion. In this regard, our results may be considered a stepping stone for further study.

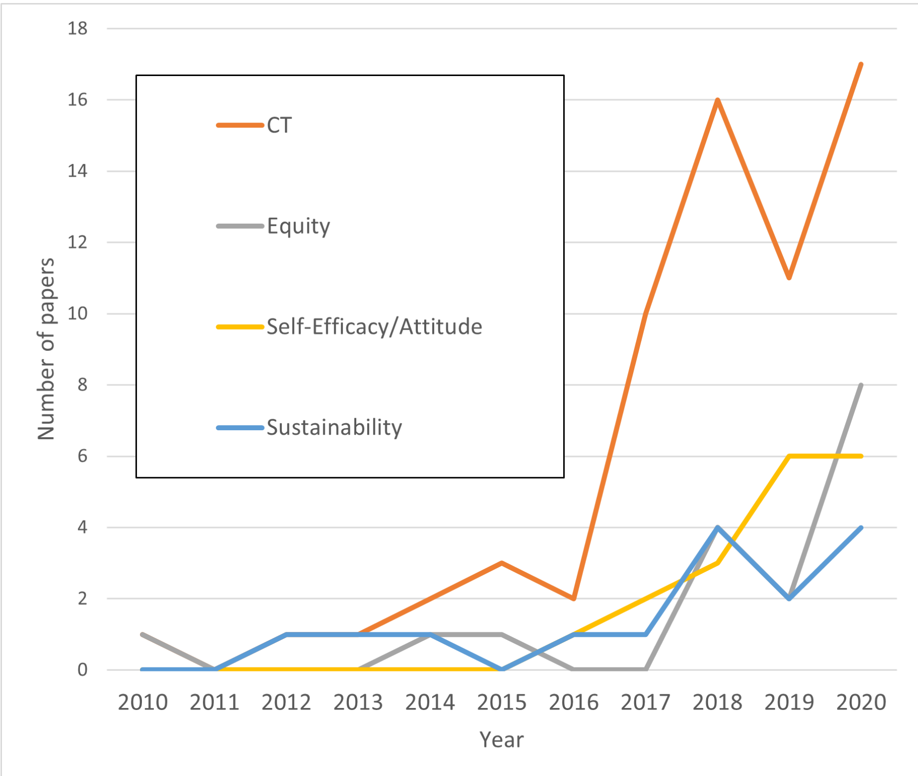

Figure 5.1: Development of some patterns during the period 2010-2020 [26].

5.1.2 Answering RQ-1 (1.3.2)

In the quest to understand the evolving landscape of professional development for in-service computer science (CS) teachers, Section 5.1 of the thesis provides a comprehensive response to RQ-1. The findings shed light on the dynamic trends, emerging research areas, and the evolving nature of international collaboration within this domain.

The research landscape pertaining to professional development for in-service CS teachers has witnessed remarkable expansion, yet it remains characterized by fragmentation. Key thematic trends have surfaced, highlighting the growing significance of computational thinking, equity in CS education, and teachers' self-efficacy. However, the limited scope of international collaboration and the absence of standardized practices underscore the need for concerted efforts to foster cohesion and coherence within this domain.

5.2 RQ-2: Design of the Professional Development Framework

In RQ-2, I examine how to design and assess a professional development program for a diverse group of in-service teachers that promotes engagement and motivation and fosters an active learning environment using project-based learning methodologies. This question has further been divided into four sub-questions (RQ-2.1-RQ-2.4) examined in more detail in papers P1, P3, P6, and P7.

Through analysis of teacher reflections, surveys, and interviews collected over multiple iterations of an online professional development program, several elements were found to positively impact engagement and motivation, including personalized and flexible content delivery tailored to individual needs; incorporation of hands-on, project-based learning opportunities directly applying the content; fostering of collaboration through a communication platform; ensuring ongoing guidance and support throughout; relating the programming content directly to subjects taught; projects bridging training to practice; and continuous evaluation allowing refinement of the framework. These findings provided guidance on designing effective online teacher professional development that promotes active learning and motivation.

An initial program was designed (See Figure 5.2) and evaluated (RQ-2.1) in P1. The program aimed to provide flexibility in time and content for each participant. The diagram depicts two distinct courses, "Course 1: IT6203" and "Course 2: IT6204", each comprising essential components such as "Syllabus", "Webinars", "Assignments", and "Assessment (Grade A-F) Inspera". The "Assignments" section further categorizes tasks as "Mandatory", "Optional", and "Flexible", encompassing associated projects. Both courses utilize the "Communication Platform: Slack" for seamless interaction. In the upper left section of the flowchart, a box illustrates "Textbooks" and "Online resources", directing attention towards "Learner-Centered Design and Computing Education". The flowchart offers a comprehensive overview of the courses' structure, interrelationships between components, and the organization and prioritization of tasks and resources.

The role of a communication platform was investigated in P3 to figure out how it impacts teachers' engagement and motivation in the learning process (RQ-2.2). The in-service teachers' lack of time and the fact that they must include programming in their subjects increase the need to connect the training more closely with the teachers' practice. In P6 (RQ-2.3), a new model called "Projects as a bridge between training and practice" is suggested to enhance the in-service teachers' experience in learning.

In P7 (RQ-2.4), the whole program was evaluated in the third iteration, and the in-service teachers' experience with their PD was investigated. Figure 5.3 shows the main program changes in the last (third) iteration. The diagram depicts the framework of two courses, Course 1: IT6203 and Course 2: IT6204. The syllabus for course 1 (IT6203) includes the textbook "Starting out with Python," an assessment module with "Pass/Fail" results, and four webinars, each lasting one hour. Additionally, a mini-project is linked to the webinars. The assignments are categorized as "Mandatory" and "Optional," with three tasks identified under Mandatory assignments as Task 1, 2, and 3 (though Task 2 is only linked to Task 1). The syllabus for course 2 (IT6204) includes an assessment module with "Pass/Fail" results and eleven two-hour webinars. The assignments are divided into "Flexible" and "Optional" categories, and a project is associated with the assessments. The "Flexible" assignments consist of Task 1, 2, and 3, similar to Course 1. However, Task 2 here is not linked to Task 1 or 3. A teaching plan is linked to the project. Both courses utilize the communication platform "Slack," with multiple channels created for lower secondary, upper secondary, and assignments.

5.2.1 Initial Design of the Programming Course for Teachers Supporting Flexible Learning Trajectories

In RQ-2.1, we wanted to create a training program for in-service teachers that boosts their interest and commitment by making programming applicable to the individual teacher subject matter. In designing and implementing the first iteration of the program, we identified several areas for improvement:

1. Flexibility is an essential factor for teachers and has several dimensions. In-service teachers aim to incorporate programming into their subjects but often have limited time. Online learning [17] can be a suitable learning model since it offers flexibility [92]. Another aspect is flexibility concerning the learning path. Specifically, the design of individualized lesson plans.
2. When the number of participants rises, the course's adaptability does not scale properly. A large number of participants results in a significant workload for the instructors making it challenging to manage [5]. This issue can be resolved to a great extent by modifying the assessment form, which significantly decreases the workload and facilitates the program's scalability [5].

The preliminary results in P1 were encouraging. The participants point out that the program has increased their interest, and they see the relevance of what they have learned [5].

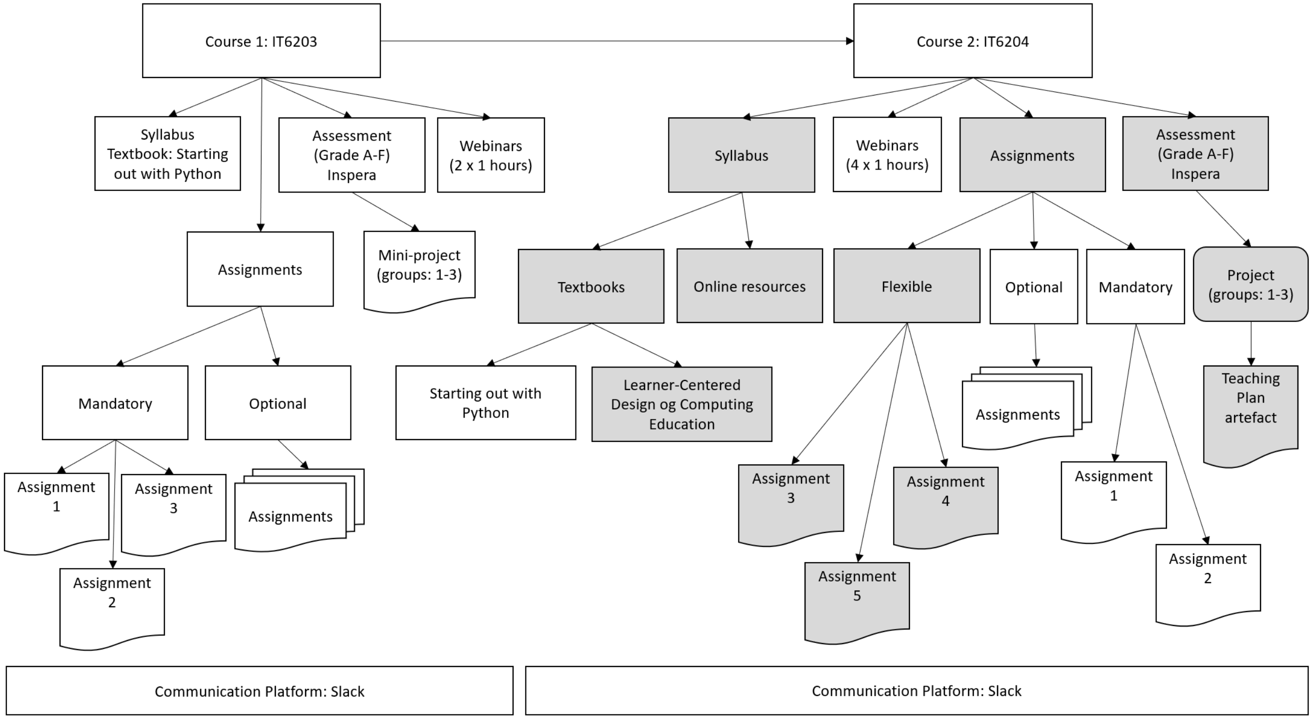

Figure 5.2: Overview of the program structure (iteration 1).

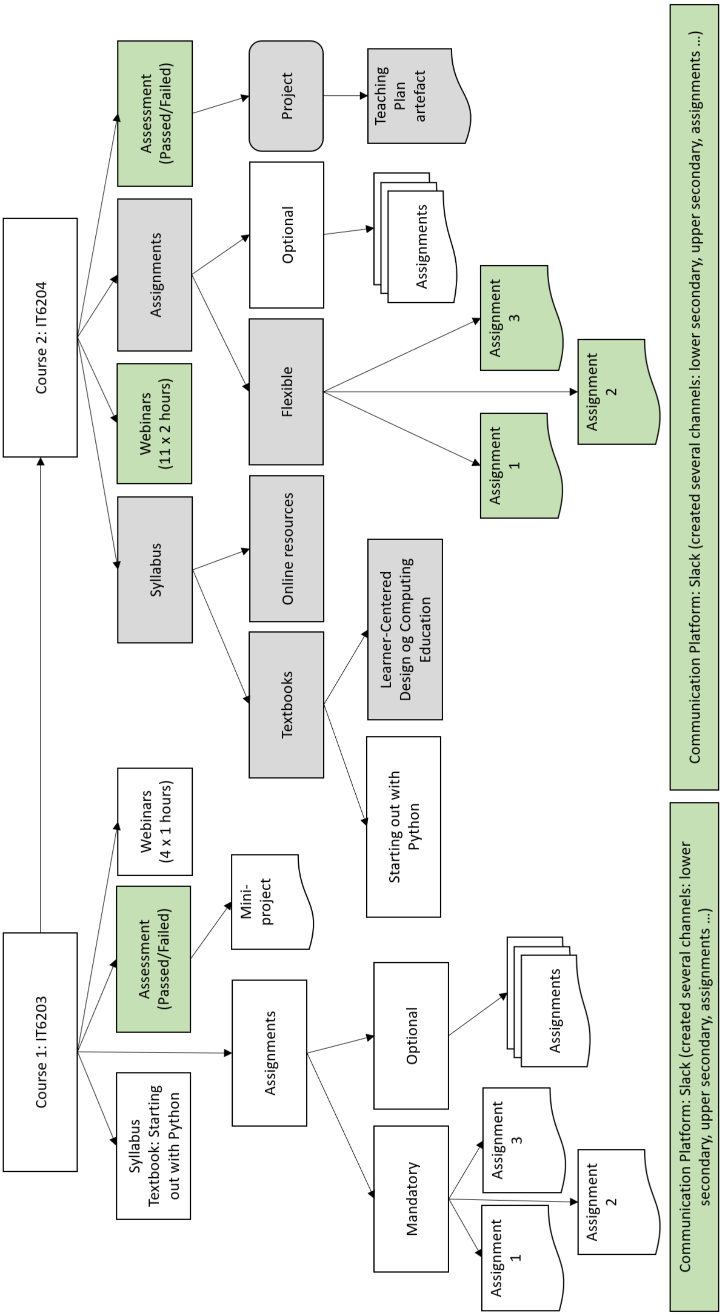

Figure 5.3: Overview of the program structure (iteration 3).

5.2.2 Supporting Students' Learning Processes in a Flexible Learning Trajectory Environment

In RQ-2.2, we wanted to improve collaborative learning in an online environment using a communication platform. The literature [46] demonstrates that greater involvement improves attendance, motivation, and learning. The communication platform Slack [93] was used in this program to influence the active learning process. Slack supports instant chat and channels where users exchange ideas, files, request assistance, and send messages. Our findings show that students collaborate with their peers and instructors via various channels and groups, which is consistent with, Darvishi's [93] conclusions.

Students express their satisfaction, especially concerning communication between instructors, TAs, and students (in-service teachers). The majority of participants reported their satisfaction with the communication platform since they got prompt answers to their inquiries.

One of the tenets of collaborative learning [94] states that group work leads to better comprehension than would likely be the case if an individual worked alone. Our survey data analysis implies that cooperation among classmates, TAs and instructors has enhanced the learning outcomes. Slack is among the numerous communication platform that can lead to higher course engagement and learning. In a training program, there is a need for timely and prompt responses to inquiries [95].

5.2.3 Projects as a Bridge Between Training and Practice

In RQ-2.3, we evaluated the use of project-based learning (PjBL) to create lesson plans that are suitable for direct use in classrooms. In this context, a lesson plan is an artifact created by in-service teachers during their projects and customized to their pupils.

Evidence suggests that PjBL benefits both educators and students [47]. This study (P6) evaluated how PjBL can improve motivational variables by bridging training and practice. Our results show that when the assignment is to create a personal lesson plan that incorporates the disciplinary knowledge and pedagogy of the teachers, they become eager to put significant effort into the project [2]. Further, the direct use of the project outcome, a lesson plan, is an extra motivational factor for the teachers.

5.2.4 Evaluation of the Program

In RQ-2.4, we aimed to highlight the advantages and shortcomings of the PD program proposed in P7. In the first part of our PD program, we teach essential programming techniques [82]. In the second part, we focus on learning how to teach programming [83]. However, some teachers' feedback suggests the need to connect programming to their subject matter right from the beginning [27].

Teachers report their satisfaction with the program's adaptability [27]. The participants were encouraged to choose their learning routes due to:

1. On a personal level, participants' previous knowledge, attitudes, and motivation vary.

2. On a content level, they teach various topics.
3. In a school setting, leadership, and technical infrastructure are not uniform.

Establishing a community of practice, collaboration, and growth via participation in courses and seminars occupies teachers considerably [27]. They underline the significance of collaborating with other educators to exchange competencies and inspire one another. Continuous growth through collaboration is a crucial aspect that must be fostered at several levels: between instructors, teachers, students, and school management. In their reflection notes, teachers participating in our PD program emphasize the need for further development via extra courses, short-term seminars, communities of practice, school-based collaboration, etc. Therefore, they see the PD as the beginning of a lifelong learning process in programming.

Due to changes in the national curriculum, many schools may see PD as a one-time activity that does not involve long-term commitment [27]. As in-service teachers already follow their students through the same school calendar, PD programs that coincide with the school calendar might present practical difficulties for them (e.g., exam overlap and hectic school start). This underlines the necessity of a flexible professional development where teachers can have greater control over the disposition of their time.

Considering the challenges of learning and teaching programming, as well as the lack of time, we claim that the function of the educator may no longer be the same [27]. Teachers should not expect to be thoroughly trained before they start teaching programming but rather function as a facilitator [44]. Many teachers highlight that learning by doing should be an inherent part of the teaching process. Further, many believe they should not be concerned with the lack of sufficient knowledge in the topic, but rather that they will learn and develop with the pupils.

5.2.5 Contribution #2 (1.4)

Contribution two is the recommendation and evaluation of a professional development framework for in-service teachers of CS that provides flexibility in meeting the learning objectives in our PD program. Introducing flexibility in learning paths in a fully online setting calls for an active learning environment where instructors, teacher assistants, and learners are actively involved [40, 68, 96]. For example, learners may participate in debates, activities, brainstorming sessions, simulations, quizzes, or tests. Adopting a communication- and collaboration tool such as Slack [93] can impact this process [79]. Other channels such as e-mail, telephone, and learning platforms should be part of communication to strengthen social presence. As the participants often have little available time, provision should be made for immediate reaction to inquiries. Our findings show that the teachers appreciate instantaneous feedback, which helps create a higher activity level.

5.2.6 Answering RQ-2 (1.3.3)

Elements like flexible personalized content, project-based learning, collaboration opportunities, and ongoing support were found to engage teachers and motivate learning through an analysis of the online PD program iterations. These provided insights to effectively address

RQ-2. The elements identified in the answer to RQ-2 demonstrate the importance of tailoring content, providing flexibility, incorporating hands-on learning, fostering collaboration, and offering ongoing support in promoting active learning and motivation in an online PD program. By considering these elements, future PD programs can be designed to better meet the needs of their participants and ensure a more impactful learning experience.

5.3 RQ-3: The Challenges of Learning to Program

RQ-3 aimed to understand teachers' attitudes towards and perceptions of challenges in learning programming, with RQ-3.1 and RQ-3.2 providing deeper insight. RQ-3.1 explored whether teachers felt coding skills should be universally required to participate in society, revealing their views on the importance and necessity of programming education. Meanwhile, RQ-3.2 sought to understand what difficulties teachers anticipated their own students facing based on their personal experiences learning to code, shedding light on how their struggles shaped expectations for learners. By investigating teachers' perspectives on both the debate around coding for all as well as how their challenges influenced student challenge predictions, these sub-questions contributed valuable context to fully comprehend teachers' outlooks on programming - from attitudes and beliefs to how their journey informed empathy for novice learners.

Analysis of teacher reflections revealed that initial attitudes towards learning programming were often negative due to preconceptions of difficulty but grew more positive as experience and success increased with supportive guidance. Key challenges identified included variables, data types, conditionals, loops, and abstract thinking skills. Teachers' own struggles influenced how they viewed challenges students may face, impacting confidence, but with proper time and addressing knowledge gaps, learning programming became less daunting. Teachers developed a deeper understanding of novice difficulties and how to better support them, highlighting the importance of identifying specific challenges and providing ongoing assistance to build self-efficacy in both learning and teaching programming. Next Sections 5.3.1 and 5.3.2 reflects on the findings in each sub-question.

5.3.1 In-Service Teachers' Attitude Towards Learning to Program

In RQ-3.1, we wanted to determine teachers' attitudes toward programming for all. We examined the study topic from an educator's viewpoint and sought to ascertain their perspectives. Teachers appreciate that computer literacy is vital in the future as people will often have to solve computational reasoning problems [80].

Teachers' ability to think abstractly is another significant concern in this research (P5). Abstract thinking is a crucial component of computational thinking. It is defined as a collection of problem-solving techniques related to computer concepts [97] and has been presented as a multidisciplinary collection of cognitive skills, including interpersonal- and performance-related abilities [98]. Numerous in-service educators concede that this is vital for fostering higher-order thinking and building problem-solving abilities in trans-disciplinary domains [99, 100].

Many in-service educators report (P5) that programming abilities will be one of the essential future career skills. Teaching programming in schools will raise pupils' interest and encourage them to seek programming-related fields of study. However, learning to program is time-consuming [21]; therefore, pupils must start the process significantly earlier.

Teachers argue that learning to program can be fun and inspire pupils in disciplines such as robotics and electrical control. This attitude is essential and can support increased learning outcomes for pupils. In the literature, we find that positive motivational attitudes are necessary and are linked to students' persistence in achieving good results [101].

The findings of this research (P5) demonstrate that educators feel programming can be engaging and enjoyable [102] despite its difficulty to learn and teach [20, 103]. Although the majority of individuals have good views regarding programming, negative attitudes are also reported. Time constraints are the primary cause of these concerns. Humble and Mozelius [51] report on this problem and examine the challenges and potential of incorporating programming in K-12 settings.

5.3.2 Teachers' Perception of Learning Difficulties in Programming

Based on in-service teachers' perception of difficulties in learning to program and what they think pupils would struggle with, we wanted to determine how this influences their teaching pedagogy (P8). In RQ-3.2, we aimed to identify the most significant learning problems within the professional development framework developed for in-service teachers.

Programming is not intrinsically complicated to learn, as suggested by Luxton-Reilly [104]. Instead, challenges may be related to how the topic is taught. The core of the problem may be setting unrealistic expectations for students because the educators are attempting to cover too many topics in beginner courses [105]. Our findings validate this assertion and demonstrate that learning to program is not inherently troublesome if teachers are given sufficient instruction and learning time. This does not negate the difficulty of mastering some topics. Specifically, the data collection demonstrates three degrees of increasing complexity [89, p. 131]:

1. Understanding basic concepts.
2. Understanding how to apply basic concepts.
3. Developing working programs to solve specific problems.

Teachers often stress issues related to applications of the basic concepts rather than understanding them. It is also essential to recognize that teachers believe difficulties stem from how programming is presented rather than the subject itself.

We analyzed teachers' reflections and compared what they experienced as problematic to learn with what they thought their pupils would struggle with. The disparities are concerning since they may indicate a misunderstanding of pupils' real issues on the part of the teachers. Teachers anticipate that their pupils may encounter challenges at a lower level, such as conceptual understanding. We noted that enough practice is one of the most often debated learning problems among teachers. However, few of them discuss it in relation to pupils. Teachers also discuss time-related issues, especially concerning competing requirements from other subjects. If we accept the teachers' perspective that students struggle with basic concepts, then it is expected that students need at least as much time as teachers to practice. This is concerning, although it should be noted that many educators consider this problem when discussing instruction. Here we note that it would be beneficial to conduct future studies where children who have been instructed in programming are interviewed and examined to further our understanding of the disparity between teachers' perceptions and reality.

5.3.3 Contribution #3 (1.4)

Our findings indicate that programming is not inherently difficult to master if teachers are provided with enough training and learning time, as in our case. However, this does not imply that certain subjects are not difficult to master. The perspective of in-service teachers towards the inclusion of programming and coding education in school curricula has evolved positively, according to our findings in P5. When reflecting on their pupils' learning, teachers evaluate similar concerns and difficulties when addressing pupils. However, they report [80] that pupils would have problems at lower levels, such as conceptual understanding.

5.3.4 Answering RQ-3 (1.3.4)

Paper P5's findings shed light on teachers' generally positive attitudes towards programming. The majority of educators recognized the value of programming and its potential to engage and captivate students. However, a notable concern emerged in the form of time constraints, highlighting a practical barrier that educators face when attempting to integrate programming into their teaching. The insights from Paper P8 revealed that teachers identified three progressive levels of difficulty in learning programming: understanding concepts, applying concepts, and developing full programs, underscoring the complexity of learning programming and the need for targeted support at each stage.

Another finding from the research is that teachers often underestimated the challenges students face in learning programming, suggesting a potential gap in understanding the depth of students' struggles and emphasizing the need for a more comprehensive grasp of student difficulties. The synthesis of these findings paints a nuanced picture of teachers' perceptions of programming, acknowledging its inherent challenges and the necessity for a more accurate understanding of the obstacles that learners encounter.

The implications of these findings are multifaceted. The recognition of time constraints as a significant concern warrants attention from educational policymakers and curriculum developers, pivotal in facilitating the effective integration of programming into educational settings. Additionally, the identified discrepancies between teachers' perceptions and students' actual challenges underscore the need for targeted professional development and support for educators. The insights gleaned from the aforementioned papers provide valuable understanding of teachers' attitudes, challenges, and perceptions regarding programming in education, allowing educators and educational stakeholders to work towards a more informed and supportive approach to integrating programming into the educational landscape

5.4 RQ-4: The Challenges and Opportunities of Teaching Programming

In RQ-4.1 and RQ-4.2, we study teachers' perspectives on teaching programming and their view of challenges and opportunities. First, we investigated the long-term impact of their programming teaching self-efficacy in P2. When teachers were interviewed a year after completing the training program, many reported relatively low programming skills, but, at the same time, they conveyed to have learned enough to teach the subject. Thus, they had relatively high self-efficacy, meaning they do not need to be excellent programmers to instruct pupils in programming. Therefore, the teachers seem to gain sufficient programming knowledge over two semesters (as in our case).

We analyzed teachers' reflection notes to explore what they experience as challenging and what opportunities they think programming gives them when integrating it into their subject areas (P4). Some essential obstacles teachers report concerning teaching programming are lack of time, application of programming in a multidisciplinary context, and varying levels of knowledge and motivation among educators. Many teachers recognize the necessity of tailoring teaching to each learner. However, they fear that customizing training can be challenging given their level of knowledge in programming [77].

Teachers perceive motivation as one of the primary building blocks toward a successful experience with teaching programming. In the reflection notes, motivation is often associated with the joy of teaching something that they have not taught before. Further, teachers feel that their contribution to society and the future job market, signified by learning programming to teach it, is rewarding. This desire is also mirrored in how educators perceive their projects, as they aim to provide engaging, creative, and inclusive activities.

5.4.1 In-Service Teacher Training and Self-Efficacy

In RQ-4.1, we aimed to examine the effect of the training program for in-service teachers, emphasizing the teachers' self-efficacy in instructing programming. A teacher's self-efficacy refers to their ability to teach the subject so that students achieve the required learning objectives [78]. Positive self-efficacy is associated with improved student and teacher performance and profoundly impacts teachers' psychological well-being [52]. Measuring teachers' self-efficacy [78] is crucial for further developing a professional development framework.

The research findings (P2) reveal that the provided training benefited teachers' self-efficacy. Several participants reported that the program improved their perspectives on learning and instructing programming. These reports are promising compared to past research in which teachers showed poor self-efficacy in teaching programming [39, 53, 106].

Some teachers report negative mindsets toward the inclusion of programming in school [78]. It is, therefore, useful to consider the inclusion of attitude-building activities that can be created through collaboration and participation. Further, teachers see cooperation among themselves through sharing ideas and discussions as important in maintaining and enhancing their programming skills. We note that many countries have national societies to support teachers, such as the CAS network in the UK [107]. From this viewpoint, it is essential throughout the training to foster a feeling of community among the teachers who attend the lectures and maintain this community after the course has concluded.

5.4.2 Challenges and Opportunities of Teaching Programming

Based on the analysis of teachers' reflections in study P4, we examined teachers' perspectives on stumbling- and stepping stones when preparing for teaching (RQ-4.2). Figure 5.4 provides a visualization of the main stumbling and stepping stones identified in the study. On the left side are the stumbling stones, each with a brief explanation. For example, one stumbling stone mentioned was a lack of time, which can negatively impact teachers' motivation when planning lessons and activities. The stepping stones are shown on the right side. Some stumbling stones that were transformed into stepping stones through various supports are highlighted in blue boxes. For instance, while the COVID-19 pandemic presented significant challenges, some teachers saw it as an opportunity to discover new tools or forms of collaboration. Similarly, the flexibility afforded in customizing lesson plans was overwhelming for some but motivating for others who found relevance in their subject areas. This visualization aims to depict how stumbling stones can be addressed and turned into stepping stones with the right supports. It also acknowledges that experiences are individual and context-dependent by showing that what hinders one teacher's practice may enable another's. The figure provides a high-level summary of the interplay between challenges and enablers identified through the teachers' reflections.

For teachers, learning to program and instructing others go hand in hand. Therefore, teachers must be made aware throughout training that their skills must be updated continuously, even for seasoned educators [108]. Many teachers see the advantage of the availability of online resources. However, for some teachers, it is overwhelming [77]. Teacher training might address this problem by giving specific knowledge about credible resources and how to locate and modify them. This problem may be seen within the context of collaboration, and community development [109]. Teachers' training should emphasize collaboration and community participation, finding successful forms of cooperation in various contexts [46]. According to our study, collaboration, and community participation are additional catalysts for many educators. They involve collaboration with colleagues and pupils and using resources from other communities.

Training should create a culture of sharing and cooperation in which teachers are not just consumers but also creators of learning materials and skills, as well as an awareness of local and international communities. This is vital because it may inspire educators and help maintain communities. Despite the acknowledged advantages of collaborative learning [109], it is essential to note that many educators and students avoid it because of the time and effort it requires.

5.4.3 Contribution #4 (1.4)

The findings in P2 show that the proposed professional development framework has enhanced the self-efficacy of teachers and that the effect is long-lasting. Although several teach-

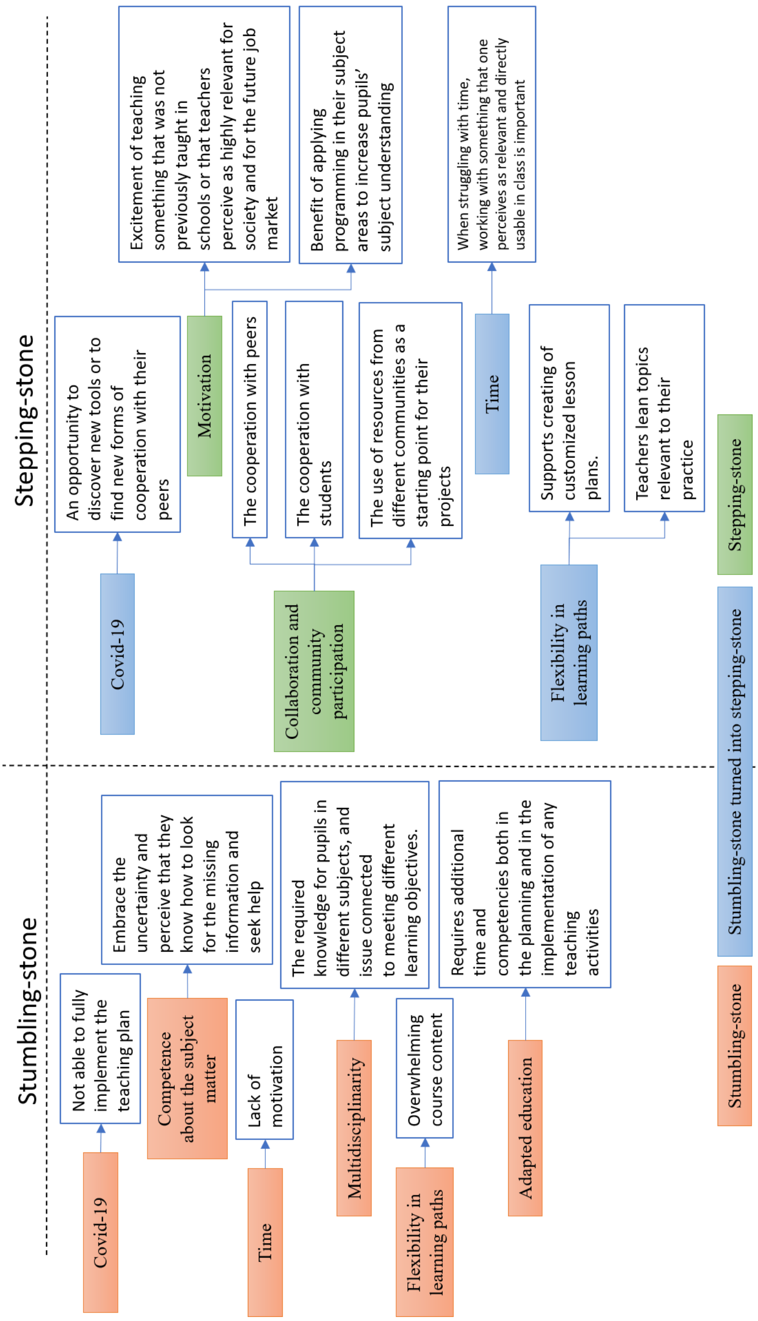

Figure 5.4: Main stumbling- and stepping stones when teaching programming from a teacher's perspective [77]

ers indicated anxiety over their degree of programming expertise, this did not always correlate with low self-efficacy.

Paper four (P4) aimed to identify the elements that promote teachers' positive learning and the challenges they face that might negatively impact their future role as programming instructors. Although some of these elements have been recognized in prior research, we specifically examined how obstacles and enablers might impact policy and teacher education, which has been missing in prior research. Our approach also emphasizes the need to identify teachers individually and collectively within a complex environment that affects their operations. The novelty is in taking a more holistic, ecosystem-based perspective to analyze how various interrelated factors collectively shape teachers' experiences, rather than discrete lists of challenges or recommendations. This provides a more nuanced understanding to inform more systemic solutions through both policy and teacher development. The focus is on the interplay between stumbling and stepping stones for teachers.

5.4.4 Answering RQ-4 (1.3.5)

Paper P2 underscored the profound impact of a training program on teachers' self-efficacy in teaching programming. The training program was found to bolster educators' confidence in imparting programming knowledge to students. Paper P4 highlighted how collaborative and integrative approaches not only fostered effective teaching but also promoted a conducive learning environment for students. The need for flexible and adaptable teaching approaches to accommodate varying student abilities was another finding that underscores the imperative for educators to tailor their teaching methods to suit the diverse learning needs and capabilities of their students, thereby fostering an inclusive and supportive learning environment.

The link between motivation and factors such as engaging project-based activities underscores the need to infuse programming education with elements that inspire students, thereby fostering a sustained interest and enthusiasm for the subject. The research underscored the role of ongoing professional development and support in addressing challenges and maintaining educators' skills in teaching programming. This finding emphasizes the need for continuous learning and growth among educators to effectively navigate the evolving landscape of programming education. Tailored training programs and targeted support mechanisms were found to be instrumental in addressing challenges and bolstering educators' confidence in teaching programming.

5.5 Answering the Main Research Question

This study aimed to establish a professional development framework that enables in-service teachers to develop digital skills and competencies linked to learning and teaching programming. In particular, we examined what challenges and opportunities programming skills provide from a teacher's perspective (see main research question defined in Section 1.3.1).

Darling-Hammond *et al.* [29] outlines seven criteria of successful teacher professional development that contribute to improvements in teacher behavior and enhanced student learning outcomes. These criteria were used retrospectively to analyze the proposed professional development framework and its alignment with recommended best practices, rather than prospectively guiding the design of the course interventions. The criteria are:

1. **Focus on specific content**: The recommended training program stresses that the material must be relevant to each participant. *This is made possible by the framework's flexibility in terms of content. In addition to emphasizing how to teach programming, the program utilizes teachers' educational expertise to design programs that may be applied directly in their practice, increasing their programming teaching self-efficacy [2, 5, 95].*
2. **Incorporation of active learning**: The design of professional development activities must consider how and what educators learn. A form of active learning is when educators participate in the same learning activities they plan for their pupils. *Teachers develop teaching plans through the training program, and during this process, they collaborate with other educators to create and implement activities. The construction of communication platforms and synchronous online teaching techniques to facilitate conversation and close touch with teachers and other participants is vital to achieving interactivity. In addition, the participants are expected to complete activities relevant to their competence areas. This increases engagement, and the relevance of programming in their instruction [2, 79].*
3. **Support for collaboration**: A feature of well-designed PD is that it provides for collaboration between teachers. *In the training program, a communication platform is built to allow teachers to exchange ideas and cooperate on learning. They may collaborate with other educators with similar backgrounds and interests to create teaching plans artifacts by exchanging ideas and making contacts for further cooperation [2, 79].*
4. **Use of effective practice models**: Participants get access to practical lesson plans from peers who participated in previous training programs. *Compendiums, including task collections and suggested answers for different grade levels and topic areas, are made accessible so that teachers have access to challenges that are relevant or related [5].*
5. **Providing coaching and expert support**: Successful adoption of new curricula, techniques, and methods can be achieved by providing teachers with guidance or other forms of professional facilitation [29]. *In this paradigm, project-based learning is vital since it fosters collaboration and active learning. It is intended to use the in-service teachers' pedagogical and content knowledge and the PD instructors' and teaching assistants' competence in the programming domain to aid participants in developing their individualized lesson plans [2, 79].*
6. **Offering feedback and facilitating reflection**: Reflections and feedback are crucial for recognizing needs and encouraging learning according to the individual's needs. *The participants must be able to provide feedback and be encouraged to reflect on their learning and growth regularly [2, 79].*
7. **Sustained duration**: Time and resources are required for quality training, which should be an ongoing process. Programming is time-consuming to learn, and it

must be regularly maintained [27, 42]. The school management should consistently assist the teachers in this endeavor. *We establish a strong connection between training and practice in the bridge model of programming for in-service teachers [2], where they construct their lesson plans and use them in their classrooms. Teachers must continually maintain this artifact (lesson plan) by studying, practicing, applying, and reflecting on new ways that promote practice adjustments.*

The professional development framework for in-service teachers shown in Figure 5.5 has a holistic approach and intends to provide sustainable training in learning and teaching programming (Location of the boxes are not significant). The program consists of the following main elements:

The bridge model [2]. It is the core of the framework and a pedagogical model for connecting the training to practice. It provides flexibility in terms of learning paths in a fully online environment. Participants follow four phases (preparation, specialization, realization, and rectification) to learn how to program and create an artifact (lesson plan) that can be directly used in their practice.

Learning to program. During this study, we identified a set of challenges and opportunities perceived by in-service teachers: (a) programming is not inherently difficult to master, (b) learning to program can be motivating, fun and inspiring, (c) hard for educators to separate learning from teaching programming, (d) three degrees of increasing complexity: understanding basic concepts, applying basic concepts, and developing working programs to solve specific problems.

Teaching programming. We have investigated teachers' perceptions of the difficulties and opportunities of teaching programming. Figure 5.4 depicts the identified stumbling- and stepping stones. Teachers in study P5 report how some stumbling stones can be turned into stepping stones. COVID-19 prevented many teachers from testing their lesson plans in the classroom. However, several teachers report that COVID-19 was an opportunity to discover new tools or find new forms of cooperation with their peers. The program's flexibility implies that teachers can adapt teaching plans to their practice. Some teachers report this as an obstacle since the course content is perceived as overwhelming due to its extensive syllabus.

As part of the framework, a set of guidelines are provided for the implementation purposes:

1. Learning Environment:
   a. Implement an entirely online teaching approach to facilitate greater accessibility for teachers.
   b. Utilize a cost-effective approach by leveraging online methods, considering the diverse geographical locations of participants.
   c. Incorporate both synchronous and asynchronous lectures, with recorded synchronous lectures available on the Learning Management System (LMS).
2. Participants:
   a. Tailor the program for in-service teachers across various grade levels and subject areas.

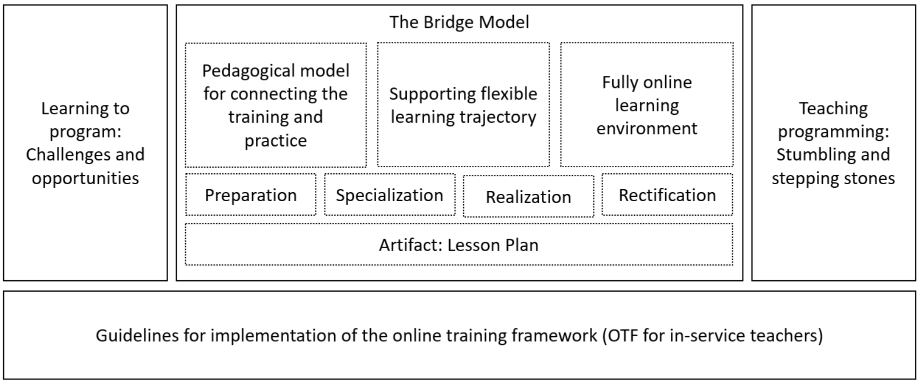

Figure 5.5: The Online Professional Development Framework OPDF for In-service Teachers
.

3. Time:
   a. Allocate dedicated time for participants to learn and incorporate programming into their teaching subjects, recognizing the time-intensive nature of this professional development.
4. Collaborative Learning:
   a. Integrate a communication platform such as Slack, Discord, or a comparable service to foster collaboration among participants.
5. Flexible Learning Trajectories:
   a. Provide a flexible curriculum that allows participants to customize their learning paths.
   b. Ensure the curriculum covers relevant areas where programming applications can be integrated into the school context.
   c. Offer flexibility in the types and scope of activities and exercises to align with each participant's learning path.
6. Learning and Teaching Programming:
   a. Emphasize the simultaneous learning and teaching of programming, recognizing the challenge of separating these two aspects for teachers.
   b. Establish a strong connection between learning programming concepts and their application in teachers' respective disciplines.
7. Project-Based Learning:
   a. Emphasize project-based learning to create an active learning environment and connect training to practical applications.
   b. Encourage the development of customized artifacts tailored to individual teachers' needs to enhance their self-efficacy in teaching programming.

8. Multidisciplinary Approach:
    a. Evaluate teachers' use of pedagogical and subject-matter expertise in building lesson plans through staged projects.
    b. Provide feedback from instructors and teaching assistants to blend pedagogical, disciplinary, and programming abilities in creating lesson plans applicable to teachers' practice.
9. Assessment:
    a. Implement formative assessment throughout the program to support teachers' progress.
    b. Consider a pass/fail evaluation approach to ensure scalability of the program while providing crucial support for participating teachers.

By following these clear guidelines, the implementation of a professional development framework for in-service teachers can be more effectively structured and executed.

This professional development framework attempts to meet the need for high-quality digital competency that is flexible, decentralized, and tailored to the teachers' diverse needs. The framework provides flexibility in terms of the learning paths for teachers and trains them in how to program and teach programming. It supports teachers in creating their 'lesson plan' through a project-based learning environment that can be used as their teaching material for their pupils. Developing such an artifact increases teachers' confidence, self-efficacy, and motivation, which aligns with findings that integrating subject content, collaborative learning and connecting to teachers' individual contexts can enhance engagement and motivation. Additionally, the program is online, which makes it cost-effective and flexible.

We examined the research development (RQ-1) to better understand the trends and examine the gaps and problem areas. Furthermore, we have developed and evaluated the program over several iterations to design an online flexible teachers' professional development program that promotes active learning and motivates teachers (RQ-2). In RQ-3, we investigated teachers' attitudes towards learning to program and what they perceive as problematic, and in RQ-4, we looked at elements that promote teaching programming and increase self-efficacy.

5.6 Implications for practice

The study has several implications for different stakeholders in computer science education as depicted in Figure 5.6. The research highlighted that policymakers need to prioritize long term professional development, school administrators must support teacher communities and dedicate resources, teacher educators should design effective online programs incorporating key elements, and students ultimately benefit as more motivated teachers confidently bring programming into classrooms through active learning.

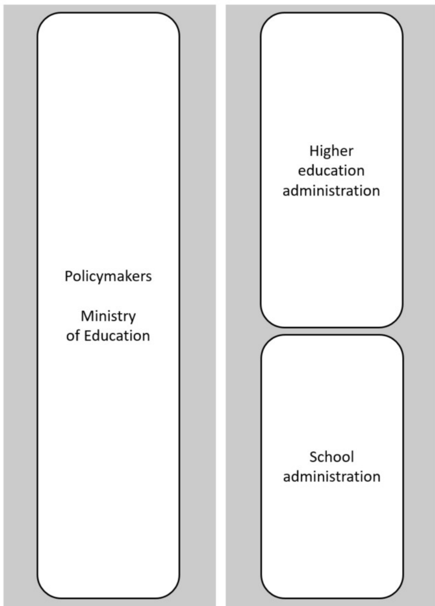

Figure 5.6: Key stakeholders affected by the study

5.6.1 Implications for Policymakers

Sustained professional development programs for in-service teachers in computer science and programming are essential to meet the growing demand for digital literacy and programming skills. By mandating and funding such initiatives, developing flexible programming curricula, protecting time and resources, facilitating online communities of practice, incentivizing training, and fostering partnerships, policymakers and educational institutions can ensure that educators are equipped with the expertise and support necessary to deliver high-quality computer science education. Ultimately, investing in the professional development of teachers is an investment in the future of education and the workforce.

5.6.2 Implications for School Administration

The enhancement of computer science education within K-12 institutions requires a comprehensive approach that encompasses professional development opportunities for educators, the facilitation of collaborative communities, incentivization, infrastructure support, communication, and strong leadership. By investing in these initiatives, schools can ensure that students are equipped with the skills and knowledge necessary to thrive in an increasingly technology-driven world.

Busy schedules were a key barrier for teachers. Administrators must protect dedicated time for self-paced study and collaborative project work during and after school hours. The communication platform fostered peer learning and motivation. Schools should facilitate virtual spaces for ongoing professional sharing, support and networking among programming teachers.

5.6.3 Implications for Higher Education

Developing comprehensive pre-service teacher education programs and flexible, online professional development opportunities for in-service teachers, as well as conducting research, providing credentials, establishing communities of practice, developing open online resources, advocating for policies, forging partnerships, and tracking metrics are all essential components of a successful effort to prepare educators to teach computer science and programming concepts in K-12 schools. By focusing on these areas, we can ensure that educators are well-equipped to provide high-quality instruction in this critical area of education.

5.7 Implications for Online Education Theory

In recent years, the landscape of education has undergone a significant transformation with the widespread adoption of online learning. This shift has not only impacted traditional student education but also the professional development of in-service teachers. The study under discussion has profound implications for online education theory, particularly in the context of training programs for in-service teachers. By adapting the model proposed by Picciano [41] to include 'self-paced/independent study,' the study highlights the need for flexibility in online courses and the impact it has on the learning process for teachers.

The model proposed by Picciano [41] for a fully online course has been adapted to the specific context of in-service teacher training. This adaptation, as depicted in Figure 5.7), incorporates 'self-paced/independent study' to accommodate the unique requirements of teachers. Online teaching, as emphasized, offers the advantage of flexibility in terms of time and place. Moreover, a flexible online course designed for a diverse group of participants necessitates active shaping of their learning process.

The research findings reveal that in-service teachers do not perceive learning basic programming concepts as overly demanding. Consequently, they are capable of engaging in independent study to a greater extent. Furthermore, their existing pedagogical knowledge and skills competency enable them to effectively organize and pursue self-paced/independent study. As a result, the program's flexibility extends beyond time and place to include the element of 'content.' In this context, 'flexibility in content (learning objectives)' empowers participants to decide which topics to study and the depth of complexity they wish to explore.

The inclusion of 'content flexibility' significantly increases the need for self-paced/independent study, thereby influencing the model proposed by Picciano [41]. As illustrated in Figures 2.6 and 5.7, this adaptation underscores the evolving nature of online education and the necessity to accommodate the diverse learning needs of participants. The traditional model of online education may need to be reevaluated to incorporate the growing demand for personalized, self-directed learning experiences.

The study's findings underscore the importance of flexibility in online education, particularly in the context of in-service teacher training. The adaptation of existing models to include 'self-paced/independent study' and 'content flexibility' reflects the evolving nature of online education theory. As the landscape of education continues to evolve, it is imperative to recognize and address the diverse learning needs of participants, thereby ensuring the effectiveness of online training programs for in-service teachers.

5.7.1 Alternative theories for future research

Chapter 2 has delved into various theories, shedding light on their potential impact on online learning models. However, the exploration of alternative sociocultural learning theories could further enrich our understanding of teacher learning activities in an online environment.

Cultural-Historical Activity Theory (CHAT) [110] presents a compelling lens through which to analyze the contextual factors that influence teacher learning activities in an online environment. By examining the interplay between tools, rules, community, and division of labor, this theory offers a comprehensive framework for understanding the complexities of online teaching. Through the application of CHAT, researchers can gain deeper insights into the dynamics of online teacher learning, thereby enhancing the design and implementation of effective online learning models.

Community of practice (CoP) theory [111] provides a valuable perspective on how online platforms can facilitate the formation of communities where teachers collaborate, share practices, and develop their identities as educators. By harnessing the power of online communities, educators can engage in meaningful interactions, exchange knowledge, and collectively enhance their pedagogical practices. This theory holds significant promise in shedding light

Figure 5.7: Teacher-Led Fully Online - Adapted to flexible learning trajectory program
.

on the role of online platforms in fostering collaborative learning among teachers, ultimately contributing to the advancement of teacher education in digital contexts.

The application of Boundary Crossing theory [112] offers a nuanced understanding of how teachers navigate the boundaries between formal training and classroom practice in online environments. By examining the facilitators and inhibitors of boundary crossing in digital contexts, researchers can identify ways to optimize online learning experiences for teachers. This theory provides a fresh perspective on the challenges and opportunities inherent in the integration of online learning models within the broader landscape of teacher education.

By integrating these sociocultural theories with existing models, researchers can unlock new dimensions of understanding in teacher learning design and experiences in digital contexts, offering a holistic view of the complex interplay between sociocultural factors and online learning models. This integration presents a fertile ground for theoretical advancement, thus paving the way for innovative research in the field.

5.8 Evaluating the Research

Research evaluation is necessary to ensure the data's validity, results, and interpretation. Information about each study's research quality associated with various data-gathering techniques can be found in each publication. This section briefly reflects on the transferability and validity of the research in the project. I provide a quick summary and outline the key threats to the entire project, as well as the mitigation strategies I used.

5.8.1 Credibility and Legitimation

Validating a research study implies that its results and applications may be of high, low, or mediocre quality. In quantitative research, the researcher is concerned with both validity and reliability. Reliability relates to the replication of the study and, thus, the research process's openness [113]. The study design, data collecting, analysis, and interpretation phases of quantitative research face several risks to its validity. Onwuegbuzie and Johnson [114] presents an overview of 50 different internal and external validation threats that might occur at various stages of this process.

The sample population in studies P1, P3, P6, and P8 are teachers who participate in the programming courses. Therefore, examining challenges and opportunities in learning and teaching programming is closely linked to our program. We have pointed to this as a weakness/limitation in the research. However, the number of participants in the project has been significant, that is, between 26% (22 out of 86 in the first cohort) to 96% (186 out of 194 in the third cohort). We tried to get as many people involved as possible, informed about the project and voluntary participation, and outlined the procedures and steps to boost the research's reliability.

In qualitative research, the focus is on validity rather than reliability and seeks to establish whether the results are credible or believable, and the word "credibility" is often used. Evaluating qualitative validity involves determining if the received knowledge is truthful, reasonable, transferrable, reliable, and verifiable. Studies P2 and P7 were interview studies, while P4 and P5 were based on examining teachers' reflection notes. As a step towards increased validity, multiple research assistants conducted interviews, transcriptions, and analyses. During the coding process, the method of "Percent Agreement for Two Raters"[1] was used to calculate the inter-rater reliability. To reduce researcher bias, I was not involved in the interview process.

For the mixed research (P1, P3, P6, and P8), we collected quantitative (closed-ended) and qualitative (open-ended) data. To comprehend research difficulties, we then formed interpretations based on the combined strengths of both data sets. Onwuegbuzie and Johnson [114] argues that we should use the word "legitimation" when discussing mixed methods research and defines nine legitimation types as shown in Table 5.1. Some of the legitimation types most applicable in this study are discussed below.

Sample integration refers to the threat linked to how the sample is selected and the composition of the individuals in the group. The selections in this study are from three different cohorts but from the same group participating in the program at the university. Our samples include diversity in the form of gender and what school level (upper or lower secondary)

[1] https://www.statisticshowto.com/inter-rater-reliability/

Table 5.1: Typology of Mixed Methods Legitimation Types adapted from Onwuegbuzie and Johnson [114, p. 57]

Legitimation type	Description
Sample Integration	The extent to which the relationship between the quantitative and qualitative sampling designs yields quality meta-inferences.
Inside-Outside	The extent to which the researcher accurately presents and appropriately utilizes the insider's view and the observer's views for purposes such as description and explanation.
Weakness Minimization	The extent to which the weakness from one approach is compensated by the strengths from the other approach.
Sequential	The extent to which one has minimized the potential problem wherein the meta-inferences could be affected by reversing the sequence of the quantitative and qualitative phases.
Conversion	The extent to which the quantitizing or qualitizing yields quality meta-inferences.
Paradigmatic mixing	The extent to which the researcher's epistemological, ontological, axiological, methodological, and rhetorical beliefs that underlie the quantitative and qualitative approaches are successfully (a) combined or (b) blended into a usable package.
Commensurability	The extent to which the meta-inferences made reflect a mixed worldview based on the cognitive process of Gestalt switching and integration.
Multiple Validities	The extent to which addressing legitimation of the quantitative and qualitative components of the study result from the use of quantitative, qualitative, and mixed validity types, yielding high quality meta-inferences.
Political	The extent to which the consumers of mixed methods research value the meta-inferences stemming from both the quantitative and qualitative components of a stud

they teach (P2 and P7). Other factors, such as age, teaching experience, etc., are not considered. We have not thought to include teachers who do not participate in our program when investigating the challenges of learning and teaching programming.

Inside-Outside legitimation refers to "the extent to which the researcher accurately presents and appropriately utilizes the insider's view and the observer's views for purposes such as description and explanation" [114, p. 57]. The risk is if the researcher offers both an insider and an outsider perspective. My background in the software development industry and my role as the instructor mean that I am considered an insider. Therefore, I have used external resources or researchers' thoughts to be insiders when conducting the interviews and analysis.

Sequential legitimation relates to the threat associated with the sequencing of the research. The question is how the findings and conclusions would have differed if the quantitative and qualitative research methods had been conducted in reverse order. The research method of this thesis was mixed-methods with a parallel approach, where quantitative and qualitative data are collected and analyzed simultaneously. However, in those studies (P5 and P8) where the questionnaire contains closed and open-ended questions, the closed questions can influence how open-ended questions are answered.

Weakness minimization refers to the amount to which the second approach's strength compensates for the previous approach's weakness. We have used several teacher reflections from multiple cohorts (P3, P4-P8). This strategy balances the risk associated with quantitative and qualitative research.

5.8.2 Generalizability and Transferability

The research method is qualitative and mixed-methods with a parallel approach, where quantitative and qualitative data are gathered and evaluated together. In addition, the findings come from distinct cohorts within the same institution and program. Whether the outcomes are generalizable or transferable to other contexts is up to the reader to determine the context's applicability and effects on their situations [115].

In this study, we have researched in-service teachers who aim to learn to program and teach basic programming to their pupils. Rapid and cost-effective training of many educators is necessary due to the high demand. Therefore, we intend that the findings in this study are transferable to other higher education institutions nationally and internationally.

Our study concerns teachers who plan to include programming as part of other subjects (e.g., STEM). This means that they initially do not have enough time for this training. It is also reasonable to believe that many teachers are not interested in this field since it is not their own discipline. In addition, we know from the literature that learning to program can be challenging [15, 16]. A success factor is, therefore, that teachers who participate in such a program get enough time to improve their skills. Given that the teachers have some free time (as in our case), it is also necessary to focus the training to benefit the individual participant.

Flexibility is, therefore, the other important factor, given that the participants have varied backgrounds and plan to apply programming in different disciplines. However, introducing flexibility concerning the syllabus (each participant can set up their learning path) adds new complexity to the program. Creating an active learning environment according to the principles of the community of inquiry will therefore be essential in this context.

A student-active online learning environment must be founded on the community of inquiry theory (See Section 2.2.2), where social presence is one of three essential elements. Therefore, a communication platform [79] that facilitates and contributes to building such an environment becomes necessary.

Our findings show [79] that if the teachers access an excellent communication platform and perceive their training as relevant, it can promote social presence and create a high level of commitment where they ask questions and discuss topics. It is also important that the instructors and TAs are reflexive and respond quickly to questions that arise. If it takes too long to get clarification, the teachers can quickly give up and withdraw, and the whole thing can quickly become too demanding, and they may give up. The program may be transferable to another context, given that the above recommendations are followed.

5.8.3 Reflexivity and the Role of the Researcher

Factors that impact qualitative research include reflexivity, transferability, interpretation, and analysis. A qualitative researcher is considered part of the research process; thus, previous experiences, assumptions, and beliefs can influence this process. Reflexivity is a type of critical reflection on the position of a researcher and is about identifying preconceptions that the researcher brings to the project [116]. The purpose of being reflexive is to recognize any personal perspectives and researcher biases that may influence the study. Researcher biases such as the way data are collected, choice of collection methods, analysis, and reporting can affect the research.

As a software engineer working in the software industry over many years, I did not have knowledge of educational development work and programming didactics in connection with teaching programming at schools. In 2018, I was employed at NTNU and started this project, and my responsibilities, among others, was pedagogical development work, researching, and teaching. Although I knew how to program, this field was new to me since it is not just about solving different problems through programming, but the pedagogical approach and didactical issues become central.

As a practicing programmer and instructor in the course, I was aware that my views and perspective could influence the interview subject. Therefore, I did not conduct the interviews myself in the interview studies (P2 and P7). We hired research assistants with no programming experience and were not involved in the course (outsider). Research assistants also participated in the transcription and analysis of the data.

In studies P1, P3-P6, and P8, we used teachers' reflection notes as the dataset in the qualitative research. As a researcher, professional software developer, and instructor of the program, it was challenging for me to remain impartial and put aside my own experience. I was aware of these dangers. To minimize the effects of bias throughout the analysis, I needed to maintain objectivity by putting aside my own opinions and emotions and adopting the position and viewpoint of a researcher.

Table 5.2 shows resources and their contribution to data collection and analysis in the project. In paper three (P3), I hired one research assistant to anonymize the Slack data before the analysis phase. This data set and the teachers' feedback through questionnaires

Table 5.2: Overview of resources used in the project

Date	Paper	Type of resources	Role
2019	P1	Researcher 1 (outsider)	Involved in the analysis
2019	P1	Researcher 2 (outsider)	Was not involved in the analysis
2019	P1	Student (outsider)	Was not involved in the analysis
2019	P1	Researcher 3 (outsider)	Was not involved in the analysis
2020	P2	Student (outsider)	Interviewed, transcribed, some analysis
2020	P2	Researcher (outsider)	Was not involved in the analysis
2020	P3	Student (outsider)	Anonymized the Slack data
2021	P4	Researcher (outsider)	Was involved in the analysis
2021	P4	2 research assistants	Were involved in the analysis
2021	P4	Research assistant	Was involved in the analysis
2021	P6	Researcher (outsider)	Was involved in the analysis
2021	P6	Researcher (outsider)	Was involved in the analysis
2021	P7	Researcher (outsider)	Was involved in the analysis
2021	P7	Research assistant	Was involved in the analysis
2022	P5	Research assistant	Was involved in the analysis
2022	P8	Researcher (outsider)	Was involved in the analysis
2022	P8	2 research assistants	Did the interviews and were involved in the analysis

essentially formed the base for the quantitative part of the study. For the qualitative part, open-ended questions were the data set, and I analyzed them myself. To avoid bias, I have tried to be as objective as possible.

5.8.4 Ethical Concerns

The most crucial aspect of research is its ethical content. Researchers must adhere to ethical standards from the beginning to the finish of a study [117]. When doing research, there are several ethical factors to keep in mind, such as voluntary participation, informed consent, confidentiality, and the communication of results.

Throughout the study, we have made it clear to all participants that their participation is entirely voluntary and that they are free to withdraw at any moment. They have been informed about the purpose of the research and which body supports the project financially so that they can choose not to participate if they wish. We have also been clear that all data used in the research is anonymized. We applied NSD[2] to get consent for data collection. In some situations, we analyzed participants' reflection notes which were not anonymized at the beginning. We engaged others outside the research group to anonymize the data in these cases.

[2] https://www.nsd.no/

5.8.5 Reflection on the methodology

The design science methodology employed in this research was well-suited to comprehensively evaluate the teacher professional development program and answer the research questions. A mixed methods approach using both qualitative and quantitative data collection and analysis allowed a holistic understanding of the complex phenomenon under study. The qualitative methods, such as thematic analysis of reflection notes and interviews, provided rich insights into teachers' experiences, perspectives and development over time. Meanwhile, the quantitative surveys captured trends and patterns across larger samples. Integrating both data types through joint displays strengthened the validity of findings. The evaluation was also rigorous, with data collected at multiple time points and systematically analyzed using established processes like coding and theme development.

Overall, the methodology was fit-for-purpose and yielded meaningful conclusions to guide ongoing improvement of the program. However, a limitation of the data collection methods was the reliance on self-reported data (teachers self-reported their experiences, views, challenges etc), which can introduce bias and inaccuracies. The sample selection approach also presented some challenges, as it was difficult to ensure a representative sample of the population of interest.

5.8.6 Limitations

Sections 5.8.2 to 5.8.4 evaluate the research and highlight the key concerns and the mitigation strategies used. Here, I summarize the main restrictions that apply to the project as a whole.

Among the limitations is the sequence of events in the research process, particularly the absence of a cohesive research project at the outset. Rather than being meticulously designed as a research endeavor from the start, this study emerged from the development and implementation of an initial teacher training program. The research commenced with a series of research questions, methodologies, and data collection processes, potentially hindering the establishment of a unified and integrated research design. Additionally, the publication of separate studies or papers at different stages may have led to inconsistencies or gaps, rather than a fully cohesive research design. The iterative nature of the research process may have also constrained the ability of subsequent studies to systematically build on or incorporate insights from earlier ones. Moreover, the occasional lack of clear distinction between program development/implementation and the research components may have affected the ability to draw definitive conclusions about the program's effectiveness. A more comprehensive and coherent research plan established from the outset could have facilitated the derivation of stronger and more definitive conclusions.

All quantitative and qualitative data collected through interviews, questionnaires, and reflection notes are related to participants from the same program, university, and country. The results can be relevant in a comparative context, but generalization might prove to be difficult. Another limitation related to the data collection is that the program (textbook and lessons) may have influenced the participants' attitudes towards programming. In this study, the effort needed to broaden participation in computing education with regards to gender gap [118] and equity in the classroom [119], and teachers' perceptions on how to address these issues remain for further investigations in the context of programming for in-service teachers. Assessment is marginally investigated in this study due to time constraints. Our initial results show that assessment in programming is a harder problem than educators realize. Consequently, teacher education programs must discuss this issue openly and urge teachers to include evaluations in their class plans.

5.9 Summary

This chapter provided a detailed discussion of the findings of the research project and their implications for the field of computer science education. The chapter began by summarizing the main findings of the research project, including the development of a professional development framework for in-service teachers, the challenges and opportunities associated with learning and teaching programming, and the effectiveness of the professional development framework in improving the professional development of computer science educators.

The chapter then went on to discuss the implications of the research findings for the field of computer science education. The development of a professional development framework for in-service teachers is an important step towards improving the quality of computer science education in secondary schools. The importance of project-based learning and active learning strategies in teaching programming, and the need for ongoing professional development opportunities for computer science educators is highlighted.

The chapter concluded with a discussion of the limitations of the research project and suggestions for future research. It is acknowledged that the research project was limited in scope, as it only involved participants from a single country. It is suggested that future research could explore the effectiveness of the professional development framework in other contexts and with larger groups of participants.

Chapter 6

Conclusions and Future Work

6.1 Concluding Remarks

This research aimed to design, develop, and refine a professional development framework that prepares in-service teachers to acquire digital skills and critical competencies needed to teach programming and incorporate it into their subjects.

Based on the qualitative and mixed methods analysis of interviews and teachers' reflection notes, we explored different framework characteristics and examined how flexibility in several dimensions affects the program's design. Other prominent characteristics in focus were collaborative learning in a project-based environment, scalability, self-efficacy in teaching programming, challenges and opportunities from an in-service teacher's perspective, and last but not least, the importance of all stakeholders' involvement. The research resulted in the development of a professional development framework with a sustainable, holistic approach. "sustainable" refers to the ability of the professional development framework to endure and remain effective over time. It implies that the approach taken in the development of the framework considers the long-term impact on the society, and the resources involved. "Holistic" suggests that the framework takes into account the interconnectedness of various factors, aiming for a comprehensive and enduring solution rather than a short-term fix.

The data analysis's scope and the three cycles of the study offers new insight into high-quality and sustainable professional development. The research clearly demonstrates that teachers have a positive attitude towards the inclusion of programming in schools. Specifically, the inclusion of programming in, e.g., STEM subjects is seen as highly advantageous. Yet, several obstacles make the process demanding. Finally, it is important to note that teachers' roles are changing, and teachers' professional development must occur both inside and outside the classroom.

6.2 Future Work

Practice and research in other teaching environments are necessary to confirm the results. Furthermore, more studies remain to be carried out to measure the long-term effects on teachers who teach programming and the pupils' learning outcomes.

An evaluation of how the identified barriers and stepping stones might be applied to a larger group of in-service teachers would be helpful. In future investigations, it could be interesting to examine how the teacher's characteristics, such as age and experience, may affect the outcomes. It could also be interesting to explore another pedagogical problem uncovered by teachers during the execution of their lesson plan: the significance of teamwork via pair programming. Future studies may also need to focus more on the difficulties when parents cannot assist their children with schoolwork due to a lack of programming skills.

There is a need for more research related to the form of assessment both in this program and on how teachers should assess students and their learning outcomes. The decision to concentrate on lecturers' self-assessment and reflection has significance since it provides the learner's unique viewpoint. It may influence how people see themselves as programming developers and educators. Future research must examine the correlation between teachers' self-reported perceptions and their performance as programmers and educators.

Another area we have not focused much on is gender differences. Teachers in our studies have not reflected significantly in this, which is worrying, as we know from the literature that gender differences exist in computing education [118]. Concerning broadening participation in computing education, it is vital to examine problems related to programming that might contribute to more pronounced gender disparities or opportunities that lead to higher engagement for both sexes.

Bibliography

[1] R. M. Felder and R. Brent, 'Active learning: An introduction,' *ASQ higher education brief*, vol. 2, no. 4, pp. 1–5, 2009.

[2] M. Rouhani, M. Divitini and A. Olsø, 'Project-based learning and training of in-service teachers in programming: Projects as a bridge between training and practice,' in *2021 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2021, pp. 262–271.

[3] T. Anderson, R. Liam, D. R. Garrison and W. Archer, 'Assessing teaching presence in a computer conferencing context,' 2001.

[4] S. Carretero, R. Vuorikari and Y. Punie, 'Digcomp 2.1: The digital competence framework for citizens,' 2017.

[5] M. Rouhani, M. Divitini, V. Vujosevic, S. Stai and H. A. Olstad, 'Design of a programming course for teachers supporting flexible learning trajectories,' in *Proceedings of the 8th Computer Science Education Research Conference*, 2019, pp. 33–38.

[6] B. Karseth and K. Sivesind, 'Conceptualising curriculum knowledge within and beyond the national context,' *European journal of education*, vol. 45, no. 1, pp. 103–120, 2010.

[7] A. Bandura, 'Self-efficacy: Toward a unifying theory of behavioral change.,' *Psychological review*, vol. 84, no. 2, p. 191, 1977.

[8] D. Garrison, 'Communities of inquiry in online learning,' *Encyclopedia of Distance Learning*, pp. 352–355, Jan. 2009. DOI: 10.4018/978-1-60566-198-8.ch052.

[9] M. Hilbert, 'Digital technology and social change: The digital transformation of society from a historical perspective,' *Dialogues in clinical neuroscience*, 2022.

[10] Regjeringen.no, *Strategi for desentralisert og fleksibel utdanning ved fagskoler, høyskoler og universiteter*, (Accessed on 11/1/2022), Jun. 2021. [Online]. Available: https://www.regjeringen.no/no/dokumenter/desentralisert-og-fleksibel-utdanning-ved-fagskoler-hoyskoler-og-universiteter/id2861368/.

[11] L. B. Easton, 'A global perspective: What professional learning looks like around the world,' *Journal of Staff Development, v34 n3 p10-13, 14, 16, 18*, 2013.

[12] C. Redecker *et al.*, 'European framework for the digital competence of educators: Digcompedu,' Joint Research Centre (Seville site), Tech. Rep., 2017.

[13] G. E. Karlsen, 'St. meld. nr. 30 (2003–2004): Kultur for læring–et utdanningspolitisk perspektiv,' *Norsk pedagogisk tidsskrift*, vol. 88, no. 4, pp. 331–335, 2004.

[14] M. R. Blanchard, C. E. LePrevost, A. D. Tolin and K. S. Gutierrez, 'Investigating technology-enhanced teacher professional development in rural, high-poverty middle schools,' *Educational researcher*, vol. 45, no. 3, pp. 207–220, 2016.

[15] S. Abesadze and D. Nozadze, 'Make 21st century education: The importance of teaching programming in schools,' *International Journal of Learning and Teaching*, pp. 158–163, 2020.

[16] E. Lahtinen, K. Ala-Mutka and H.-M. Järvinen, 'A study of the difficulties of novice programmers,' *Acm sigcse bulletin*, vol. 37, no. 3, pp. 14–18, 2005.

[17] C. J. Heinrich, J. Darling-Aduana, A. Good and H. Cheng, 'A look inside online educational settings in high school: Promise and pitfalls for improving educational opportunities and outcomes,' *American Educational Research Journal*, vol. 56, no. 6, pp. 2147–2188, 2019.

[18] P. D. Haagensen, 'Supporting in-service teachers of programming in secondary schools,' M.S. thesis, NTNU, 2021.

[19] T. Hiltunen, 'Learning and teaching programming skills in finnish primary schools–the potential of games,' *University of Oulu, Oulu*, 2016.

[20] C. S. Cheah, 'Factors contributing to the difficulties in teaching and learning of computer programming: A literature review,' *Contemporary Educational Technology*, vol. 12, no. 2, ep272, 2020.

[21] L. E. Winslow, 'Programming pedagogy—a psychological overview,' *ACM Sigcse Bulletin*, vol. 28, no. 3, pp. 17–22, 1996.

[22] P. Hubwieser, 'The darmstadt model: A first step towards a research framework for computer science education in schools,' in *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, Springer, 2013, pp. 1–14.

[23] J. Vahrenhold, M. Caspersen, G. Berry, J. Gal-Ezer, M. Kölling, A. McGettrick, E. Nardelli, C. Pereira and M. Westermeier, 'Informatics education in europe: Are we all in the same boat?,' 2017.

[24] S. Kjällander, A. Åkerfeldt, L. Mannila and P. Parnes, 'Makerspaces across settings: Didactic design for programming in formal and informal teacher education in the nordic countries,' *Journal of Digital Learning in Teacher Education*, vol. 34, no. 1, pp. 18–30, 2018.

[25] J. Gal-Ezer and C. Stephenson, 'A tale of two countries: Successes and challenges in k-12 computer science education in israel and the united states,' *ACM Transactions on Computing Education (TOCE)*, vol. 14, no. 2, pp. 1–18, 2014.

[26] M. Rouhani, M. Divitini and A. M. Hashemi, 'Computer science in schools: A literature mapping of professional development for in-service teachers,' in *2022 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2022, pp. 164–173.

[27] M. Rouhani, M. Lillebo, V. Farshchian and M. Divitini, 'Professional development for in-service teachers of programming: Evaluation of a university-level program,' in *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, Springer, 2021, pp. 123–134.

[28] A. Joshi, A. Jain, E. Covelli, J.-h. Yeh and T. Andersen, 'A sustainable model for high-school teacher preparation in computer science,' in *2019 IEEE Frontiers in Education Conference (FIE)*, IEEE, 2019, pp. 1–9.

[29] L. Darling-Hammond, M. E. Hyler and M. Gardner, 'Effective teacher professional development.,' 2017.

[30] M. S. Garet, A. C. Porter, L. Desimone, B. F. Birman and K. S. Yoon, 'What makes professional development effective? results from a national sample of teachers,' *American educational research journal*, vol. 38, no. 4, pp. 915–945, 2001.

[31] J. Hunzicker, 'Characteristics of effective professional development: A checklist.,' *Online Submission*, 2010.

[32] S. Gibson and C. Brooks, 'Rethinking 21st century professional development,' *Literacy Information and Computer Education Journal*, vol. 4, no. 2, pp. 973–982, 2013.

[33] S.-C. Kong, M. Lai and D. Sun, 'Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy,' *Computers & Education*, vol. 151, p. 103 872, 2020.

[34] N. Monjelat and A. Lantz-Andersson, 'Teachers' narrative of learning to program in a professional development effort and the relation to the rhetoric of computational thinking,' *Education and Information Technologies*, vol. 25, no. 3, pp. 2175–2200, 2020.

[35] Y. J. Reimer, M. Coe, L. M. Blank and J. Braun, 'Effects of professional development on programming knowledge and self-efficacy,' in *2018 IEEE Frontiers in Education Conference (FIE)*, IEEE, 2018, pp. 1–8.

[36] T. E. Reding and B. Dorn, 'Understanding the" teacher experience" in primary and secondary cs professional development,' in *Proceedings of the 2017 ACM conference on international computing education research*, 2017, pp. 155–163.

[37] P. Mozelius and N. Humble, 'Lessons learnt from teacher professional development in programming,' in *INTED2020 Proceedings*, IATED, 2020, pp. 3903–3908.

[38] L. Darling-Hammond, 'Developing professional development schools: Early lessons, challenge, and promise,' *Professional development schools: Schools for developing a profession*, pp. 1–27, 1994.

[39] R. Society, 'After the reboot: Computing education in uk schools,' *Policy Report*, 2017.

[40] T. Anderson, *The theory and practice of online learning*. Athabasca University Press, 2008.

[41] A. G. Picciano, 'Theories and frameworks for online education: Seeking an integrated model,' in *A Guide to Administering Distance Learning*, Brill, 2021, pp. 79–103.

[42] A. Yadav, C. Connolly, M. Berges, C. Chytas, C. Franklin, R. Hijón-Neira, V. Macann, L. Margulieux, A. Ottenbreit-Leftwich and J. R. Warner, 'A review of international models of computer science teacher education,' in *Proceedings of the 2022 Working Group Reports on Innovation and Technology in Computer Science Education*, ser. ITiCSE-WGR '22, , Dublin, Ireland, Association for Computing Machinery, 2022, pp. 65–93, ISBN: 9798400700101. DOI: 10.1145/3571785.3574123. [Online]. Available: https://doi.org/10.1145/3571785.3574123.

[43] L. Ni, G. Bausch and R. Benjamin, 'Computer science teacher professional development and professional learning communities: A review of the research literature,' *Computer Science Education*, vol. 33, no. 1, pp. 29–60, 2021. DOI: 10.1080/08993408.2021.1993666.

[44] C. Bellettini, V. Lonati, D. Malchiodi, M. Monga and A. Morpurgo, 'Informatics and computational thinking: A teacher professional development proposal based on social-constructivism,' in *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, Springer, 2018, pp. 194–205.

[45] M. S. Knowles, E. F. Holton III and R. A. Swanson, *The adult learner: The definitive classic in adult education and human resource development*. Routledge, 2014.

[46] R. H. Kay and A. LeSage, 'Examining the benefits and challenges of using audience response systems: A review of the literature,' *Computers & Education*, vol. 53, no. 3, pp. 819–827, 2009.

[47] J. W. Thomas, 'A review of research on project-based learning. 2000,' *The Autodesk Foundation: San Rafael*, 2010.

[48] S. Han, B. Yalvac, M. M. Capraro and R. M. Capraro, 'In-service teachers' implementation and understanding of stem project based learning,' *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 11, no. 1, pp. 63–76, 2015.

[49] A. Yadav, S. Gretter, S. Hambrusch and P. Sands, 'Expanding computer science education in schools: Understanding teacher experiences and challenges,' *Computer Science Education*, vol. 26, no. 4, pp. 235–254, 2016.

[50] A. Hutchison, L. Nadolny and A. Estapa, 'Using coding apps to support literacy instruction and develop coding literacy,' *The Reading Teacher*, vol. 69, no. 5, pp. 493–503, 2016.

[51] N. Humble and P. Mozelius, 'Teacher perception of obstacles and opportunities in the integration of programming in k-12 settings,' *Proceedings of EDULEARN*, vol. 162, 2019.

[52] M. Zee and H. M. Koomen, 'Teacher self-efficacy and its effects on classroom processes, student academic adjustment, and teacher well-being: A synthesis of 40 years of research,' *Review of Educational research*, vol. 86, no. 4, pp. 981–1015, 2016.

[53] L. Mannila, L.-Å. Nordén and A. Pears, 'Digital competence, teacher self-efficacy and training needs,' in *Proceedings of the 2018 ACM Conference on International Computing Education Research*, 2018, pp. 78–85.

[54] S. Ahmad and J. Ghazali, 'Programming teaching and learning: Issues and challenges,' *Fstm. Kuis. Edu. My*, vol. 16, no. 1, pp. 724–398, 2020.

[55] V. Singh and A. Thurman, 'How many ways can we define online learning? a systematic literature review of definitions of online learning (1988-2018),' *American Journal of Distance Education*, vol. 33, no. 4, pp. 289–306, 2019.

[56] C. Dede, D. Jass Ketelhut, P. Whitehouse, L. Breit and E. M. McCloskey, 'A research agenda for online teacher professional development,' *Journal of teacher education*, vol. 60, no. 1, pp. 8–19, 2009.

[57] L. S. Shulman, 'Those who understand: Knowledge growth in teaching,' *Educational researcher*, vol. 15, no. 2, pp. 4–14, 1986.

[58] P. Mishra and M. J. Koehler, 'Technological pedagogical content knowledge: A framework for teacher knowledge,' *Teachers college record*, vol. 108, no. 6, pp. 1017–1054, 2006.

[59] H. L. Sharma and P. Saarsar, 'Tps (think-pair–share): An effective cooperative learning strategy for unleashing discussion in classroom interaction,' *International Journal of Research in Social Sciences*, vol. 8, no. 5, pp. 91–100, 2018.

[60] L. C. Kidder, 'The multifaceted endeavor of online teaching: The need for a new lens,' in *The design of learning experience*, Springer, 2015, pp. 77–91.

[61] K. Swan, 'Social construction of knowledge and the community of inquiry framework,' in *Open and distance education theory revisited*, Springer, 2019, pp. 57–65.

[62] S. J. Derry, 'A fish called peer learning: Searching for common themes,' *Cognitive perspectives on peer learning*, vol. 9, no. 1, pp. 197–211, 1999.

[63] M. McMahon, 'Social constructivism and the world wide web - a paradigm for learning,' 1997.

[64] P. Shea, J. Richardson and K. Swan, 'Building bridges to advance the community of inquiry framework for online learning,' *Educational Psychologist*, vol. 57, no. 3, pp. 148–161, 2022.

[65] C. Rodgers, 'Defining reflection: Another look at john dewey and reflective thinking,' *Teachers college record*, vol. 104, no. 4, pp. 842–866, 2002.

[66] J. Dewey, *How we think: A restatement of the relation of reflective thinking to the educative process*. DC Heath, 1933.

[67] D. R. Garrison, T. Anderson and W. Archer, 'Critical thinking, cognitive presence, and computer conferencing in distance education,' *American Journal of distance education*, vol. 15, no. 1, pp. 7–23, 2001.

[68] Y. Z. Olpak, M. Yagci and U. Basarmak, 'Determination of perception of community of inquiry.,' *Educational Research and Reviews*, vol. 11, no. 12, pp. 1085–1092, 2016.

[69] J. Short, E. Williams and B. Christie, *The social psychology of telecommunications*. Toronto; London; New York: Wiley, 1976.

[70] K. Swan and L. F. Shih, 'On the nature and development of social presence in online course discussions,' *Journal of Asynchronous learning networks*, vol. 9, no. 3, pp. 115–136, 2005.

[71] D. R. Garrison, *E-learning in the 21st century: A community of inquiry framework for research and practice*. Routledge, 2016.

[72] J. D. Bransford, A. L. Brown, R. R. Cocking *et al.*, *How people learn*. Washington, DC: National academy press, 2000, vol. 11.

[73] C. Bosch, 'Promoting self-directed learning through the implementation of cooperative learning in a higher education blended learning environment [doctoral dissertation]. 2016,' *Johannesburg, SA: North-West University*, 2016.

[74] J. Rogalski and R. Samurçay, 'Acquisition of programming knowledge and skills,' in *Psychology of programming*, Elsevier, 1990, pp. 157–174.

[75] R. Juárez-Ramırez, C. X. Navarro, V. Tapia-Ibarra, R. Macıas-Olvera and C. Guerra-Garcıa, 'What is programming? putting all together-a set of skills required,' in *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*, IEEE, 2018, pp. 11–20.

[76] B. S. Xia, 'A pedagogical review of programming education research: What have we learned,' *International Journal of Online Pedagogy and Course Design (IJOPCD)*, vol. 7, no. 1, pp. 33–42, 2017.

[77] M. Rouhani, V. Farshchian and M. Divitini, 'Teaching programming in secondary schools: Stepping and stumbling stones,' 2021.

[78] J. Thorsnes, M. Rouhani and M. Divitini, 'In-service teacher training and self-efficacy,' in *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, Springer, 2020, pp. 158–169.

[79] M. Rouhani, 'Utilizing slack as a communication platform in a flexible learning trajectory course: Supporting the learning process,' in *Proceedings of the 9th Computer Science Education Research Conference*, 2020, pp. 1–7.

[80] M. Rouhani and V. Jørgensen, 'In-service teachers' attitude towards programming for all,' in *Ludic, Co-design and Tools Supporting Smart Learning Ecosystems and Smart Education*, Springer, 2022, pp. 149–162.

[81] T. Gaddis, *Starting Out with Python [Global Edition].* Pearson Education Limited, 2021.

[82] M. Rouhani, *Introductory programming for teachers*, (Accessed on 28/11/2022), Aug. 2018. [Online]. Available: `https://www.ntnu.edu/studies/courses/IT6203`.

[83] M. Rouhani, *Applied programming for teachers*, (Accessed on 28/11/2022), Jan. 2019. [Online]. Available: `https://www.ntnu.edu/studies/courses/IT6204`.

[84] J. W. Cresswell and V. L. Plano Clark, 'Designing and conducting mixed methods research,' 2011.

[85] A. Hevner, S. March and K. Park, 'Design research in information systems research,' *MIS Quarterly*, vol. 28, no. 1, pp. 76–105,

[86] P. Järvinen, 'Action research is similar to design science,' *Quality & Quantity*, vol. 41, no. 1, pp. 37–54, 2007.

[87] K. Peffers, T. Tuunanen, M. A. Rothenberger and S. Chatterjee, 'A design science research methodology for information systems research,' *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.

[88] J. Saldaña, 'The coding manual for qualitative researchers,' *The coding manual for qualitative researchers*, pp. 1–440, 2021.

[89] M. Rouhani, M. Lillebo, V. Farshchian and M. Divitini, 'Learning to program: An in-service teachers' perspective,' in *2022 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2022, pp. 123–132.

[90] J. Sorva, 'How concrete should an abstract be?' In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, 2021, pp. 568–574.

[91] Wikipedia, *Educational stage*, (Accessed on 07/15/2021), Feb. 2015. [Online]. Available: `https://en.wikipedia.org/wiki/Educational%5C_stage`.

[92] D. C. Webb, H. Nickerson and J. B. Bush, 'A comparative analysis of online and face-to-face professional development models for cs education,' in *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*, 2017, pp. 621–626.

[93] S. Darvishi, *The use of slack as a social media in higher education: Students perceptions of advantages and disadvantage of slack during learning process*, 2020.

[94] T. Panitz, 'Collaborative versus cooperative learning: A comparison of the two concepts which will help us understand the underlying nature of interactive learning.,' 1999.

[95] M. Rouhani, 'Programming for teachers: Supporting participants in defining their learning path in a flexible online learning trajectory course,' *International Journal of Childhood Education*, vol. 1, no. 1, pp. 13–23, 2020.

[96] I. Esmonde, 'Power and sociocultural theories of learning,' in *Power and privilege in the learning sciences*, Routledge, 2016, pp. 24–45.

[97] C. Angeli and M. Giannakos, *Computational thinking education: Issues and challenges*, 2020.

[98] A. Yadav, S. Gretter, J. Good and T. McLean, 'Computational thinking in teacher education,' in *Emerging research, practice, and policy on computational thinking*, Springer, 2017, pp. 205–220.

[99] X.-M. Wang, G.-J. Hwang, Z.-Y. Liang and H.-Y. Wang, 'Enhancing students' computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt,' *Journal of Educational Technology & Society*, vol. 20, no. 4, pp. 58–68, 2017.

[100] M. Miterianifa, A. Ashadi, S. Saputro and S. Suciati, 'Higher order thinking skills in the 21st century: Critical thinking,' in *Proceedings of the 1st International Conference on Social Science, Humanities, Education and Society Development, ICONS 2020, 30 November, Tegal, Indonesia*, 2021.

[101] J. S. Oliver and R. D. Simpson, 'Influences of attitude toward science, achievement motivation, and science self concept on achievement in science: A longitudinal study.,' *Science Education*, vol. 72, no. 2, pp. 143–55, 1988.

[102] L. Cooke, J. Schugar, H. Schugar, C. Penny and H. Bruning, 'Can everyone code?: Preparing teachers to teach computer languages as a literacy,' in *Participatory Literacy Practices for P-12 Classrooms in the Digital Age*, IGI Global, 2020, pp. 163–183.

[103] H. C. Jiau, J. C. Chen and K.-F. Ssu, 'Enhancing self-motivation in learning programming using game-based simulation and metrics,' *IEEE Transactions on Education*, vol. 52, no. 4, pp. 555–562, 2009.

[104] A. Luxton-Reilly, 'Learning to program is easy,' in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 2016, pp. 284–289.

[105] A. Luxton-Reilly, 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*, 2018, pp. 55–106.

[106] E. Hartell, A. Doyle and L. Gumaelius, 'Teachers' attitudes towards teaching programming in swedish technology education,' *PATT 37*, p. 195, 2019.

[107] N. C. Brown, S. Sentance, T. Crick and S. Humphreys, 'Restart: The resurgence of computer science in uk schools,' *ACM Transactions on Computing Education (TOCE)*, vol. 14, no. 2, pp. 1–22, 2014.

[108] V. F. Savec, 'Use of ict and innovative teaching methods for stem,' *Innovative Approaches to the Application of Digital Technologies in Education and Research SLET-2019, Stav-ropol-Dombay, Russia*, pp. 1–10, 2019.

[109] V. Zygouris-Coe, 'Collaborative learning in an online teacher education course: Lessons learned,' *ICICTE Proceedings*, pp. 332–342, 2012.

[110] F. T. Igira and J. Gregory, 'Cultural historical activity theory,' *Handbook of research on contemporary theoretical models in information systems*, pp. 434–454, 2009.

[111] V. Farnsworth, I. Kleanthous and E. Wenger-Trayner, 'Communities of practice as a social theory of learning: A conversation with etienne wenger,' *British journal of educational studies*, vol. 64, no. 2, pp. 139–160, 2016.

[112] S. F. Akkerman and A. Bakker, 'Boundary crossing and boundary objects,' *Review of educational research*, vol. 81, no. 2, pp. 132–169, 2011.

[113] J. W. Creswell and V. L. P. Clark, *Designing and conducting mixed methods research*. Sage publications, 2017.

[114] A. J. Onwuegbuzie and R. B. Johnson, 'The validity issue in mixed research,' *Research in the Schools*, vol. 13, no. 1, pp. 48–63, 2006.

[115] M. Borrego, E. P. Douglas and C. T. Amelink, 'Quantitative, qualitative, and mixed research methods in engineering education,' *Journal of Engineering education*, vol. 98, no. 1, pp. 53–66, 2009.

[116] K. Malterud, 'Qualitative research: Standards, challenges, and guidelines,' *The lancet*, vol. 358, no. 9280, pp. 483–488, 2001.

[117] N. Hasan, R. U. Rana, S. Chowdhury, A. J. Dola and M. K. K. Rony, 'Ethical considerations in research,' *Journal of Nursing Research, Patient Safety and Practise (JNRPSP) 2799-1210*, vol. 1, no. 01, pp. 1–4, 2021.

[118] M. Klawe, T. Whitney and C. Simard, 'Women in computing—take 2,' *Communications of the ACM*, vol. 52, no. 2, pp. 68–76, 2009.

[119] J. Flapan, J. J. Ryoo and R. Hadad, 'Building systemic capacity to scale and sustain equity in computer science through multi-stakeholder professional development,' in *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, IEEE, vol. 1, 2020, pp. 1–8.

[120] J. Thorsnes, 'Teachers' attitudes and self-efficacy towards teaching programming and the impact of continuing education,' M.S. thesis, NTNU, 2020.

Part II

Papers

Paper I

Please note that the apostrophe (') has been mistakenly replaced with characters **âĂI**, **âĂİ**, and **âĂŹ** on chapter 2 and 4 of the published version of this paper.

M. Rouhani, M. Divitini, V. Vujosevic, S. Stai and H. A. Olstad, 'Design of a programming course for teachers supporting flexible learning trajectories,' in *Proceedings of the 8th Computer Science Education Research Conference*, 2019, pp. 33–38

This paper is not included due to ACM copyright restrictions
available at
https://doi.org/10.1145/3375258.3375263

Paper II

J. Thorsnes, M. Rouhani and M. Divitini, 'In-service teacher training and self-efficacy,' in *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, Springer, 2020, pp. 158–169

In-Service Teacher Training and Self-efficacy

Jørgen Thorsnes, Majid Rouhani, and Monica Divitini( )

Department of Information and Computer Science, NTNU, Trondheim, Norway
{jorgen.thorsnes,majid.rouhani,monica.divitini}@ntnu.no

Abstract. Programming is increasingly introduced in secondary schools, both as a stand-alone subject or integrated into other subjects, leading to growing attention to the training of in-service teachers. Teachers need to learn both (a) how to program and (b) how to teach programming, often in the context of different disciplines. The paper explores the impact of a university-level training program offered to in-service teachers, with a focus on teachers' self-efficacy in teaching programming. The paper reports the interviews with ten teachers after about one year they have completed the program. The results indicate that the training has improved teachers' self-efficacy, and the impact is lasting in time. Also, some teachers expressed concerns about their skill level in programming, but this does not necessarily associate with lower self-efficacy in teaching programming. The paper presents the results from the study and some implications for the design of training of in-service programming teachers.

Keywords: In-service teacher training · Self-efficacy · Programming

1 Introduction

In a recent report on the status of Informatics education in Europe, it is recommended that all pupils must have access to ongoing education in Informatics, and the teaching must be undertaken only by teachers who have formal education in Informatics [1]. However, there are several challenges to meet this recommendation and a growing demand for teacher training, in particular for the re-skilling of in-service teachers. In this context, it is essential to understand in-service teacher training to define relevant and effective training. In this paper, we focus on how formal training of in-service teachers impacts self-efficacy, with a focus on long-term impact. Self-efficacy refers to the belief in one's capabilities to organize and execute the courses of action required to produce a given result [2]. In terms of teaching a specific subject, the teacher's self-efficacy refers to their belief in their capabilities to teach the subject, such that pupils achieve the desired learning outcomes. Positive self-efficacy is connected to increased student and teacher outcomes, and it has a positive influence on teachers' psychological well-being [3]. However, several studies identify challenges with self-efficacy connected to programming education. For example, in two recent Swedish studies on teachers' attitudes and self-efficacy towards programming, the researchers found that many Swedish teachers lack confidence in teaching programming [4, 5]. Similar results were reported in UK

K. Kori and M. Laanpere (Eds.): ISSEP 2020, LNCS 12518, pp. 158–169, 2020.
https://doi.org/10.1007/978-3-030-63212-0_13

schools, with many teachers worried that they miss practical and theoretical knowledge of computing [6]. Given this background, we investigate how formal education at the university level for in-service programming teachers affects their self-efficacy. The main research question explored in this paper is: *How do in-service teachers perceive the lasting effect of programming education concerning their self-efficacy in programming and teaching programming?* To answer this question, we interviewed ten teachers who attended a university-level program on programming and programming education. Interviews were conducted almost one year after the completion of the course.

2 Case and Method

Case Description. Our study is connected to the in-service teacher training program at our university. The program consists of two courses of 7,5 ECTS each, the first with a focus on basic programming and the second on teaching programming. In the first course, teachers get an introduction to Python. The second course is more flexible, and teachers can select programming languages and topics on which to specialize based on their interests and needs [7]. Though there are no requirements for teachers to follow both of the courses, most do. The study program is aimed at in-service teachers in grades 8–13 (Lower and Upper Secondary School). The program is an online study, with web-based lectures and weekly activities such as online lectures and regular compulsory work exercises. Students participate in the course with the support of their school, which is committing to provide some free time to teachers to complete the course, though they continue their primary duties during the two semesters. The additional costs for the schools are partly covered by a national program of the Ministry of Education. This support leads to a very high completion rate. A survey distributed at the end of the program indicates high levels of satisfaction with the course. With this follow up study, we investigate how the training has contributed to teachers' self-efficacy and the long-term impact of the educational program.

Overall Method. This study is based on interviews with teachers that participated in the continuing education program in 2018-19. The study uses semi-structured interviews to explore the research questions by capturing teachers' reflections on their self-efficacy towards teaching programming.

Interview Guide. The interview guide was constructed based on three main elements: (a) Attitudes towards programming in school; (b) Self-efficacy in teaching programming; (c) Self-efficacy in programming. Since we are interested in understanding the impact of the program, for each element, we added questions connected to the perceived impact of the program and changes since its completion.

For Attitude (a), we asked teachers their opinion about the ongoing introduction of programming in different subjects and, specifically, in the subjects that they teach. The questions related to Self-Efficacy in Programming (b) are inspired by the Teachers' Sense of Efficacy Scale (TSES) [8], with a focus on Efficacy for instructional strategies. Concerning teachers' programming skills (c), which is also relevant to teachers' ability to teach programming [9], questions were created to investigate how the teachers feel towards their own programming skills.

The questions were tested through test-interviews with three pre-service math teachers. The interviews were conducted during the COVID-19 pandemic. Schools were closed, with all the teaching taking place online and some elective subjects being postponed. We, therefore, added a final question on COVID-19. Our goal with this question was not to investigate its broader impact, but simply to check the validity of our study.

Participants. An invitation letter was sent to all the participants of the 2018/2019 cohort. An interview was then planned with the ones who expressed interest in participating in the study. Table 1 provides an overview of the participants.

Table 1. Overview of interviewees with Gender (M/F), school level (Upper Secondary School, USS, or Lower Secondary School, LSS), Subjects they are teaching (in italics the ones where they are *not* expected to use programming); Type of school (General, G, or Vocational, V)

ID	G	Level	Subject(s)	Type
1	F	USS	Economics	V
2	M	USS	Math, physics	G
3	F	USS	Math, natural science	V
4	M	LSS	Math, natural science, programming	G
5	F	LSS	*English, social studies, gymnastics*, programming	G
6	M	USS	Math, physics	G
7	F	USS	Math, *computer and electronics*	V
8	F	LSS	Math, natural science, *religion*, programming, *work-related training*	G
9	F	USS	*Physics*, math, programming, natural science, *technology and research*	G
10	F	USS	Construction - and control technique	V

Interviews were conducted via Zoom, using the service offered internally by our university for GDPR-compliance. The interviews were recorded with an external recorder and then transcribed by the interviewer. The relevant national agency approved the research. All the participants have been informed about the study, their rights and have been explicitly given their consent.

Analysis. After the transcription of the interviews, a thematic analysis [10] was performed by one of the authors using themes connected to the Teachers' Sense of Efficacy scale. The coding process was done with NVivo (QSR International, 2018). The final categories are: Attitudes towards programming in school; Teaching programming self-efficacy; Programming skill; Impact of programming education; Impact of time after programming education; COVID-19.

3 Results

3.1 Attitudes Towards Programming in School

The interviewees are generally positive towards teaching programming in school. They also express that they are positive towards using programming in an interdisciplinary context and underline the relevance for future jobs.

However, they also expressed some concerns that the inclusion of programming will be a long process, without a quick fix, especially considering the number of teachers who do not have any competence or education in programming. One of the issues that emerged from the analysis is the importance of teachers' community and collaboration. Four of the teachers expressed that they found collaboration with colleagues important when dealing with programming in school. Those who had someone to collaborate with reported that it was beneficial. Some other teachers indicated that they do not have colleagues to work within programming and that they would like to have that. Also, some teachers reported to have colleagues who are rather hostile to programming in school:

> *… Colleagues? They can be absolutely cruel! … I heard the lecturers talk about programming in school as the future… You know, I was standing talking to one of my younger colleagues in the hallway in front of the coffee machine, and a colleague came past me, jumping out of the neighboring room and scolding me! So, it's like that, and it shouldn't even be mentioned at work…. (Teacher 3-Female-Math-USS)*

Some teachers also talked about gender differences related to programming. Particularly worrying is the resistance of gender stereotypes. One male teacher expressed that male teachers were more interested in programming than female teachers. One female teacher experienced that male teachers got much of the responsibility of programming related tasks in school, even if she is the only teacher with programming education:

> *…But I notice at work, that when being a woman – "no, you have almost no clue," they put the men to take those jobs…. (Teacher 3-Female-Math-USS)*

3.2 Teaching Programming and Self-efficacy

In general, all ten teachers responded that they could teach programming, even if this is mostly connected to a specific subject or school level, for example:

> *…I can't teach block programming, I can't make a lesson in game programming, I can't make a lesson in micro:bit … But I think I can make good lessons and exercises that are relevant to my subjects, such as solving differential equations, solving equations with numerical methods, etc.…. (Teacher 2-Male-Math-USS)*

Adapted Teaching. All of the teachers expressed that they could provide suitable challenges to capable pupils in programming, for example:

… I am very focused on giving open assignments because I have pupils on the whole scale … I really feel that with open assignments, I can differentiate to different levels, yes. (Teacher 7-Female-Math-HS)

Some teachers explained that even though some pupils might be better than them in programming, this is not a major challenge. Teachers could usually find suitable assignments together with the pupils, or the capable pupils could get appropriate challenges through open tasks or freely choosing what they work with. In a recent study [11], US K-12 computer science teachers reported that it was challenging to meet all pupils' needs on an individual level. However, in our study, only one of the teachers indicated a lower sense of self-efficacy in adapting her teaching to her pupils. Despite this, she still felt that she could provide appropriate challenges for highly skilled pupils by giving them freedom in what they were doing.

Assessment. The data analysis reveals that the teachers had a more varied sense of self-efficacy when it comes to assessment in programming. Four of the teachers expressed that they find assessment in programming difficult, for example because it is easy to find solutions online for the pupils and that they need strategies and tools for assessment. As a teacher explains:

…I think I need a strategy or tools for this. I think it is difficult. It's a little bit like putting your finger in the air when you think about the assessment of the pupils. I've had some assignments where they have to program something, and it's hard to know if they have copied the solution or whether it is their own, one must actually observe the whole process, and that is simply incredibly difficult…. (Teacher 4-Male-Math-SS)

Other teachers, however, are confident that they can assess their pupils, and some suggested oral presentations as a useful method, also to unveil whether the pupils understand their solutions or if they have copied it. One of the teachers that finds assessment challenging feels that it was little focus on this in the program.

Motivation. In general, the teachers seemed to have a relatively high sense of self-efficacy in motivating their pupils in programming. Nine of the teachers expressed that they felt they could motivate their pupils to learn to program. Some also thought that it was easier to motivate than in other subjects. For example:

…Yes. … they [pupils] get to try something new. And those who have some prior knowledge get to do something they master. So yes, I think it is easier to motivate them in programming than in accounting, for example…(Teacher 1-Female-Economics-USS)

However, one of the teachers reported challenges with motivating students in elective courses:

… It's about how you meet the students. And in discussing with them, attempt to find angles of attack that motivate them. I feel that I manage that with some

pupils, but then there is a problem in that not all pupils in elective programming are necessarily motivated to learn to program. … Some of the pupils are there just because they did not get into the elective they wanted… that's a challenge… (Teacher 4-Male-Math-SS)

Explaining and Conveying Programming Knowledge. Of the ten teachers, eight of them believe that, to some extent, they can explain programming concepts and come up with alternative explanations when pupils do not fully understand. Two of the teachers expressed that they could explain some programming concepts, but probably not all. Six of the teachers believed that they would not be able to answer difficult programming questions from the more capable pupils. The other four thought they could answer some difficult questions, but not all. However, nine of the teachers believed that they could either come back to the pupil with the answer later or find the answer together:

… the pupils are also quite understanding when you say, "I can't do this very well, but I find it very fun! And I want to show you, and then we can figure it out together". They understand that, kind of. (Teacher 1-Female-Economics-HS)

The teachers seemed to have a moderately high sense of self-efficacy in this theme, but the main challenge is that the teachers do not perceive their programming competence as very advanced.

Developing Teaching Material. Eight of the teachers expressed that they can create good lesson plans and exercises, though it might be time-consuming, and that it would be beneficial with more time for planning lessons. Two of the teachers stated that they could not create suitable lessons from scratch, but they could by adapting existing teaching resources:

… I'm probably more about finding and adapting than making them myself. I don't feel I have enough expertise for that…. (Teacher 5-Female-English-SS)

Most of the teachers state that they are using and finding teaching material online and adapt it to their classes. Four of the teachers expressed that they would like more relevant teaching material resources available. In general, the teachers indicated that they had a relatively high sense of self-efficacy in designing lessons in or with programming when there is relevant teaching material that they can adapt to their teaching. However, the willingness of experimenting with new lesson plans might be limited:

…I don't know if I'm going to make that much varied, and I'm not so secure in the coding that I just toss myself into it and just try everything possible, so I limit it to something that I see will work, or something I've experienced that worked earlier… (Teacher 9-Female- Math-HS)

Challenges in Teaching Programming. When the teachers were asked what they perceived as the biggest challenge in teaching programming, two themes were prominent: Pupils' digital competences and technical issues. Three of the teachers talked about pupils' computer skills as a challenge. For example:

> *… Several of them name their files "one" "two" "three" and such, they do not have any system. So it will be difficult when you need to help them find a structure in programming when they can't even structure other things, so I think that might be most challenging…(Teacher 1-Female-Economics-USS)*

Three of the teachers also talked about technical challenges. One of the teachers answered that some computer programs are challenging to use and cause technical problems and that she would like more user-friendly programs. Two other teachers explained that there are many technical issues, for example:

> *…The biggest challenge is technical. For example, when we code in Python, there are a lot of libraries and stuff that one needs, and then the pupils may have different versions, and different modules and libraries, and nothing matches … so we end up turning the computer off and on again, restarting, get frustrated because something that works, or code that works on one PC doesn't work on another PC. And that is by far the most frustrating, and what we spend the most time on - unnecessary time. Sometimes we also give up … And there I have no competence to find out what the problem is…. (Teacher 9-Female- Math-USS)*

3.3 Programming Skill

The teachers were asked how they perceive their own programming skills. Eight of the teachers expressed that their programming skills were sufficiently good for teaching in their subjects and grades. However, six of the teachers indicated that their programming skill was relatively low, as one teacher states:

> *…For example, it is when we have embarked on slightly larger projects, which I may not have complete control over the development in. But so far, I have not been on extremely thin ice, but I have felt that "oh, I have to go home and pick up the book and read some more" I have had some of those rounds with myself…. Teacher 5-Female-English-LSS)*

In general, the answers indicate that the teachers did not have a very high sense of self-efficacy towards programming, but that this did not severely impact their self-efficacy towards teaching programming. It also seems that even when their programming skills are relatively low, teachers perceive that they can increase this skill. Three of the teachers explained that they would like to have more follow-up in term of exercises or a local programming group to get better at programming:

> *…Yes, that it becomes just like an anonymous alcoholics group, that you have a follow-up group, "anonymous coders" who need some follow-up. Get some challenges and keep up …. (Teacher 4-Male-Math-LSS)*

Previous research suggests that it is essential for teachers of programming to attain skills in programming [9]. This is also found in this study. However, even when the teachers perceive their own programming skills as not very high, they still feel capable of teaching programming in their grades and subjects. Many of the teachers also report that they feel capable of increasing their programming skills on their own or with colleagues and that they will get better with experience. This result indicates high self-efficacy towards getting better in programming.

3.4 Impact of Programming Education on Self-efficacy in Teaching Programming

Six of the teachers expressed that they could not teach programming before the courses, but that they felt they were able to teach it in their subjects and grades after the program. This result indicates a very positive impact on the teacher's self- efficacy towards teaching programming. Three of the teachers felt they could have taught programming before the courses, but express that they felt more secure in their teaching afterward. One teacher thought that the first course had a negative impact on her ability to teach programming because of the demanding workload and difficult exercises made her more confused than competent. However, this improved during the second course. Five of the teachers also stated that programming education had a positive impact on their attitudes towards programming in school, especially in terms of why programming in school can be relevant and beneficial in other subjects.

Some of the teachers felt that the learning curve in the first course was very steep. That programming can be hard to learn is often reported in the literature, e.g., [12]. Most of the teachers still felt that they learned a lot from the first course, and most were happy with both the learning outcome and the workload in the second course. Two of the teachers, however, reported that they thought the introductory programming course was very good, while they felt the second course was either too easy or had little impact on their competence.

3.5 Impact of Time After Course

The interviews were conducted close to one year after the teachers finished their programming studies. Therefore, they were also asked how the time that had passed since the completion of the program had impacted on their teaching. Most teachers perceived their programming skills lowered over time, when not used actively:

> *…I don't have it as much in my fingers anymore, since I work less on it myself… I notice that programming is something I should keep a lot more maintained. It's like programming is a skill that you have to practice, to a much greater extent than math and physics, where you basically have it…. (Teacher 9-Female-Math-HS)*

At the same time, these teachers expressed that they could "refresh" their programming skills and knowledge with little trouble. Also, a lower self-efficacy towards programming does not seem to relate to lower self-efficacy for teaching it.

The teachers that have taught or used programming in their classes in the last year explained that they also feel more capable of teaching programming due to the experience, for example:

> *…I have used programming more in teaching this year than I did the year before. So I feel that I am more secure in the role, and promote it more, and want more people to use it…. (Teacher 7-Female-Math-USS)*

Only one of the teachers felt less competent in pedagogical aspects of programming because he has not been teaching programming in the last year, but he also states that it will come back when he starts planning for it:

…I feel less ready now because I haven't practiced it in a year. No, that's not entirely true, I'm lying, but if you had me sit down with an exam in coding now, I would have been better off a year ago than now. Pedagogically as well. But it will return when I start planning a bit again, and I've looked a bit on it, I've discussed a bit with some colleagues, and worked a little with coding …(Teacher 2-Male-Math-USS)

Some of the teachers stated that they would have liked more follow up exercises or a community of teachers to help in maintaining programming skills after the courses. This suggestion can be seen in relation to the fact that many programming teachers work in schools without other teachers in their content area [11].

3.6 Impact of COVID-19

Five of the teachers stated that the COVID-19 situation might have impacted their answers. Interesting is, for example, that two of the teachers reported that they had found new teaching methods. For example, one stated:

…Maybe towards differentiation in that I discovered that TinkerCad had some functions similar to Scratch because I have never used TinkerCad before. I have been very focused on the pupils working with it physically, which we now could not … But when I used it now, I saw that it was easier to differentiate for the students because I could use block programming … So yeah, so that's how it probably affected because I've discovered new things during this period because I had to make way for another way of teaching… (Teacher 7-Female-Math-USS)

One teacher reported that the answers might have been affected by insecurity about how to continue teaching programming in the current situation. Other teachers reported the lack of contact with colleagues and general concerns for the future, considering that many other teachers have not been able to follow planned training in the Spring. Interesting is also to see that the situation might have increased concerns about the general digital competence of other teachers:

…I am one of those who are positive towards programming. But of course, I see big challenges in including it in the subjects. Because now, when we are teaching through Teams, we have employees that struggle with technical stuff there, right. I don't think this will be done quickly…. (Teacher 8-Female-Math-LSS)

4 Discussion and Implications

The results of our study indicate that the teachers perceive that the education that they received had a positive impact on their self-efficacy towards teaching programming. Some of the teachers report that the studies also had a positive effect on their attitudes towards programming. The teachers say that their self-efficacy in teaching programming increases with experience in teaching programming, and does not significantly decrease without experience over time. The teachers report that their programming skills lower over time when not used, but that they can quickly refresh their programming skills when needed. This can be seen as a positive trend in relation to other studies where teachers showed a low level of self-efficacy in teaching programming [4–6]. Based on the results from the interviews, we identify some issues that need attention when designing training for in-service programming teachers.

Profiling and Flexible Paths. The evaluation of the two courses is rather varied. For some, the first course was too demanding, for others appropriate. The same holds for the second course. Though there is a space for improvement of the actual course content, we think that these differences are strongly connected with the nature itself of the program. Participating teachers have different competence levels and different needs since they teach different subjects to different students. One could advocate for more specific courses, focusing on specific needs or requiring defined competencies for admittance. However, this is a model that is difficult to implement for economic reasons, and with pedagogical limitations, caging teachers in a specific subject and level. The alternative is then to create modular courses that support different learning trajectories, as in the program of our study [7]. However, this requires identifying ways of profiling participants and scaffolding their participation in the course.

Mini-courses/Continuous Education. One introductive course on programming and teaching is not enough to meet the challenges that teachers meet every day and to stay updated. Teacher training should be seen as a continuous process, with the possibility to follow up a more formal and structured training with shorter training activities in the form of, e.g., seminars and workshops on specific programming and pedagogical issues. This training requires a commitment from the providers of training to design their courses as a continuous process that looks at long term opportunities.

Importance to Promote Community. Collaboration among programming teachers is useful, both to increase and maintain programming skills and to share and discuss the teaching of programming. Some teachers also experience that their colleagues are negative towards programming in school. There is a need to develop and strengthen communities of practice in the domain of teaching programming. This can be done at different levels. Many countries have nation-wide communities to support programming teachers, as the CAS network in the UK [13]. These constitute an essential model, but it might be equally important to create a landscape of communities, locally at the teachers' school and around specific training courses. In this perspective, it is essential during a course to promote a sense of community among the teachers attending the courses and nurturing this community after the course is completed as a form of continuous education.

Re-use of Resources. In our study, most of the teachers expressed that they can develop and adapt varied teaching material in programming and seem to have a relatively high sense of self-efficacy towards developing and adapting teaching material. However, some express that it can be difficult to find the most suitable teaching resources in the abundance of teaching resources in programming found online. A similar result is also found in the interview study of US K-12 computer science teachers [11]. Some teachers express that they would like more teaching resources in programming, as also indicated by the teachers in the study reported in [14]. The results also suggest that the teachers have a relatively high sense of self-efficacy towards motivating their pupils in programming, but that it can be difficult to create exercises that engage the lesser capable pupils in programming, while also giving them a sense of mastery. This indicates the need to create resources that can be adapted at different levels and to different national contexts, considering both language and study plan. As part of the training, it is important to include information about available resources and how to adapt them.

Assessment. Our respondents identify assessment as challenging, mirroring other studies that pointed out the need for quality assessment tools in computer science and coding education [11]. Though we do not have conclusive evidence, it also seems that assessment in programming is a more significant challenge than the teachers without experience perceive. It is, therefore, important that any teacher training program explicitly addresses this issue and challenges teachers to address assessment as an integrated part of the definition of their lesson plan.

Programming Skills. The results in this study can help ease the minds of future programming teachers that are concerned about their programming skills. Many of the teachers in this study perceived their programming skill as relatively low, but also sufficient for their teaching of programming, and they had a relatively high sense of self-efficacy in teaching programming. In other words, teachers do not necessarily need to be expert programmers to feel capable of teaching programming to their pupils, and it seems possible to attain sufficient programming skills for teaching programming through continuing education in programming over two semesters.

5 Conclusions

In this paper, we presented the results from interviews conducted with ten teachers, one year after they completed formal university-level training in programming. The teachers express high self-efficacy about teaching programming, though some of them are reporting low skills in programming. The study shows that teachers perceive formal training at the university level as having a positive and long-lasting impact.

The interviewees differ in terms of subjects they teach and school type. Therefore, the study covers different perspectives. However, we are fully aware that they are all connected to the same course, and it might be difficult to generalize the results. More studies are necessary. Considering that this type of training requires a heavy investment, both at the individual and the school level, we claim it is crucial to investigate which types of training have the highest impact, not only in terms of acquired knowledge but also in terms of attitudes and self-efficacy. From our study, formal training at the university level seems a promising option.

Acknowledgments. We thank the teachers who participated in the interviews. The work is partly funded by Excited, The Norwegian Center for Excellent IT Education (https://www.ntnu.edu/excited).

References

1. The Committee on European Computing Education (CECE): Informatics Education in Europe: Are We All In The Same Boat? Association for Computing Machinery, New York (2017)
2. Bandura, A.: Self-efficacy: toward a unifying theory of behavioral change. Psychol. Rev. **84**, 191–215 (1977). https://doi.org/10.1037/0033-295X.84.2.191
3. Zee, M., Koomen, H.M.Y.: Teacher self-efficacy and its effects on classroom processes, student academic adjustment, and teacher well-being: a synthesis of 40 years of research. Rev. Educ. Res. **86**, 981–1015 (2016). https://doi.org/10.3102/0034654315626801
4. Hartell, E., Doyle, A., Gumaelius, L.: Teachers' attitudes towards teaching programming in Swedish Technology education. Presented at the PATT 37, Pupils Attitudes Towards Technology (2019)
5. Mannila, L., Nordén, L.-Å., Pears, A.: Digital competence, teacher self-efficacy and training needs. In: Proceedings of the 2018 ACM Conference on International Computing Education Research. pp. 78–85. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3230977.3230993
6. The Royal Society: After the reboot: computing education in UK schools. The Royal Society, November 2017
7. Rouhani, M., Divitini, M., Vujosevic, V., Stai, S., Olstad, H.A.: Design of a programming course for teachers supporting flexible learning trajectories. In: Proceedings of the 8th Computer Science Education Research Conference, pp. 33–38. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3375258.3375263
8. Tschannen-Moran, M., Hoy, A.W.: Teacher efficacy: capturing an elusive construct. Teach. Teach. Educ. **17**, 783–805 (2001). https://doi.org/10.1016/S0742-051X(01)00036-1
9. Korkmaz, O.: Prospective CITE teachers' self-efficacy perceptions on programming. Procedia Soc. Behav. Sci. **83**, 639–643 (2013). https://doi.org/10.1016/j.sbspro.2013.06.121
10. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. **3**, 77–101 (2006). https://doi.org/10.1191/1478088706qp063oa
11. Yadav, A., Gretter, S., Hambrusch, S., Sands, P.: Expanding computer science education in schools: understanding teacher experiences and challenges. Comput. Sci. Educ. **26**, 235–254 (2016). https://doi.org/10.1080/08993408.2016.1257418
12. Guzdial, M.: Learner-Centered Design of Computing Education: Research on Computing for Everyone. Synthesis Lectures on Human-Centered Informatics, vol. 8, pp. 1–165 (2015). https://doi.org/10.2200/S00684ED1V01Y201511HCI033
13. Brown, N.C.C., Sentance, S., Crick, T., Humphreys, S.: Restart: the resurgence of computer science in UK schools. ACM Trans. Comput. Educ. **14**, 9:1–9:22 (2014). https://doi.org/10.1145/2602484
14. Kadirhan, Z., Gül, A., Battal, A.: Self-efficacy to teach coding in K-12 education. In: Hodges, C.B. (ed.) Self-Efficacy in Instructional Technology Contexts, pp. 205–226. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99858-9_12

Paper III

M. Rouhani, 'Utilizing slack as a communication platform in a flexible learning trajectory course: Supporting the learning process,' in *Proceedings of the 9th Computer Science Education Research Conference*, 2020, pp. 1–7

This paper is not included due to ACM copyright restrictions
available at
https://doi.org/10.1145/3442481.3442503

Paper IV

Please note that the apostrophe (’) has been mistakenly replaced with characters **â**™ on the reference list [29] of the published version of this paper.

M. Rouhani, V. Farshchian and M. Divitini, ‘Teaching programming in secondary schools: Stepping and stumbling stones,’ 2021

Teaching Programming in Secondary Schools: Stepping and Stumbling Stones

Majid Rouhani, Veronica Farshchian, Monica Divitini

Norwegian University of Science and Technology, NO-7491 Trondheim, Norway
{majid.rouhani, veronfa, divitini}@ntnu.no

Abstract. Programming is introduced in secondary education in a growing number of subjects. This results in an increasing number of teachers teaching programming in their classes, often without proper training. Learning programming might be complicated, even more so is teaching it. In this context, there is a need to understand teachers' perspectives on teaching programming. This paper aims to identify challenges that teachers in secondary schools face and might negatively impact their teaching, i.e., stumbling stones, as well as elements that promote teaching and give motivation, i.e., stepping stones. The paper is based on the analysis of reflection notes delivered by in-service teachers attending a university-level course on teaching programming. The teachers compile the reflection notes after they complete their final project. Projects are centred around the definition of teaching plans to be tried out in class. The reflection notes of 173 students are analysed to identify issues related to: programming; teaching programming; recurrent didactic issues; and external challenges. The analysis is then summarised in a set of stumbling and stepping stones. For example, time is identified as one of the main stumbling stones by teachers. On the other side, motivation is one of the central stepping stones that we can identify in the data, often connected to the excitement of teaching something that was not previously taught in schools or that teachers perceive as highly relevant for society and the future job market. Implications for teacher training are also identified.

Keywords: Computing Education, Secondary education, In-service teachers, Challenges of teaching programming

1 Introduction

Digital competencies are recognised as an important part of education. In the context of smart environments, students need to gain not only the basic skills required to use digital tools, but also the advanced programming skills that are required to innovate through computer-based tools, as exemplified in e.g., [1, 2]. Different studies show that this training also adds to students' systematic reasoning and critical thinking abilities [3-6].

During the last years there has been a growing interest in offering activities for learning programming, both in informal, e.g. [7], and formal settings, e.g. [8]. As a result of the growing awareness of its importance, this literacy is being included in more and more national curricula.

The research question addressed by this research is: Which are, in the experience of secondary school teachers, the elements that promote teaching of programming, i.e., stepping stones, and challenges that might negatively impact their teaching, i.e., stumbling stones? To answer, we have analysed the final reflection notes of the teachers who participated in a training program at the university level. As a compulsory assignment, over a period of six weeks, teachers created teaching plans to teach programming to their students. Teachers were then asked to deliver reflection notes about their experience with creating the plan, and possibly implementing it in their class. The reflection notes have been coded to identify issues connected to (i) programming, (ii) teaching programming, (iii) didactic issues, and (iv) external challenges. In this way, the analysis is expected to capture a broad understanding of the teachers' experiences in their transition into teaching the new topic. The qualitative analysis of the delivered reflection notes is presented in this paper.

The main contribution of the paper is in providing a teacher-centered perspective of teaching programming in secondary schools, with focus on in-service teachers that are beginning to teach programming and integrating it in different subjects. The focus is on understanding the overall experience of teachers, acknowledging that these experiences might differ considerably depending on a number of local and individual factors.

2 Literature review

A growing body of literature reports on experiences with the introduction of programming in different educational systems, as for example computer science education in Israel and the United States [9]; in Brazil [10]; in the Nordic Countries [11]; in Finland, Sweden, and Lithuania [12]. A recent report on the status of Informatics education in Europe advocates the introduction of Informatics at all school levels and recommends that the teaching should be undertaken by teachers who have formal education in the subject. However, this is far from the current status in many countries, with teachers who might be teaching without having received the proper training [13]. A significant part of the duty of implementing programming education plans falls upon the educators. This is particularly problematic, considering that many of them have insufficient preparation to teach programming and computational thinking [14]. In this paper, we address the challenge of training in-service teachers to empower them to teach their students the programming skills that are critical for smart citizens.

Several challenges related to the teaching of computer programming have been reported, and teachers often do not know how to obtain adequate training and how to integrate the acquired knowledge into their educational environments [15]. Programming is a troublesome subject to learn and master [16]. It is a complicated subject that requires constant exertion, distinctive methodology, and multi-layer expertise. The way toward getting these aptitudes is a trial-and-error experimentation procedure and persistence [17]. Teachers are often left to teach something that they themselves do not feel comfortable with, considering that it is generally agreed that it takes about ten years to transform an amateur into a specialist developer [18].

Teaching principles will unmistakably influence the results of courses that educate programming [19]. Jenkins [20] explains that instructors of writing computer programs are not communicators of information like in numerous different subjects. Their essential job is rather motivating the students. The students must be roused to take part in undertakings that will cause them to learn, and it is the instructor's job to guarantee this. The world is filled with problems that require complex solutions and issues that are not reported in books and manuals. It is therefore naive to try to "drill" and "order" students, so they know the correct answers [21]. It is important to connect with students and engage them, so they assume liability for their learning and have the drive to make their future. In this perspective, it must be an instructor's primary task to guarantee that every one of their students is appropriately inspired. However, measuring or finding what drives people are not straightforward [20].

This short overview of related work clearly identifies teaching programming as an important but rather challenging endeavour, from a technical and pedagogical perspective. With our study we aim at contributing to the growing body of knowledge about teaching programming in K-12. Our focus is on teachers and how they experience teaching programming on the overall, rather than on specific technical or pedagogical issues.

3 Case and research method

Our study is connected to the in-service teacher training program offered by our university to in-service teachers who have to teach programming courses or integrating programming in other subjects. After completing the program, teachers are expected to be able to teach programming and use it in their discipline. In the following sections, we will briefly explain the context and the case being investigated in this study.

3.1 Context

The program consists of two courses of 7,5 ECTS (European Credit Transfer and Accumulation System) each, over two semesters. The first course focuses on learning basic programming and the second on teaching programming (for more information on the courses, refer to [22]). Though there are no requirements for teachers to follow both courses, most do. The study program is aimed at in-service teachers in grades 8-13 (secondary school). The program is an online study, with web-based lectures and weekly activities such as online lectures and regular compulsory work exercises. Students participate in the courses with the support of their school, which is committing to provide some free time to teachers to complete the course (one day a week), though they continue their primary duties during the two semesters. The additional costs for the schools are partly covered by a national program of the Ministry of Education. This support leads to a very high completion rate.

3.2 Course Description

The focus of this study is on the second course, aiming at helping teachers to develop the competencies needed to teach programming in their subjects, after they have acquired basic programming skills. Since most of the teachers attended the first course, the participants to our study are all expected to have at least a basic understanding of programming concepts and of Python, that is the main programming language studied in the course. The 2020 cohort consisted of 173 teacher students. Fig. 1 shows at which school level participants are teaching. More than 90% of the teachers are teaching in secondary schools, with the large majority in upper secondary schools (In the national educational system this approximately corresponds to the last three years of the K-12 system.) More than 80% of the teachers are teaching STEM-related subjects. Others teach in different subjects such as language, history, music, arts and crafts, etc. 24% of participants have already taught programming in their schools. 61% of participants had some level of programming knowledge before starting the first course. Most have been doing block-based programmings such as Scratch, LEGO Mindstorm, Arduino, micro:bit, and Blockly.

The participants are from the same country but coming from different schools and districts. All schools follow the same national curriculum. Participants vary in terms of teaching experience, including teachers who just started their practice and others who have been teaching for many years. The motivation to take these subjects courses also varies. Most say they are motivated, while others are more skeptical about the inclusion of programming in their subjects. A survey distributed at the end of the program indicates high levels of satisfaction with the course.

On the overall, the participants are diverse in terms of their background knowledge in programming, their interests in learning and teaching it, and the school level. We have not addressed demographic information in this study as we aim to include a large group with different backgrounds, needs, desires, and motivations.

To support each teacher in identifying their own learning trajectory, the course is organized around a 6-week project aiming to define and evaluate activities to be used in teacher's classrooms when teaching programming. As an outcome, teachers create a teaching plan in programming for their specific subject that they can use in their practice [22]. The project is mandatory and part of the final assessment. Teachers receive feedback from the teaching staff during the semester and must pass a first pilot phase to continue with the actual implementation.

Inputs to this project include (1) Teacher's disciplinary knowledge: Knowledge of the subjects they are teaching might help teachers in designing a teaching plan where programming is applied. Teachers can identify relevant problems to be resolved by pupils when learning to program, which may lead to increased learning outcomes in the subject; (2) Learning outcomes during the program: Teachers learn the basics of programming concepts, algorithmic thinking, and problem-solving; (3) The existing teaching plans: The teaching plans that have been created by other teachers who have completed this program in the past are also available to be used as part of creating a new and customised teaching plan; (4) External resources: Teachers can also use other resources such as teaching plans created outside of this program/course; and (5) Support from instructors: The project consists of several phases (analysis, design, and implementation). Instructors and teaching assistants support teachers during each stage by evaluating and giving constructive feedback [23]. This form of feedback may provide teachers with the information they need to improve their learning (formative assessment) [24].

Based on a predefined template, teachers start investigating the purpose and objectives of their teaching plan. Besides, instructors can support participants and give constructive feedback before implementing the teaching plan. Ovando [25] concludes in his research that constructive and meaningful feedback may lead to successful teaching and learning and personal achievement. Feedback should be deliberately given to educators and students occupied with instructing and learning [25].

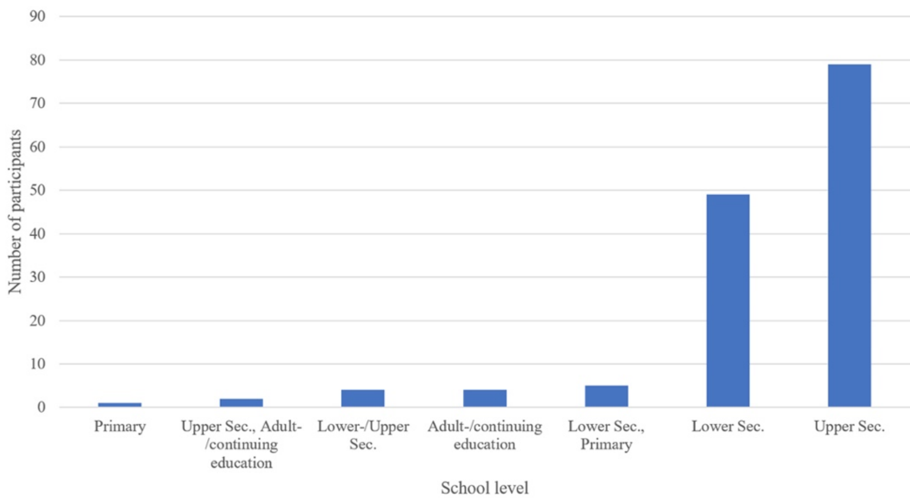

Fig. 1. School level at which participants teach

3.3 Data set

The final delivery of the project includes reflection notes. There is no predefined template for writing them. We simply ask participants to reflect on their experience during the project, including the definition and the implementation of their teaching plan. The 173 participating students' reflection notes are the basis for the study presented in this paper. The number of participants and the type of the documents resulted in a large and rich data set.

We informed students that we would use anonymised data in research. Before we started the analysis, an assistant outside the authors' team anonymised all the reflection notes.

The reflection notes' length and quality vary considerably, with some being very shallow notes and others offering very in-depth and lengthy analyses. This must be considered carefully in the study because the quality of the documents varies, and the opinion of some teachers, the ones who have written most, might be predominant. However, the advantage is that not having a template, teachers are free to focus on what they experience as relevant and essential. In this way, it is easier to capture teachers' perspectives, with less framing from the teachers and the researchers.

3.4 Method of Analysis

To address our research question, we conducted a thematic analysis of the data [26]. The analysis started with themes already identified during the analysis of a similar data set [27]. The previous study, involving a different cohort of teachers attending the same course, explored the impact of the training program on teachers' self-efficacy. Though the research questions addressed in the two studies are different, they share the overall aim of understanding challenges connected to teaching of programming and implications for professional development. In this perspective, general themes related to programming; teaching programming; recurrent didactic issues; and external challenges have been assessed as relevant also for the study presented here. In addition, new codes were added as emerging [28]. Our research is therefore based on both deductive and inductive thematic analysis.

Fig. 2. Word-cloud illustrating code frequency

Data was coded by one of the authors using Nvivo. To limit the thread to validity connected with a single coder, we added an initial phase for quality assurance. Two reflection notes were first coded together by the coder and another author, to help the coder to familiarise with the material and the existing codebook. The coder then went ahead coding 15 additional notes. At this point, the coder and one of the authors thoroughly discussed the coding and the revision of the codebook. No new themes were identified at this point, but some codes were added to avoid bias in the analysis and help capturing both negative and positive issues. After this initial phase, the coder continued with the independent coding of the material but discussing with the other authors whenever doubts arose. At the end, the coding resulted in the following distribution: external challenges with 5 codes occurring 578 times in 153 reflection notes; teaching programming with 8 codes occurring 539 times in 147; didactic issues with 5 codes occurring 429 times in 130 and programming skills with 5 codes occurring 220 times in 99. (Be aware that one participant's reflection notes could receive multiple codes and each code could appear multiple times in a document.) Fig. 2 shows a word cloud illustrating code frequency.

It is important to note that this study is mainly intended to identify trends and main issues and it does not aim at performing a quantitative analysis of the occurrence of specific codes. From this perspective, when we report the number of times a specific code is occurring, it should be mainly interpreted as an indication of the magnitude rather than an exact number.

After the coding, all the authors were involved in the analysis and discussion. In particular, one of the authors had read all the reflection notes before the discussion.

4 Analysis

In this section, we present some of the results of the study. Given length constraints, we only provide an overview of each theme. Text in italics indicates direct quotes from the reflection notes, translated from Norwegian by the authors.

4.1 Issues with programming

Adequate programming level. Many of the teachers were satisfied with the programming skills that they acquired, with 67 of them explicitly mentioning it in their reflection notes. These teachers seemed confident in their programming skills and felt they had learned enough to teach programming to their pupils successfully: *After spending a lot of time on programming in the last year, one is sitting with necessary programming skills, an understanding of what programming is, and how it happens, and one knows a little about what possibilities and challenges the course entails.*

It is also interesting to note that several teachers feel that, even if they have not reached full proficiency, their level might be "good enough." These teachers seemed to believe that they learned enough basic principles to teach pupils confidently about programming: *I feel that my code's level is within what can be achieved by students at 1P* [note by the authors: a math subject in upper secondary school] *without previous programming experience*. Some teachers expressed their concern that the pupils' increasing digital competencies in the class will soon exceed the teacher's digital capability: *One of the challenges has been to get to the right level concerning the students' competence in programming. As I mention in the actual teaching plan, most students, as of today, will have little expertise in programming. Still, it probably does not take many years after introducing subject renewal before the students can do this at a higher level than me.*

Challenges with programming. Some teachers reported challenges with programming, with the topic being explicitly addressed by 39 teachers. This often refers to making small mistakes that drastically change the results of a project: *It has taken many hours of troubleshooting and frustration over things that at first seem fine, to suddenly no longer work. During this project, I have also learned how little it takes for things not to work and how it becomes appropriate to connect everything.*

However, it is essential to underline that this is not necessarily connected to a perceived lack of skills but seen as part of the programming process: *When I have programmed the solutions, I have very often received error messages along the way because I have forgotten something, or used the wrong variable name or such simple things. What I notice about myself when this happens is that I think of this as a natural part of writing code.*

Challenges with pupils' homework. The experience that teachers gained helps them to understand better the challenges that pupils might meet. In particular, one teacher who tested the project with pupils reported that some pupils needed help and support from their parents. Still, the parents were unable to help due to a lack of programming knowledge: *Two students, who need a lot of support and regular confirmations, found it very difficult with programming and home-schooling. In the end, we gave them a different task, and we will look at the programming together when the schools reopen. These students were entirely dependent on parents sitting with them all the time, and the parents also did not know programming and Scratch.*

This poses a challenge that may be an issue for many pupils once programming becomes mainstream in school. Those who need more help while doing schoolwork might lack the necessary support to complete the program, considering that they might miss access to someone with programming skills.

4.2 Teaching programming

Teaching for the future. Many teachers (63) report that they are looking forwards to teaching programming. A vast number of teachers recognise the value of programming in schools and are looking forward to implementing them in their classes. A factor that might influence this attitude is the perceived novelty and relevance of the subject, with teachers feeling responsible for introducing it engagingly: *And ultimately, it is essential that the teacher can teach programming in a relevant, engaging, and lifelong way that engages students. Programming has come to stay in this world, on a par with other subjects we have had for decades. We create education for the future!*

Teachers' confidence. Some teachers (23) report that the knowledge and experience they have gained make them confident about their role. One teacher writes: *I can now meet the new school year with a kind of calm as I have a plan on how to start the work with programming.* Another teacher claims to be ready for the new challenges that might come with the new national curriculum: *In the autumn, we will implement new curricula in science, and there will undoubtedly be opportunities to use other methods along the way. Then I know how it can be done!*

Teachers seem to bring their pedagogical experience and approach into the new teaching of the subject, finding ways to preserve the way they see themselves as teachers, building on their previous experience: *I am mainly an active supervisor who talks to the students during the project and gives detailed and learning-promoting feedback*. Teachers are aware that they might need to acquire additional competencies, but they might still have developed the capability to do it: *Finally, I can say that these two programming courses have made me well enough equipped to complete teaching with programming in the classroom, although there is still much more to learn. In any case, I have seen new opportunities, and I look forward to using the skills I have gained in the classroom.*

External resources. External resources available online are often reported as an important source of inspiration for teachers (69): *I have, through the course, become more proven on various resources available on the Internet. I have learned about various websites I did not know about before. In this way, I feel better equipped to find good plans online when I need to create teaching plans with programming as an element in the future*. It is however, important to underline that finding, selecting, and adapting existing resources might be a challenging and time-demanding process. As explained by one teacher: *I used a lot of time looking for good examples that I could use to teach MicroPython. ...Something was easy to understand, but I had to test out things, take them apart, and put them together again into simpler programs that are easier for pupils to understand. I used a lot of time for this.*

Developing a teaching plan. Even with the available resources, some teachers (37) reported that it was challenging, but at the same time instructive and rewarding to create a teaching plan that was interesting, engaging, and suitable for the pupil's knowledge level and their own. Some teachers also encountered technical difficulties that made it challenging to complete the project. Others experienced as challenging to adapt programming to their teaching, especially for subjects where programming is not explicitly included: *One challenge was that I do not teach subjects that today say something explicit about programming, and I, therefore, had to work a little extra to try to sneak programming into the topics I have.*

In general, teachers value the capability that they have developed to introduce programming in schools, as evident in the two following examples: *I feel I have had a good repetition of some programming concepts for my part, but most importantly for my learning was to try to understand how I think the students will remember and try to imagine what challenges they will have*. Another teacher says: *The most significant learning I got when I discovered how I learned to see the possibility of connecting programming to already existing teaching programs. This made programming a positive contribution to the subject instead of an "additional burden." Incorporating the programming into teaching programs that are already in use is perceived as meaningful and useful.*

Some teachers (19) reported being worried about the teaching of programming. In most cases, they do not refer to themselves, but the perceived lack of knowledge among teachers, as evident in the two following excerpts: 1. *Many of today's teachers have little experience and knowledge in programming from their education, including me. Even though I am a relatively newly qualified secondary education teacher with a master's degree, graduating in spring 2017, my background knowledge would not be enough to teach students programming*. 2. *Many of my colleagues are unsure whether they will master providing training in a whole new skill as programming is. At my* *school I am it the only mathematics teacher with continuing education in programming. I hope that more people can use the teaching program I have created.*

Worth to mention are also some concerns about the time needed to define good teaching plans, with the need to redefine the level of ambition. As one teacher says: *I want to continue to build on the course material, with more themes and tasks. The plan was to have more content, but this is what I managed in these weeks.* The incremental approach that is suggested by teacher seems to be a good way to proceed for many teachers struggling with time limitations.

4.3 Didactic issues

Pupils' motivation. Very few teachers (3) have reported challenges in motivating and creating engagement among pupils. One teacher reports that although many of the pupils use various forms of technology, it is difficult to convince them of the importance of learning programming. They are satisfied in their roles as consumers and have low motivation for learning basic programming. Another teacher reports that students are often used to completing tasks with a single solution and are therefore struggling with programming's ambiguous nature. Another teacher said they struggled to find a balance between the pupil's independent learning and teacher guidance: *Regarding the students' learning outcomes in this project, I have asked myself the following questions: how much of the teaching should be controlled by the teacher, and how much should the students themselves be allowed to try and fail? This is a pedagogical challenge that all teachers face when planning to teach.*

At the same time, other teachers also see programming as motivating. E.g., one teacher mentioned that some boys are not interested in STEM and that they were excited to test their project because they believed it would increase their motivation: *I even imagine some boys with a low interest in science subjects, which could come out immensely strengthened from an encounter with programming. It suddenly becomes relevant in a completely different way, and if the teachers dare to let the students explore this universe themselves, it can go a long way. I'm excited.*

Adapted education. A majority of the teachers (101) reflected on adapting their teaching plan to the pupils' digital competencies in the class. This issue was mentioned in 101 different reflection notes. Most teachers were aware that pupils in the school have varying digital skills. Therefore, they created a teaching plan that would be adaptable and suitable for different competencies: *Suppose a teaching plan is to be designed around the learner's prerequisites. In that case, there is one utterly certain thing, and that is that you must include opportunities for differentiation. Students will encounter programming in school with very different prior knowledge. Some students may already master several programming languages; there will be others who have never worked with it. Although this will change in the time after introducing subject renewal, students will always need challenges at their level.*

This clearly poses a challenge to the creation of inclusive teaching plans, especially when teachers do not have a clear overview of pre-existing knowledge: *I have no knowledge of what they can of programming before.* Many teachers perceive the pupil's digital competence to be somewhat low, except for some pupils who have a particular interest in computers and programming who attempted to learn more in their free time.

Focus of teaching. The teacher's teaching plans had different objectives. What appears to be a common theme amongst many of the teacher's teaching plan (68) was to create motivation for learning programming amongst the pupils in their class. Another common focus was to implement programming interdisciplinary in the courses they were already teaching. For many, the focus was not on teaching programming as its own subject, but rather to understand how it could be used as a resource in other classes: *Our goal was to create a teaching plan for the students that create engagement, arouses interest in programming, and links it to other subjects the students have.*

Assessment of pupils. Very few teachers (15) reflected explicitly around pupils' assessment. This might be because few teachers had the opportunity to test the teaching plan with their pupils. Another reason might be connected to the curriculum's unclarity regarding the learning objectives of programming when integrated into other subjects and the uncertainties related to the upcoming introduction of a new national curriculum: *I have not set up detailed criteria concerning what the students should be able to do in programming. I have deliberately done this as it does not say anything about this in the curriculum. But I interpret the curriculum so that one can use programming as a tool to explore in mathematics. This is also why I have chosen that the assessment situation should be a presentation of what the students have done.*

Formative assessment seems to be an essential element, with some of the teachers concerned about pupils getting continuous feedback about their work and progress. Some teachers also underline the importance of pupils taking an active role in their learning. This relates to their learning process and their participation in shaping the learning activities, as in this example: *As a teacher, I (and we) are concerned with assessing learning and not with learning. Therefore, the task was not just to make something, but also to show something about the process along the way. I think it is important that students think about their role in learning. The student's voice is important, and I think the surveys that were conducted were good tools for mapping how we should set up the teaching in joint sessions, group sessions, and individual sessions. I could also catch those who had problems along the way and who did not say this directly. I think it worked well, and I was motivated to do the best for the students to come as far as they could in the time we had available. The last week was set up for the students to present the result, one by one, and have a conversation with a teacher, where they received feedback on the work.*

Cooperation and Community. Many teachers reported collaborating with other teachers (76) and collaborating with and among pupils in their class (68). Here we include (a) teachers collaborating with other teachers in the course; (b) collaborating with other teachers at their place of employment; and (c) collaboration with and among the pupils in their class.

Due to the ongoing pandemic, many teachers had not been able to test their projects and were, therefore, unable to reflect on the teaching plan's success. This made even more critical cooperation in the creation of lecture plans and their evaluation: *With the students' feedback, it could have been easier to adjust the plan along the way. Therefore, I have spent a lot of time discussing the teaching plan with* *my partner, so it has helped me work in pairs*. Some teachers created their teaching plan with colleagues from their school in mind: *It has been a goal that other teachers can use the teaching program, and it was therefore important that the description shows both what we should do and why! The rationale is central for the teacher to use the teaching program as a bridge between different representations of growth speed, thus helping the students increase their mathematical understanding*.

Many teachers also recognised that collaboration between students in the class was crucial: *Being able to use fellow students in discussions also provides more learning for the student. They must precisely explain each other, use correct subject terminologies, and not least have problems described by more than one person (teacher). All this can give students motivation to create something on their own and/or together with others*. Another teacher says, *I have also thought that it would be wise to organise the students in learning pairs in this programming task. There is a lot of learning in that students can discuss possible algorithms with a learning partner. They then get new ideas and reflect along the way in the work. They also dare to try more when working together in pairs*.

While many teachers recognise the importance of cooperation among pupils, it should be considered that this requires time and must be properly integrated in the planning. As explained by one teacher *My experience with this project is that programming takes time. The pupils need to get enough time to familiarise with the problem definition, to discuss the solutions, to discuss how a program must look like and answer the problem. I must set aside time enough so that pupils can discuss and reflect, not only programming*.

An interesting aspect is inter-generational collaboration. Many teachers reported that they got help from people from younger generations, such as their children or pupils. They said that they had learned a lot from more youthful individuals: *The actual maneuvering and how to build and destroy blocks and some commands I learned from my two sons*. This shows that young people may be an untapped resource for the teachers who plan on teaching programming.

Gender gap. An underwhelming number of teachers (7) mentioned the gender gap in programming in their reflection notes. Very few of the teachers who mentioned it reflected on ways to reduce this gap or make their teaching plan more appealing to girls, but mostly they acknowledged the issue. One female teacher explicitly mentioned that she aimed at being a role model for her pupils, while another teacher

reported designing a project, an automatic greenhouse, that could be more attractive for girls

One teacher said that she was a female science teacher and wanted to use her position to motivate her female students to learn to program: *First, I have become more proven in my role in teaching. As a female science teacher, I want to inspire and motivate more girls to choose technology education. Simultaneously, I want to use the position I have to show that programming is not just for the guys, possibly some nerdy stuff that not everyone can. If the threshold for starting is just low enough, most people will experience mastery in the beginning*.

One teacher mentioned that they created a teaching plan based on topics and themes of interest to girls: *One of my main goals is to get more girls interested in programming. In a preliminary study, several girls showed greater interest if the programming could be linked to biotic factors such as exercise, diet, and food. This* *gave me the idea to create a fully automatic greenhouse in the school's basement where Arduino devices monitored light, fertilisation, and irrigation.*

One teacher who had the opportunity to test their teaching plan in their high school 'higher math class experienced that the girls in their class, many of whom were academically strong in the subject, found programming to be challenging: *Part 2 of the teaching plan was tested on an R1 class due to the Coronavirus and the schools closed. I knew it would present some challenges, but this was instructive for me. We had two sessions (2 and 3 hours), where we worked on text programming with Python. In these sessions, I saw how big a difference is in the students' starting point for learning programming. What perhaps surprised me the most was that some students, who are usually the best (especially girls), did not do so well. They are conscientious and have the right work strategies, but I observed that they became more restrained and found this difficult in the second session.*

The same teacher also reported that the girls showed low interest and motivation in programming and struggled to partake in class discussions about the new teaching plan. The teacher believed that this might be because they focus on their academics and fail to show interest in things that they perceive not to be related to or important to the subject: *One of the nice things was that we had some good discussions about my scripts, and the students showed how they thought the program could be set up. But in this discussion, some of the students and especially some of the girls became a little absent. It may be that some of the schoolchildren's interest was not so great to learn this since it is not part of the curriculum in R1 this year, and they instead prioritise learning the curriculum in which they will be assessed.*

4.4 External challenges

In the analysis of the reflection notes, we identified several external challenges, i.e., challenges that the teachers did not perceive as related to their programming or pedagogical skills. These are all issues that are, to a large extent, outside the sphere of inference of individual teachers.

Use of programming in other subjects. Many of the teachers (57) reflected on teaching programming's challenges integrated into other topics. Some teachers noted that the math used in programming sometimes was challenging for the pupils to understand, requiring the teacher to consider yet another variable in adapting their project: *The students' varying prior knowledge in mathematics will also be essential for the program's educational approach, most of the assignments contain both mathematical algorithms and concepts.*

Some teachers also reflected positively towards using programming in math, possibly increasing the pupil's interest in math, as in the following example: *Through the work with this teaching plan, more people will hopefully get some mathematical "aha-experiences." Numerical methods open up a new field in mathematics that students have not seen before. This will probably be engaging for many students. They know that programming can be the fastest (or only) way to solve some mathematical challenges.*

COVID-19. COVID-19 was the most significant external challenge for this year's teachers (95). All schools were shut down, and all lectures were transferred online.

This confusion prevented most of the teachers from completing their teaching plans in class with their pupils. Due to this, it was difficult for teachers to reflect on their project and how it would work in practice. Some teachers lost access to necessary school resources due to the schools shutting down. Some teachers also reported that they struggled personally with adapting to the situation, and the same was for their pupils. Changing to online schooling as a result of COVID-19 was somewhat tricky and time-consuming. This caused a lot of teachers to experience a lack of time to work on new things.

Time constraints. Seventy-eight teachers mentioned issues connected to time in their reflection notes. Some issues have already been reported discussing, e.g., challenges with programming, relating to the time required to get things working, both for themselves and when planning activities in class. Time-related issues have also been reported in connection to pedagogical issues. Here we identify additional issues connected to elements that are largely outside the control of individual teachers. For example, one teacher expressed concerns about the time that has been allocated for programming in school and the connected challenges: *It is difficult for me to see how I can manage to teach to my students a traditional programming language in the little time that I have available in my subject, when they come to my class without previous knowledge*. A similar concern is also expressed by another teacher that says: *The first thing that struck me is that I cannot teach everything I have learned and experienced, without spending a lot of time....Time is a scarce resource in the classroom.*

Technical issues. Some teachers (30) did stumble upon some minor technical problems, though they were mostly resolved quickly. This also inspired the teachers to revise their activities to make them more robust. For example, one teacher states: *After many runs of the system, it did not always work, which created great frustration, I have been made aware that this built-in temperature sensor in the micro bit can register a little high temperature. This is because it is located inside the processor, and it is hot. In a proper implementation with students, I will have an extra thermometer on the teacup so that one can take control measurements of the temperature.*

School resources. Some teachers (20) mentioned access to school resources in their reflection notes. Most of the schools seem to supply the necessary resources to implement the teaching plan. One teacher noted that their municipality-provided teachers with Chromebooks but that they were not allowed to download apps on it, which was challenging: *The municipalities I work in have decided that we should only have Chromebooks and that no apps should be installed. Therefore, we need to find useful web resources that can be used on Chromebooks.*

5 Stumbling and Stepping stones

After the analysis, we categorised the identified issues into stumbling and stepping stones, based on whether they are perceived as impacting negatively or positively on teachers' experience.

5.1 Stumbling stones

Covid-19 lockdown is clearly emerging as a stumbling stone encountered by this cohort of teachers. This has impacted teachers' capability to implement their teaching activities as planned and created a general stressful situation. However, this is not the focus of the paper, where we instead aim to identify recurrent issues independently by the peculiarity of 2020.

Time is one of the main stumbling stones emerging from the analysis. This relates first and foremost to the problem for teachers to find time for completing their project. Even for this group of teachers, who have got official time for their school training, time was scarce. We can expect to find time for training to be even more difficult for other teachers. The paucity of time is a significant concern because it impacts teachers' capability to up-skill, keep their competencies updated with technological and pedagogical innovations, and ultimately create meaningful and engaging learning activities for their pupils.

Concerns about time are also connected with the implementation of activities in schools, especially in light of formative assessment and continuous support to pupils.

A number of teachers also reported challenges with programming, often requiring more time than expected to fix bugs and getting things working. This frustration might then turn into a lack of motivation [20].

This last point also relates to the perception of programming being difficult, in line with existing literature [14, 16, 17, 29, 30].

Developing programming skills and teaching programming are, not surprisingly, often intertwined in the perception of teachers. *Competence about the subject matter* is essential to give the teacher the confidence to teach [14, 30]. However, teachers might feel comfortable and do an excellent job also with limited programming skills as long as they can embrace the uncertainty and perceive that they know how to look for the missing information and seek for help.

Another possible stumbling stone is *multidisciplinarity*. Though teachers are not necessarily negative, many express concerns connected to the definition of relevant projects, the required knowledge for pupils in different subjects, and issue connected to meeting different learning objectives. Another concern is emerging when we get an eye-bird perspective of the teaching plans developed by the teachers. A majority of the teachers created interdisciplinary programming projects for STEM classes such as higher math, physics, biology, and chemistry. This might be problematic in terms of "Programming for All" [31] since it cuts out Arts and Humanities.

Another challenge worth to mention is the varying level of knowledge and motivation among students. Many teachers acknowledge the need to adapt their teaching to individual pupils. This is clearly a stepping stone for pupils, who can get tasks and scaffolding adequate to their needs. However, *adapted education* might turn into a stumbling stone for teachers since this requires additional time and competencies both in the planning and in the implementation of any teaching activities.

5.2 Stepping stones

Motivation is one of the main stepping stones that we can identify in the data, often connected to the excitement of teaching something that was not previously taught in schools or that teachers perceive as highly relevant for society and for the future job market. Even teachers who might be struggling themselves with programming, seem mostly be motivated to teach it and learn more, overcoming the challenges that they might encounter. This motivation is also reflected in the way many teachers look at their projects, striving to creating activities that are engaging, innovative, and inclusive. Hopefully, this can create a virtuous cycle that can benefit the motivation of both teachers and pupils, sustaining it in time.
Another stepping stone connected to motivation is the application of programming in different disciplines. Teachers are motivated to learn and teach programming when they see the benefit of applying programming in their subject areas to increase pupils' subject understanding. Also, the pre-existing confidence in their main subject might help them to deal better with the uncertainties of programming.

Collaboration and community participation are the other major stepping stone for many teachers. As presented in our analysis, this includes: (i) the cooperation with peers, (ii) the cooperation with students, and (iii) the use of resources from different communities as a starting point for their projects.

6 Turning stumbling stones into stepping stones

Based on the results and the previous discussion, in this section we draw some implications for teacher training. Before starting, we want to underline that the results show that a stepping stone for one might be a stumbling stone for another. Even Covid-19 was seen by some teachers as an opportunity to discover new tools or to find new forms of cooperation with their peers. This does not intend to underestimate the challenges that teachers meet that are concrete and legitimate given their context. The main point here is that teacher training should focus on giving teachers the conceptual tools and skills to address known challenges and limit their impact on their practice.

The course that we studied supports flexible learning trajectories, i.e. teachers can focus on different topics and choose how they intend to teach programming to their students. When defining their projects, they can select relevant subjects, the appropriate level of complexity, and align this with the learning objectives. This seems to be a promising approach that gives motivation. When struggling with time, working with something that one perceives as relevant and directly usable in class is important. However, data shows that teachers choose themes related to the subjects they teach. Since most of the teachers in this study program teach STEM subjects, the teaching plans are also associated with them. As previously discussed, this might influence negatively the development of inclusive programs that would benefit from a higher level of multi-disciplinarity. This should be addressed both at the policy level, when deciding which teachers should get access to training or admission criteria and the existing programs, for example promoting awareness of projects bringing together different subjects outside STEM.

For teachers, the process of learning to program themselves and teaching it to others often goes hand in hand. This is extra demanding because programming in itself is complicated and will take years before one can become steady in programming. Therefore, it is essential during training to increase teachers' awareness about learning as a continuous process, and that their skills must be continuously maintained [21], also for experienced teachers [32]. This is challenging since time is one of the main constraints. Still, if teachers take small steps and are concerned with creating engagement and interest in students [33], it can lead to students who become more autonomous and more active in the learning process [18].

The availability of online resources is clearly an advantage that many teachers are building on. However, for other teachers it is overwhelming. Training of teachers might consider this issue by explicitly providing information about reliable resources and about how to search and adapt them. This issue might be seen under the broader umbrella of cooperation and community building. Teacher training should explicitly address collaboration and community participation, identifying successful forms of cooperation in different settings. Some teachers explicitly mentioned the challenges connected to the lack of peers they can relate to. We can expect this to be particularly serious for small and rural schools. Learning how to create connections among teachers outside owns school [34] is therefore paramount. In this context, we want to underline that large international communities of teachers have been only marginally mentioned. Language might be a limiting factor, together with differences in national curricula. Training should therefore include awareness of local and international communities, different forms of participation, and promote a culture of sharing, where teachers are not only consumers but also providers of learning material and expertise. This is important because it might motivate teachers and, at the same time, help with sustaining communities. Training should promote participation in local communities, i.e. with teachers sharing same national curricula, language, and school culture, as well as in global communities to confront different perspectives and promote a reflective practice. It is however important to point out that some studies show that, despite the documented benefits of collaborative learning, many online instructors and students avoid it due to the time and commitment it requires [35]. It is therefore important to create among teachers awareness of the potential benefits and help them to develop the right skills for a fruitful collaboration.

Our data show the importance to promote self-efficacy, i.e. the belief of a teacher in their capabilities to teach the subject [36]. Improving teachers' self-efficacy requires working on teachers' competencies and giving them better control of external challenges. We know that learning to program and teaching programming is demanding [16]; therefore, preparing a teaching plan may affect teachers' self-efficacy. By developing teaching plans supervised by course instructors, teachers may be better equipped to start teaching programming for their students. In addition, it might be useful to make teachers aware of challenges that they might face and heuristics to address them. This requires however a good understanding of the specific needs of the teachers. For example, as already mentioned, the possibilities of cooperation with peers might be very different in small rural schools from large urban schools. As another example, the teachers in our cohort have not reported major issues with availability of resources. This might however be a major challenge in other contexts, as reported e.g., in [37].

There are two missing stones in the data, despite we have been explicitly looking for them. Gender issues are only marginally mentioned by a very limited number of teachers. Considering the gender gap in computing education [38], it is worrying that this is an issue that is not high in the teachers' reflections. Assessment is another issue that is appearing rarely in the analysed documents. Again, given the importance and challenges connected to assessment in computing education [39, 40], it is a concern. These two aspects should be addressed more explicitly in teacher training to increase their awareness.

7 Conclusion

This paper presented the results from the analysis of the reflection notes delivered by 173 in-service teachers attending a university-level computing education program. In this study, we aimed at identifying challenges and enablers connected to teaching programming in secondary schools. Though some of these challenges and enablers have already been identified in previous literature, our main contribution consists of providing a holistic approach that looks at stumbling and stepping stones together and how they might influence policies and teacher training. This approach also underlines the importance of looking at teachers individually and as part of a complex ecosystem that affects how they operate, bringing opportunities and challenges.

We are fully aware that the study focuses on a particular group of teachers, i.e., the ones who have been given by their school the time to improve their teaching. Therefore, their perception of challenges and possibilities is impacted by the course. On one side, this is an interesting group because it allows identifying stepping and stumbling stones core to learning and teaching programming, even when this happens in optimal conditions, i.e., with allocated time and expert support. Simultaneously, regular in-service teachers not receiving any training in programming might face additional challenges that are not identified in a study with this specific population. As part of future work, we intend to study the broader population of in-service teachers to investigate the extent to which the identified stumbling and stepping stones might be generalised. It should also be noted that we have not connected the reflection notes to demographic information, mainly for privacy reasons. In future studies, it might be interesting to study how the teacher's profile, e.g., age and experience, might influence the results. It might also be interesting to study another didactical issue that teachers discovered during the implementation of their teaching plan: the importance of collaboration through pair programming. Another area that might need more attention in future research is challenges with pupil' homework when parents cannot help due to lack of programming knowledge. The same issue may apply when learning other subjects where parents are not proficient and cannot help children. Do teachers consider the difficulties with learning programming to be more challenging than those met when learning languages?

Another limitation of the work is that we analysed reflection notes where teachers decide what to focus on and the time they have to complete the task. This offers the advantage of providing a direct perspective on the teachers' point of view and what they value as relevant. Simultaneously, the fact that something is not mentioned does not necessarily imply that teachers do not think about it. To overcome this limitation, we are planning interviews with the teachers to collect a complementary data set. Finally, studies across countries would be relevant to understanding how the educational system impacts teachers' experiences.

Acknowledgments. We thank all the teachers who attended the course and participated in the study. The work is partly funded by Excited, The Norwegian Center for Excellent IT Education (https://www.ntnu.edu/excited).

References

1. Jaskiewicz, T., et al., Hacking the hackathon format to empower citizens in outsmarting “smart” cities. Interaction Design and Architecture(s) Journal 2019. 43: p. 8-29.
2. Poplin, A., L. Shenk, and U. Passe, Transforming Pervasive into Collaborative: Engaging Youth as Leaders with GIS through a Framework that Integrates Technologies, Storytelling, and Action. Interaction Design and Architecture(s) Journal 2017. 35: p. 182-204.
3. Clements, D.H. and D.F. Gullo, Effects of computer programming on young children's cognition. Journal of Educational psychology, 1984. 76(6): p. 1051-1051.
4. Duncan, C., T. Bell, and S. Tanimoto. Should Your 8-Year-Old Learn Coding? New York, NY, USA: Association for Computing Machinery.
5. Jeannette, W., Computational Thinking - Jeannette Wing. 2008.
6. Dağ, F., Prepare pre-service teachers to teach computer programming skills at K-12 level: experiences in a course. Journal of Computers in Education, 2019. 6(2): p. 277-313.
7. Motschnig, R., et al., When Kids are Challenged to Solve Real Problems – Case Study on Transforming Learning with Interpersonal Presence and Digital Technologies. Interaction Design and Architecture(s) Journal 2017(34): p. 88-111.
8. Cools, S., et al., The Diorama Project: Development of a Tangible Medium to Foster STEAM Education Using Storytelling and Electronics, in Citizen, Territory and Technologies: Smart Learning Contexts and Practices 2017, Springer. p. 169-178.
9. Gal-Ezer, J. and C. Stephenson, A tale of two countries: Successes and challenges in K-12 computer science education in Israel and the United States. ACM Transactions on Computing Education (TOCE), 2014. 14(2): p. 1-18.
10. Santos, P.S.C., L.G.J. Araujo, and R.A. Bittencourt. A mapping study of computational thinking and programming in brazilian k-12 education. in 48th Annual Frontiers In Education Conference. 2018. IEEE.
11. Kjällander, S., et al., Makerspaces across settings: Didactic design for programming in formal and informal teacher education in the Nordic countries. Journal of Digital Learning in Teacher Education, 2018. 34(1): p. 18-30.
12. Dagiene, V., et al. Students' performance on programming-related tasks in an informatics contest in Finland, Sweden and Lithuania. in ITiCSE '14: Proceedings of the 2014 conference on Innovation & technology in computer science education. 2014. ACM.
13. CECE, Informatics Education in Europe: Are We All In The Same Boat? 2017, ACM.
14. Cooke, L., et al., Can Everyone Code?: Preparing Teachers to Teach Computer Languages as a Literacy. 2020, IGI Global. p. 163-183.

15. Hiltunen, T., Learning and teaching programming skills in Finnish Primary Schools--The potential of games. University of Oulu, Oulu, 2016.
16. Cheah, C.S., Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review. Contemporary Educational Technology, 2020. 12(2): p. ep272-ep272.
17. Jiau, H.C., J.C. Chen, and K.-F. Ssu, Enhancing self-motivation in learning programming using game-based simulation and metrics. IEEE Transactions on Education, 2009. 52(4): p. 555-562.
18. Winslow, L.E., Programming pedagogy - A psychological overview. ACM Sigcse Bulletin, 1996. 28(3): p. 17-22.
19. Robins, A., J. Rountree, and N. Rountree, Learning and teaching programming: A review and discussion. Computer science education, 2003. 13(2): p. 137-172.
20. Jenkins, T. Teaching programming--A journey from teacher to motivator. in 2nd Annual conference of the LSTN Centre for Information and Computer Science. 2001.
21. Liu, W.C., J.C.K. Wang, and R.M. Ryan, Understanding motivation in education: theoretical and practical considerations. 2016, Springer. p. 1-7.
22. Rouhani, M., et al. Design of a programming course for teachers supporting flexible learning trajectories.
23. Du Toit, E., Constructive feedback as a learning tool to enhance students' self-regulation and performance in higher education. Perspectives in Education, 2012. 30(2): p. 32-40.
24. Heritage, M., Formative assessment: What do teachers need to know and do? Phi Delta Kappan, 2007. 89(2): p. 140-145.
25. Ovando, M.N., Constructive Feedback. International Journal of Educational Management, 1994.
26. Braun, V. and V. Clarke, Using thematic analysis in psychology. Qualitative Research in Psychology, 2006. 3(2): p. 77-101.
27. Thorsnes, J., M. Rouhani, and M. Divitini. In-Service Teacher Training and Self-efficacy. in International Conference on Informatics in Schools: Situation, Evolution, and Perspectives. 2020. Springer.
28. Saldaña, J., The coding manual for qualitative researchers. 2015: Sage.
29. Wang, X.-M., et al., Enhancing studentsâ ™ computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt. Journal of Educational Technology & Society, 2017. 20(4): p. 58-68.
30. Hotaman, D., The teaching profession: knowledge of subject matter, teaching skills and personality traits. Procedia-Social and Behavioral Sciences, 2010. 2(2): p. 1416-1420.
31. Guzdial, M., Learner-centered design of computing education : research on computing for everyone. Morgan & Claypool Publishers. 147-147.
32. Savec, V.F., Use of ICT and innovative teaching methods for STEM. Innovative Approaches to the Application of Digital Technologies in Education and Research SLET-2019, Stav-ropol-Dombay, Russia, 2019: p. 1-10.
33. Jenkins, T. Teaching programming--A journey from teacher to motivator.
34. Ferk Savec, V., Use of ICT and innovative teaching methods for STEM. 2019.
35. Zygouris-Coe, V., Collaborative learning in an online teacher education course: Lessons learned. ICICTE Proceedings, 2012: p. 332-342.
36. Zee, M. and H.M.Y. Koomen, Teacher self-efficacy and its effects on classroom processes, student academic adjustment, and teacher well-being: A synthesis of 40 years of research. Review of Educational research, 2016. 86(4): p. 981-1015.
37. Sentance, S., L. Singh, and P. De Freitas. Challenges Facing Computing Teachers in Guyana. in SIGCSE '20: Proceedings of the 51st ACM Technical Symposium on Computer Science Education. 2020. ACM.
38. Klawe, M., T. Whitney, and C. Simard, Women in Computing---Take 2. Commun. ACM, 2009. 52(2): p. 68-76.

39. Klawe, M., T. Whitney, and C. Simard, Women in Computing---Take 2, in Commun. ACM. 2009, Association for Computing Machinery: New York, NY, USA. p. 68-76.
40. Grover, S. and R. Pea, Computational thinking in K--12: A review of the state of the field, in Educational researcher. 2013, Sage Publications Sage CA: Los Angeles, CA. p. 38-43.

Paper V

M. Rouhani and V. Jørgensen, 'In-service teachers' attitude towards programming for all,' in *Ludic, Co-design and Tools Supporting Smart Learning Ecosystems and Smart Education*, Springer, 2022, pp. 149–162

Chapter 12
In-Service Teachers' Attitude Towards Programming for All

Majid Rouhani and Victor Jørgensen

Abstract Coding instruction has increased widely throughout primary and secondary education in many countries, and educators are just beginning to understand the complexities of teaching students to code. A question raised is whether everyone should learn to code to be fully literate participants in our future society? What are the teachers' main arguments concerning the concept of programming for all? We investigate these questions from a teacher's perspective and aim to determine what attitudes teachers have towards programming for all. We gave teachers a task to describe their thoughts and perceptions on programming for all and collected data during a programming course for in-service teachers. This paper reports on preliminary findings. Although the vast majority in this study have positive attitudes towards programming, we can also see negative attitudes. These concerns are mainly related to lack of time. In-service teachers in this study believe that programming can be fun and engaging. They come with many arguments on reasons why they should include programming in school.

Keywords Online course · In-service teachers · Programming for all · Attitudes towards programming

12.1 Introduction

The trend to compose programming into the educational program will keep developing, grow, be driven through corporations like Code.org [1], Apple and Microsoft [2]. Youth are remixing, adjusting and making their applications, programming and computer games [2]. Programming is challenging to learn and teach and is a complicated subject that requires multilayer expertise [3, 4]. Teachers are often left to teach something that they do not feel comfortable with, considering that it may take about ten years to transform an amateur into a specialist developer [5]. On the one hand, we have the growing trend to include programming in schools and, on the other hand,

M. Rouhani ( ) · V. Jørgensen
Norwegian University of Science and Technology, 7491 Trondheim, Norway
e-mail: majid.rouhani@ntnu.no

Ó. Mealha et al. (eds.), *Ludic, Co-design and Tools Supporting Smart Learning Ecosystems and Smart Education*, Smart Innovation, Systems and Technologies 249,
https://doi.org/10.1007/978-981-16-3930-2_12

the complexity and challenges of learning and teaching programming that may put teachers in a demanding and challenging situation. In this study, we take a closer look at teachers' perspectives on 'programming for all' and ask the following research questions: Why should everyone learn to code to be fully literate participants in our future society? What are the in-service teachers' main arguments concerning the concept of programming for all? We investigate what attitudes in-service teachers have towards programming for all. This involves classifying the arguments as positive or negative attitudes. Many people need to be computer literate and sometimes use computational thinking (CT) to solve problems. Some of them will need to use programming at some point. This paper emphasizes that need and uses the words of teachers to make that case. The paper contributes to the body of knowledge around in-service teachers' training to meet the growing demand for teachers being prepared to teach programming in different subjects and school levels. In this study, we investigate in-service teacher's attitude participating in a university-level course. This group is diverse; some with no prior programming knowledge, some are already teaching programming and others intend to include programming as part of other disciplines.

We have organized this paper as follows: The next section presents a literature review on related work. In Sect. 12.3, we introduce the case and explain our methodology. Section 12.4 analyse the quantitative data around the main themes identified in the data set. Section 12.5 discusses the main findings in this study, and Sect. 12.6 present the conclusions and further work.

12.2 Literature Review

To improve the quality of lessons and teach effectively, Makhmudov et al. [6] suggest that teachers should have the ability to read and write simple computer programs; the ability to use computer programs and educative documentation; the ability to use computer terminology, particularly as it relates to hardware; the ability to recognize educational problems that can and cannot be solved using the computer; the ability to locate information on computing as it relates to education; and the ability to discuss the moral and human-impact issues as they relate to the societal use of computers as well as the educational use of computers. Recent efforts to revitalize the importance of CT aim to democratize computing knowledge as an essential body of knowledge that learners need to have to cope well with the twenty-first century [7]. CT is solving problems, designing systems and understanding human behaviour by drawing on the concepts fundamental to computer science [8]. Researchers and teachers foster CT through the use of programming [2]. Programming involves planning, testing, debugging and improving the source code [9]. Both concepts (programming and coding) are common in education, and there are advantages and disadvantages to using these words as synonyms. To investigate these research questions, we need to understand the teachers' perceptions of the underlying concepts such as computer literacy, CT, programming and coding, and motivational beliefs and attitudes.

12.2.1 Why Should Everyone Learn to Program?

Students need to be familiar with the fundamentals of programming a computer regardless of whether they intend to become computer programmers [10]. Coding as essential literacy is fundamental, and everyone should learn to code to be fully literate participants in our future society [11, 12]. Many researchers believe that engaging in programming tasks has unique potential in promoting higher-order thinking performance since it can enable students to obtain programming knowledge [13, 14]. Computing is defined as the act of using computers to gather, calculate, analyse, store, protect and present information faster and more accurately than can a human, and most often, less expensively [15]. Hence, programming is an essential part of computing. One argument for teaching everyone about computing is that we need more workers to program (jobs). Computing is part of students' lives; they live in a computational world. Computational thinking, computing literacy, productivity and broadening participation are other arguments used [14]. Including necessary coding in primary curricula provides teachers with an effective means of exercising their students' general and higher-order thinking skills [16]. To effectively promote students' learning performance in programming courses, it is necessary to include proper teaching strategies [17]. In this study, we take a closer look at teachers' perspectives on 'why everyone should learn to program'.

12.2.2 Motivational Beliefs and Attitude

Passionate views and perspectives towards a school subject are pertinent for learning and instructive achievement [18]. Malmi et al. [19] suggest that an improved understanding of the complex factors (such as the academic and social capital [20]) influencing students' internal factors, such as beliefs, emotions and attitudes, would help us develop more effective interventions to influence students' perceptions about learning to program and improve pass rates and learning results. Motivational beliefs in science have been shown to predict academic achievement reliably. These beliefs reflect the motivational value a person accredits to a subject or task and named value beliefs. Positive motivational opinions are associated with students' persistence in performing well even when their interest and intrinsic enjoyment decrease [21]. Besides attitude, self-efficacy perception also plays a role in the students' success in a computer programming course. Self-efficacy refers to the trust in the skill that the individual has to perform work [22]. In-service teachers often must teach programming as part of other subjects. Introducing programming in addition to the topics they already teach may complicate the situation for them. Therefore, motivational beliefs and positive attitudes are essential aspects of a programming course for this group of teachers. The question is whether teachers find it meaningful that everyone should learn programming or perceived as a disruption in teaching their subjects?

Meaningful information is bound to be remembered because it effectively interfaces with propositional networks. Meaningfulness saves time and promotes learning [23]. In this context, there is a need to understand teachers' attitudes on teaching programming to all.

12.3 Case and Research Method

12.3.1 The Case: A University Level Programming Course for Teachers

Our university provides an online programming course for in-service teachers who teach programming courses or integrate programming into other disciplines. We have divided the program into two subjects, each of 7.5 credits and adapted the program for teachers who teach in grades 8–13 (secondary school). These courses are 'basic programming for teachers' and 'applied programming for teachers' and use Python as the programming language. After completing the program, we expect teachers to teach programming and use it in their subject area. Teachers participate with their school's support, committing to providing some free time to teachers to complete the course, though they continue their primary duties during the two semesters. The Ministry of Education partly covers the schools' additional costs. We conducted a pre-course survey to figure out what different levels and subjects the teachers are teaching. We received 186 responses (a rate of 93%). 78% are teaching at upper secondary school, and 20.4% teach at lower secondary school. More than 80% of the teachers are teaching STEM-related subjects. Others teach in different topics such as language, history, music, arts and crafts. 24% of participants have already taught programming in their schools. 61% of participants had some level of programming knowledge before starting the first course. The participants all come from the same country, but they represent various schools and districts. All schools follow the national curriculum. Participants have varying levels of teaching experience, ranging from new teachers to those who have been teaching for many years. The incentive to enrol in these subjects' courses varies as well. The majority claim they are inspired, although others are less certain that programming should be included in their classes. We have not addressed demographic information in this study as we aim to include a large group with different backgrounds, needs, desires and motivations. The first subject curriculum covers basic programming knowledge, while the second subject covers different aspects of how to teach programming. When teachers start the second course, they start working on definitions of some important concepts related to computing education, e.g. 'why should everyone learn to code' and 'what are the expected challenges'. Teachers use approximately three weeks on this part of the course. Before continuing to other topics, they deliver an assignment to reflect on their attitude towards programming for all and teaching programming. We informed teachers that anonymized data will be used in research and had the option of not being included. The question we asked participants was: 'why should everyone learn to code?'

12.3.2 Research Method

Data Collection. We collected data from participants' reflection notes from the 2019 and 2020 cohorts to answer our research questions. We analysed a total of 328 reflection notes. For 2019, there were 22 out of 70 notes available for data analysis. For 2020, 169 out of 170 was used at the start and 137 at the end of the course. The main themes identified and discussed in this paper are: 'arguments for why everyone should learn to program, or why we should teach programming in school', and 'the attitude of teachers towards programming for all'. The research team processed the document's qualitative data in several steps:

- Documents were anonymized outside of the research team. Thus, a collection of reflection notes for the period 2019 and 2020 was available to the research team for further analysis.
- Researchers have an in-depth knowledge of the course. They are involved in developing the program and worked as teacher assistants. Hence, well known to the reflection notes in this study.
- The analysis started with themes already identified during the analysis of a similar data set [24]. The previous study, involving a cohort of teachers attending the same programming course, explored the impact of the training program on teachers' self-efficacy. Though the research questions addressed in the two studies are different, they share the overall aim of understanding challenges connected to programming and implications for professional development.
- To limit the thread to validity connected with a single coder, we added an initial phase for quality assurance. Few reflection notes were first coded together and discussed to familiarize the coder with the existing codebook. The coder then went ahead, coding additional notes.
- Researchers coded few documents independently to check inter-coder reliability. The method 'Percent Agreement for Two Raters' was used. We manually compared the codes and gave the score of 1 to those that are the same and 0 to those that are different. The number of 1s divided by the total of codes compared will be inter-rater reliability. The results from the inter-coder reliability check (two times during the study) were considered sufficient ($\approx 88\%$ in average).

Limitations and Threats. By analysing teacher's reflection notes, we looked for their perspectives on 'programming for all'. To ensure high validity, we used a significant sample size (328 reflection notes for two years) for this study, and the teacher's reflected without being directed by any questionnaire. Participants differ in terms of the subjects they teach, school type and motivational factors. We used a pre-existing codebook from a previous study involving a cohort of teachers attending the same programming course, which explored the impact of the training program on teachers' self-efficacy. Based on the research questions, the research group added new codes. All researchers in this study have an in-depth knowledge of the course and well known to the reflection notes being analysed. In-service teachers' attitudes and arguments may have been affected by the literature study in the course. Most participants are already aware of the benefits and importance of learning programming. Results are mainly from one course on programming for teachers and their experiences, and it might not be easy to generalize the results. Researchers need to investigate how programming is taught to this group at other institutions and countries compared to this approach.

12.4 Analysis

The summary of the data gathered is shown in Table 12.1. Reflection notes collected at the beginning of the course are related to the question: 'why should everyone learn to code?' This is the case both for 2019 and 2020. At the end of the course, many participants reflected on these questions without explicitly asking them to do so. A teacher may have several complementary or opposing reasoning. In this analysis, we only look at the number of arguments and not which teacher these belong to. In the following, we will give a brief explanation for each category and some of the results.

Table 12.1 Overview of themes—at the beginning and end of the course

	Year	Theme	# teachers	# code references
Start	2019	Arguments for learning to program	19	25
Start	2019	Arguments for not learning to program	4	12
Start	2019	Critical towards programming in school	2	5
Start	2019	Positive towards programming in school	16	31
Start	2020	Arguments for learning to program	214	252
Start	2020	Arguments for not learning to program	5	5
Start	2020	Critical towards programming in school	14	25
Start	2020	Positive towards programming in school	100	133
End	2020	Critical towards programming in school	6	7
End	2020	Positive towards programming in school	64	92

12.4.1 Teachers Attitudes and Arguments Towards Programming for All

Arguments. We used this theme to identify teachers' perceptions of why everyone should learn to program. We were looking to see to what extent teachers feel that these arguments align with their expectations and practices. We would also identify which reasons they categorize as most important and whether they possibly have other ideas that they believe are important for everyone to learn or not learn to program. A brief explanation of the codes with the most frequency is presented in Table 12.2.

The Attitude of Teachers. We used the theme' attitude of teachers to classify their views on programming for all. The category consists of two codes: 'Positive towards programming in school' and 'Critical towards programming in school'. For 2020 data collected at the end of the course, the code 'Positive towards using coding interdisciplinary' has also been used. A summary of themes and the number of participants is found in Table 12.1.

Positive Towards Programming in School. One hundred sixteen teachers have positive attitudes towards programming. Table 12.3 shows an overview of the codes with the most frequency.

Critical Towards Programming in School. Twenty-two is crucial or questionable against programming for all. These concerns are related to lack of time, colleges that are skeptical of introducing programming, lack of motivation, not seeing the benefits, etc. As part of other subjects such as mathematics, programming will cause less time to learn mathematics. One teacher write: 'Mathematics has been given responsibility for the teaching of programming in schools, and several critical voices have been afraid that the programming part may contribute to less time for the subject of mathematics'.

12.4.2 Arguments for Not Learning to Program

Nine teachers argue that programming is not relevant in some subject areas like health care, construction and electrician. Here are some examples: 'If we think of a student in secondary school, for example, in construction, I see little opportunity to apply programming in that area'. Another teacher writes: 'But personally, it is a bit difficult to see the usefulness of programming when working as an electrician. In my time out at work, I have never been involved in programming, nor have I heard anyone in my company doing this. Feel it is much more relevant for pupils in automation and computer electronics'. Another argument that speaks against programming for all is the high costs of teachers' education and device acquisition. A teacher writes:

Table 12.2 Teachers' arguments towards programming for all

Code	Description	Teacher quotes
A tool for better understanding the subjects	Sixty-eight teachers seem to be concerned with using programming as a tool in the subjects they teach. In this way, they may increase learning outcomes in subjects where programming is applied. Our data show that teachers are most interested in being able to use programming in other subjects	'In physics, we work a lot with formulas and calculations. Here the pupils can program solution formulas and let the computer carry out the calculations in different tasks. Through this, the pupils will learn the formulas better, and they will also have to reflect on what conditions the formulas apply to'
Abstract thinking capabilities	Forty-three argue that learning to think abstractly is essential for developing problem-solving skills. Pupils become more adept at thinking about a solution to a problem without trying it out	'In secondary school age, many pupils can think abstractly; it is an excellent time to work with algorithmic thinking to achieve algorithmic competence in the long run'
Recruiting	Thirty-six teachers argue that programming is the most crucial job skill in the future. Today's and the future's society requires workers who can program. Introducing programming in primary school will increase interest and ensure that more people choose fields of education in programming	'I teach in a lower secondary school. Many of our pupils have begun to think about their future and what they want to be. Many of our pupils have already become interested in IT and programming and want to learn more about this. We have offered programming as an elective for three years, and 15–20 pupils out of 120 have chosen this every year. This shows that there is an interest in learning this'
Interesting, fun and engaging	Twenty-eight teachers believe that programming can be fun and can create enthusiasm in subjects like robotics, control of electrical systems, etc.	'Programming can "trigger"/create enthusiasm for subjects that have not previously been so accessible in schools, such as robotics, control of electrical systems, Lego systems, and development of innovative solutions/apps and not least, it can create curiosity and creativity'

Table 12.3 Attitudes of teachers: positive towards programming

Code	Description	Teacher quotes
Joy and engagement	Programming can evoke joy, commitment, curiosity and be fun	'What gives increased interest and joy? Programming can increase pleasure. Succeeding at some basics, like coding a program that calculates interest, that you can be allowed to spend on trials. I think it creates joy, learning, and curiosity'
Future opportunities	Computing education, including programming, is essential to all, and everyone should get equal opportunity to succeed in the future job market	'It is essential to teach computer knowledge and technology to everyone to give everyone an equal opportunity to succeed in the job market. Minorities and other vulnerable groups will not be able to acquire this knowledge independently. Therefore, it must enter the school—both in terms of job opportunities and understanding society and one's own life'
Time-consuming process	Learning programming might be a time-consuming process. It may, therefore, be advantageous to start this process early. This subject area is large and demanding, which requires learning over many years	"That I make sure that pupils at least get a taste of what they are surrounded by continuously, I consider very natural and important, regardless of whether you end up in this industry or not. It is natural to learn languages such as Norwegian, English, German, Spanish, etc. For many, learning different programming languages will be just as essential"
Contribution to better understanding of other subjects	Many teachers believe that programming help pupils understand other topics better	'Programming may be a useful tool for solving various problems and challenges that we face and better understand other subjects such as STEM-related subjects'

'Arguments that can be used against learning all programming are high costs associated with teachers' hardware and education. Pupils who do not like problem-solving can have an additional discouraged school day'.

Some of the teacher responses seem to contain misconceptions, e.g. the implication that programming is counter to students' wanting to 'work with people and relationships' and give 'cheaper' to learn to program because one senior teacher is cheaper than multiple junior tutors. Student misconceptions and judgments of

Table 12.4 Reference to arguments for learning to program form the textbook

Code	# references
Broadening participation	239
Computational literacy	299
Computational thinking	194
Jobs	320
Learn about world	290
Productivity	139

programming and firmly related subjects have been investigated for quite a while [25–27].

12.4.3 Arguments for Learning to Program from the Literature

In this study, we are interested in what arguments in-service teachers believe are most important to them. Views referenced by teachers without further reflection are categorized under the theme 'Arguments from the literature' and omitted in this study. The number of teachers that used these arguments is 190, with 1706 codes identified which are overlooked and not further analysed. Table 12.4 shows the code frequency.

12.5 Discussions

Our study results indicate that in-service teachers are positive towards programming for all, even though this is challenging and difficult to achieve [3, 4]. Teachers in this context are most concerned with using programming as a tool to increase the pupil's understanding of multidisciplinary subjects. Many in-service teachers are short on time, and therefore the application of programming in their discipline may be a motivating factor. In this way, they may increase learning outcomes in subjects where programming is applied [13, 14]. Abstract thinking capabilities is another area that concerns highly in-service teachers in this study. Computational thinking can be considered a problem-solving toolset that applies to computing principles such as abstraction [7, 28] and has been offered as a cross-disciplinary set of mental skills [29]. Many in-service teachers agree that this is an essential component of developing problem-solving skills in interdisciplinary subjects and promoted higher-order thinking [13, 14]. Many in-service teachers argue that programming is one of the most crucial job skills in the future, and at the same time, a time-consuming process; therefore, pupils must start learning to program early. Several teachers express that programming may even out differences in the population. Everyone will have the same opportunity, regardless of social background and gender. In the literature, we find that computing education, including programming, is essential to all, and everyone should get an equal opportunity to succeed in the future job market [14]. Technology discoveries will create millions of new jobs in programming and related areas. It will be challenging to find talents with the required education and skills for these occupations [30]. Recruitment seems to occupy in-service teachers significantly. However, they find some of these arguments in the textbook [14], which may have influenced their perceptions after becoming more proven on this idea. Teachers mention few other views, such as 'programming may bring innovation' and 'develop pupil's creativity skills'. There are, however, relatively few teachers who are engrossed in these. In-service teachers also express scepticism about introducing programming as part of other subjects (such as mathematics), which may go beyond their allocated time.

The results of this study indicate that teachers believe programming can be engaging and fun [2]. Although the vast majority in this study have positive attitudes towards programming, we can also see negative attitudes. These concerns are mainly related to lack of time. Humble and Mozelius [31] also report this issue in their article, where they discuss obstacles and opportunities in integrating programming in the K-12 setting. Having enough time to teach students how to program and using it as a tool for developing other skills is highly important [32]. Introducing programming as part of other subjects might complicate teachers' situation and negatively affect the learning outcomes. The schools receive some government financial support for sending in-service teachers to these courses. Few reports that their motivation for attending the program has been to achieve credits (ECTS) and not being motivated to learn to program and believe that everyone does not need to know how to program. We find some of the same arguments among other critics[1] of 'programming for all'[2] (future jobs, computational thinking, etc.) [33]. Our results show 25 references to being critical towards programming at the beginning of the course (see Table 12.1), while we found only seven references for the same theme at the end of the course in 2020. There is no link between these results; those seven references may be from the same teachers or different ones. This decline in the number of in-service teachers being critical to 'programming for all' may indicate a change in their attitudes during the course. However, we have not focused on studying whether teachers in this course change attitudes along the way.

12.6 Conclusion and Future Work

The preliminary results of a study on how in-service teachers conceive and interpret programming for all and why programming should (or should not) be included in

[1] https://techcrunch.com/2016/05/10/please-dont-learn-to-code/.

[2] https://www.wise-qatar.org/coding-cognitive-abilities-michael-trucano/.

school curricula were summarized in this paper. The paper investigates the value of computational thinking, coding and programming in school and in-service teachers' reactions to the prospect of programming becoming a full autonomous subject matter. Most of the in-service teachers in this study believe that programming can be fun and engaging. They come up with many arguments and reasoning for why they should include programming in school. Considering that this type of training requires a substantial investment, both at the individual and the school level, we claim it is essential to investigate teacher's attitudes towards programming for all.

This study suggests that in-service teachers may be concerned with using programming as a tool in the subjects they teach. In this way, they may increase learning outcomes in subjects where programming is applied. Simultaneously, they express scepticism about introducing programming as part of other subject areas, which may go beyond the allocated time available. Although the vast majority in this study have positive attitudes towards programming, we can also see negative attitudes. These concerns are mainly related to a lack of time and disagree that everyone must learn to program. We are fully aware that in-service teachers' attitudes and arguments for why everyone should learn programming may have been affected by the literature study (textbook) in the course. Another limitation of this study is that most participants are already aware of the benefits and importance of learning programming (compared to the ones less convinced of its value and hence would not be in this course). Participants differ in terms of the subjects they teach, school type and motivational factors. Therefore, the study covers different perspectives. The main contribution of this paper is that our data indicates a clear positive evolution of the in-service teacher's perception towards the dissemination of coding and programming in school curricula.

Although the results are promising, they are mainly from one course on programming for teachers and their experiences, and it might not be easy to generalize the results. Researchers need to investigate further how programming is taught to this group at other institutions compared to this approach. Studies across countries would be relevant to understanding how the educational system and other programming courses impact teachers' attitudes. Many of the teachers participating in this course had not yet taught programming in their classes. So, they were not speaking from that experience. A follow-up study might be relevant after they have included programming in their courses for a couple of years.

References

1. Kalelioğlu, F.: A new way of teaching programming skills to K-12 students: Code. org. Comput. Hum. Behav. **52**, 200–210 (2015)
2. Cooke, L., et al.: Can everyone code?: preparing teachers to teach computer languages as a literacy. In: Participatory Literacy Practices for P-12 Classrooms in the Digital Age, pp. 163–183. IGI Global (2020)
3. Cheah, C.S.: Factors contributing to the difficulties in teaching and learning of computer programming: a literature review. Contemp. Educ. Technol. **12**, ep272 (2020)

4. Jiau, H.C., Chen, J.C., Ssu, K.-F.: Enhancing self-motivation in learning programming using game-based simulation and metrics. IEEE Trans. Educ. **52**, 555–562 (2009)
5. Winslow, L.E.: Programming pedagogy—a psychological overview. ACM SIGCSE Bull. **28**, 17–22 (1996)
6. Makhmudov, K., Shorakhmetov, S., Murodkosimov, A.: Computer literacy is a tool to the system of innovative cluster of pedagogical education. Eur. J. Res. Reflect. Educ. Sci. **8** (2020)
7. Angeli, C., Giannakos, M.: Computational thinking education: issues and challenges (2020)
8. Wing, J.M.: Computational Thinking (2008)
9. Blackwell, A.F.: What is programming? In: PPIG, p. 20 (2002)
10. Popyack, J.L., Herrmann, N.: Why everyone should know how to program a computer. In: IFIP World Conference on Computers in Education, pp. 603–612 (1995)
11. Hutchison, A., Nadolny, L., Estapa, A.: Using coding apps to support literacy instruction and develop coding literacy. Read. Teach. **69**(5), 493–503 (2016)
12. Martin, C.: Coding as literacy. In: The international encyclopedia of media literacy, pp. 1–8 (2019)
13. Wang, X.-M., et al.: Enhancing students' computer programming performances, critical thinking awareness and attitudes towards programming: an online peer-assessment attempt. J. Educ. Technol. Soc. **20**, 58–68 (2017)
14. Guzdial, M.: Learner-centered design of computing education: research on computing for everyone, p. 147
15. Cortada, J.W.: What is computing? In: Living with Computers, pp. 1–4. Springer (2020)
16. Falloon, G.: An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. J. Comput. Assist. Learn. **32**, pp. 576–593 (2016)
17. Adu-Manu, K., Arthur, J., Amoako, P.: Causes of failure of students in computer programming courses: the teacher learner perspective. Int. J. Comput. Appl. **77**, 27–32 (2013)
18. Leifheit, L., et al.: Development of a questionnaire on self-concept, motivational beliefs, and attitude towards programming. In: Proceedings of the 14th Workshop in Primary and Secondary Computing Education, pp. 1–9 (2019)
19. Malmi, L., et al.: 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 (2020)
20. Bourdieu, P.: The forms of capital (1986). In: Cultural Theory: An Anthology, vol. 1, pp. 81–93 (2011)
21. Oliver, J.S., Simpson, R.D.: Influences of attitude toward science, achievement motivation, and science self concept on achievement in science: a longitudinal study. Sci. Educ. **72**, 143–155 (1988)
22. Horzum, M.B., Cakir, O.: The validity and reliability study of the Turkish version of the online technologies self-efficacy scale. Educ. Sci. Theory Pract. **9**, 1343–1356 (2009)
23. Schunk, D.H.: Learning Theories: An Educational Perspective, 6th edn. (2012)
24. Thorsnes, J., Rouhani, M., Divitini, M.: In-Service Teacher Training and Self-efficacy
25. Mayer, R.E.: The psychology of how novices learn computer programming. ACM Comput. Surv. (CSUR) **13**, 121–141 (1981)
26. Bayman, P., Mayer, R.E.: A diagnosis of beginning programmers' misconceptions of BASIC programming statements. Commun. ACM **26**, 677–679 (1983)
27. Clancy, M.: Misconceptions and attitudes that interfere with learning to program. Comput. Sci. Educ. Res. 85–100 (2004)
28. Selby, C.C.: Relationships: computational thinking, pedagogy of programming, and Bloom's taxonomy. In: Proceedings of the Workshop in Primary and Secondary Computing Education, pp. 80–87 (2015)
29. Yadav, A., et al.: Computational thinking in teacher education. In: Emerging Research, Practice, and Policy on Computational Thinking, pp. 205–220. Springer (2017)
30. Gordon, E.E.: Future Jobs: Solving the Employment and Skills Crisis (2013)
31. Humble, N., Mozelius, P.: Teacher perception of obstacles and opportunities in the integration of programming in K-12 settings. In: Proceedings of EDULEARN (2019)

32. Jawawi, D.N.A., et al.: Introducing computer programming to secondary school students using mobile robots. In: 2015 10th Asian Control Conference (ASCC), pp. 1–6 (2015)
33. Tellidis, E.G.: Should we teach our students computer science? In: Education and Technology: Manitoba Makers and Coders, p. 147 (2017)

Paper VI

M. Rouhani, M. Divitini and A. Olsø, 'Project-based learning and training of in-service teachers in programming: Projects as a bridge between training and practice,' in *2021 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2021, pp. 262–271

This paper is not included due to IEEE copyright restrictions
available at
https://doi.org/10.1109/EDUCON46332.2021.9453934

Paper VII

M. Rouhani, M. Lillebo, V. Farshchian and M. Divitini, ‘Professional development for in-service teachers of programming: Evaluation of a university-level program,’ in *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, Springer, 2021, pp. 123–134

Professional Development for In-Service Teachers of Programming: Evaluation of a University-Level Program

Majid Rouhani( ), Miriam Lillebo, Veronica Farshchian, and Monica Divitini

Department of Computer Science, NTNU, Trondheim, Norway
{majid.rouhani,monica.divitini}@ntnu.no

Abstract. Professional Development (PD) organizations provide training programs for computer science teachers through teacher PD. Programming as part of a teacher's PD has grown in importance in K-12 education. As a result, an increasing number of teachers, often without sufficient training, are teaching programming in their schools. It's challenging to learn to program, and it's even more challenging to teach it. Teachers must be able to program and teach programming, which is usually done in the context of multiple disciplines. In this study, we take a closer look at a teacher's perspective on teachers' PD in programming. We focus on the PD program offered by our university, composed of two courses over one academic year, and define the following research questions: Which strengths and weaknesses of the PD program provided are suited for in-service teachers when learning to program? Learning programming takes time. How could this long learning process be supported? This is an interview study of sixteen in-service teachers who joined the program. The main findings seem to be that teachers are comfortable with how the program is set up. They prefer flexible courses so that they can adapt the implementation with their regular teaching. They also emphasize the need for a continuous learning process and a community of practice (CoP) to continue developing. The main contribution of this study is the evaluation of the PD program at the university.

Keywords: In-service teachers · K-12 · Professional development · Programming

1 Introduction

Computer science (CS) in general and programming, in particular, have long been thought to be relevant to information technology (IT) experts. However, in recent years, this viewpoint has shifted. Software and technology are increasingly playing a role in practically every element of society and life at the current rate of digitalisation [1]. The teachers' PD within digital competence will therefore become more critical in the time ahead. Effective PD must prioritise teachers' and students' needs and address various elements at both the individual and situational levels [2]. Match PD to instructors' backgrounds, link it with the course's curriculum and employ effective motivational design

E. Barendsen and C. Chytas (Eds.): ISSEP 2021, LNCS 13057, pp. 123–134, 2021.
https://doi.org/10.1007/978-3-030-90228-5_10

to promote teacher engagement are three recommendations made by Qian, Hambrusch's [3] study for building effective online PD for CS teachers. Learning programming is a difficult task, according to the scientific community [4]. Besides, teaching teachers may be more challenging and complex than other forms of adult education [5]. Therefore, it is essential to have a teacher perspective to map their needs and expectations on PD in programming.

In this study, we take a closer look at a teacher's perspective on their PD in programming and seek the answer to these research questions: (RQ.01) Which strengths and weaknesses of the PD program provided are suited for in-service teachers when learning to program? Learning programming takes time and might be challenging to learn and teach. (RQ.02) How could this learning process be supported? This is an interview study of sixteen in-service teachers who are preparing to teach programming at school. The sample includes teachers who attended a programming course at the university level. We used the thematic analysis method to analyse the qualitative data.

The following section presents the related work and positions the article in the context of the PD of in-service teachers in programming. Section 3 presents the case and research method. In Sect. 4, we present the results and discuss the main findings in Sect. 5. Conclusions and further work is presented in Sect. 6.

2 Related Work

There has been growing interest in incorporating programming into teacher education and PD in recent years [6]. This process, however, has not been easy; various studies have shown difficulties in teaching computer programming, and teachers sometimes do not know where to get proper training or how to use what they have learned in their classrooms [7]. At the same time, we know that learning programming is a time-consuming process, and in-service teachers usually do not have the required time. There are also challenges associated with teaching programming. Furthermore, the tools and technologies used when developing programs change over time and maintaining this knowledge is essential [8]. Teachers must prepare to integrate digital competencies into their teaching as computer science becomes more widely integrated into school curricula in a growing number of countries. This integration is a moving target, with new methods, tools, and applications appearing and disappearing at such a rapid pace that teachers must have the confidence to explore what is brand-new, relevant and plan their educational activities to include digital competencies independently and continuously [9]. PD is widely agreed to be critical for curriculum innovation in computer science teaching. However, in many countries, preparing PD for teachers is problematic due to instructors' lack of technological experience and abilities, pedagogical understanding, and topic knowledge of computer science [10].

Effective PD seems necessary, but what does it mean, and how do teachers view this from their point of view? The American Learning Policy Institute published a report in 2017 that outlines seven characteristics of effective teacher PD that lead to changes in teacher behaviour and improved student learning outcomes: focus on specific content, incorporation of active learning, support for collaboration, use of effective practice models, providing coaching and expert support, offering feedback and facilitating reflection, and is of sustained duration. Effective professional learning incorporates most or all of these elements [11]. Several studies have found that PD is more successful when six features are included, defined as required or sufficient circumstances, e.g. [12, 13]. Despite minor changes in terminology and disagreements at the periphery, the core statements are highly consistent, as demonstrated by a recent meta-synthesis [14]. Sims and Fletcher-Wood [15] reexamines in their research the evidence that supports this consensus, suggesting that the reviews on which it is based have significant methodological weaknesses, such as using inadequate inclusion criteria and relying on an incorrect inference procedure. As a result, the consensus is likely to be incorrect. According to the argument, researchers might make more headway in identifying characteristics of good PD if they looked for alignment between evidence from basic research on human skill acquisition and aspects of rigorously evaluated PD treatments.

According to a survey of research published in the United States between 2010 and 2014, most computer science PD programs were less than a week-long and not stretched out over time, rather than taking the form of a one-time summer workshop [10]. However, there has been discussion in recent years about the need to move away from this type of traditional intensive training proposal to create spaces where teachers play an active role in learning, requiring studies that address the 'teaching experience' in these new training settings [16]. According to numerous research, the most productive context for informal workplace learning is a school culture that supports and values collaborative learning [17]. Cascade training, a "train the trainer" technique in which the first generation of trainers receives training and then delivers the specific content to the next generation of trainers, is a very complimentary paradigm in education. This procedure can be repeated indefinitely [18]. This study's main contribution is to investigate the PD program provided by the university in programming and how they expect the long learning process to be supported from a teacher's perspective.

3 Case Description and Research Design

The program is designed for in-service teachers in grades 8 through 13. (secondary school). With web-based lectures and weekly activities, the program is online, with no physical gatherings. Students enrol in these courses with the approval of their school and agree to provide teachers with free time to complete the course. The completion rate of the program was 87%. A nationwide program funded by the Ministry of Education partially covers the higher expenditures for schools[1].

One hundred ninety-two in-service teachers from different districts of the country participated in this PD program. More than 90% of the teachers in this study work in secondary schools, with most of the teaching in upper secondary schools (In the national educational system, this corresponds to the last three years of the K-12 system). STEM-related subjects are taught by more than 80% of the teachers. Others instruct in various areas, including language, history, music, arts and crafts, and so on. In their schools, 24% of participants have already taught programming. Before beginning the first course, 61% of participants had some programming experience. The participants

[1] https://www.udir.no/kvalitet-og-kompetanse/.

all are from the same country, although they represent various schools and districts. All schools follow the national curriculum. Participants have varying levels of teaching experience, ranging from new teachers to those who have been teaching for many years. The incentive to enrol in these academic courses varies as well. The majority think they are motivated, but others are less convinced that programming should be included in their classes. The curriculum comprises two courses totalling 15 ECTS credits. The first course (7.5 ECTS) covers the fundamentals of programming (variables, operators, if- and loop statements, lists, functions and libraries). The programming language we use is Python. The textbook "Starting out with Python" [19] is used. Student activities in the first course include three mandatory exercises, a reflection note and a mini-project (two weeks duration). Students can also complete specific assignments from the book. In the second course, we focus on how to teach and apply programming to interdisciplinary subjects. Student activities are four mandatory exercises, a feasibility study and a project. We have organised this course around a 6-week project aiming to define and evaluate activities to be used in teacher's classrooms when teaching programming. As an outcome, teachers create a teaching plan in programming for their specific subject that they can use in their practice.

3.1 Research Design

The Overall Method. This study is based on sixteen interviews with in-service teachers in K-12. The study uses semi-structured interviews to explore the research questions by capturing teachers' reflections on their PD program.

Interview Guide. The interview guide was constructed based on the following main elements: Part A (the PD program): (1) The professional development program at the university is divided into two programming subjects. In the first part, the focus is on learning to program, while in the second part, the focus is on teaching programming. What do you think about this form? (2) The second programming course is designed to be flexible, where students can set up their learning trajectories. What is your opinion on learning programming in a subject like this? (3) Do you think it would be necessary to have additional courses which build upon this program? Part B (PD in general): (1) What is the best way for in-service teachers to learn to program? (2) Have you ever been in a situation where you had to learn some programming concepts with your students? How has it affected your role as a teacher? (3) Learning programming takes time. How could this long learning process be supported? (Courses, seminars, CoP, school activities, peer learning, etc.)

Participants. The research team sent an invitation letter to all the participants of the 2020/2021 cohort. Sixteen teachers expressed interest in participating in the study: T1 (F, USS, Mathematics, physics, technology and research theory), T2 (F, USS, Mathematics and science), T3 (M, USS, Mathematics and science), T4 (F, LSS, Mathematics, social studies and Norwegian), T5 (F, USS, Physics), T6 (M, LSS, Arts and Crafts, Programming), T7 (M, USS, Chemistry, mathematics and science), T8 (M, LSS, Programming, mathematics, and science), T9 (M, LSS, Programming, mathematics, and science), T10 (F, USS, Mathematics), T11 (F, LSS, Programming, mathematics, and science), T12 (F, USS, Mathematics, biology, and science), T13 (F, USS, vocational subjects in business support for ICT service, user support for ICT), T14 (F, LSS, Mathematics, science, technology and design), T15 (M, LSS, Programming, mathematics, and science), T16 (M, LSS, Norwegian, English, media and information, and programming). Interviews were conducted via Zoom, using the service offered internally by our university for GDPR-compliance. The interviews were recorded with a Zoom recorder and then transcribed by the interviewer. The relevant national agency approved the research. All the participants have been informed about the study, their rights and have been explicitly given their consent.

Data Preparation and Coding. We recorded interviews with teachers over zoom. Two of the researchers divided the recordings equally and proceeded to transcribe them individually. Some teachers requested approval of transcriptions before using them in this study. The two researchers coded a few transcriptions individually to create some initial codes and categories. The research team agreed on a joint codebook. Since we only use two coders, they used the method "Percent Agreement for Two Raters[2]." Approximately 20% of transcriptions were coded by two researchers independently, and the research team calculated the inter-rater reliability. When we achieved an IRR of about 80%, we encoded the rest of the transcripts.

4 Results

Some of the study's findings are presented in this section. Due to length limits, we can only give a high-level overview of the most frequently used codes. Direct quotes from the reflection notes are translated from Norwegian by the writers.

4.1 Teachers' PD in the Context of the Programming Courses at the University (Part A)

All interviewees have given their opinion on how the program has contributed to their PD in the area of computer science. We ask teachers about their views on how this form of PD fits and support their advancement in the field. In the following sections, we explain the most frequently codes discussed by participants.

Organization/Structure. Although there was both positive and negative feedback on the organisation and implementation of the courses, most are optimistic about the way the program is set up and performed. The positive aspects are that the program focuses on both learning programming and applying it in practice, flexibility concerning the choice of topics in the course and time for completion of the activities, and focus on programming didactics. The negative aspects are about the connection between lessons and exercises; exercises must be more relevant to teachers' disciplinary knowledge; flexibility of the course is challenging for some and struggle to find the right balance of work; and desire for even more lectures/follow-ups. Fourteen teachers liked how the

2 https://www.statisticshowto.com/inter-rater-reliability/.

program was organised or structured. Amongst these teachers, many noted that they appreciated how the program was structured throughout the year, with the fall semester having an introduction to programming and the spring semester focusing on teaching programming in school. E.g., "I think it has been excellent. It was very nice to get that run-up with learning basic programming this fall and then direct it towards teaching this spring. Especially now, considering that it is very concerned about how we should use it further in teaching. So, this spring semester has been beneficial considering that, so I think it's helpful with the division that has been. It has been beneficial to learn the basics before moving on to thinking about using them for your teaching" (T10).

The Flexibility of the Course. The course, which runs in the spring, is flexibly arranged. This means that teachers essentially choose topics that are relevant to their subject areas. Throughout the semester, they work with subject matter and exercises that build upon each other, and teachers can use that to create a teaching plan for use in their teaching later. Fourteen teachers reflect on the flexible nature of the course. Teachers feel that this allows them to balance their coursework with their work and family life and focus on relevant topics to their subjects. E.g., "… And the spring has been quite flexible in a way, which was very good. It was a bit more to learn, but you got to use it. And that was perhaps the most important thing; not just learning the code, but also using it. Yes, I think there is a good distribution like that" (T13). "Do you mean that we can choose a lot ourselves what we want to do in the exercises? Yes, I think it is essential because we are in different subjects and have different grade levels. We have other focus areas, so I think it was very nice to imagine real situations in my classes" (T4). Few teachers also mention the negative impact of flexibility because it may be confusing and overwhelming. Some teachers may have little knowledge or understanding of programming to decide what is helpful for them to learn. An introduction to each part of the course could be beneficial to make awareness when deciding which areas they have to focus on throughout the course.

Teacher's Needs and Interests. Although many (14) express that they have learned enough programming and can apply it in their subject areas, some (7) also point to more miniature courses to learn in-depth topics such as Arduino, micro: bit, raspberry pi etc. Some others suggest dividing the program into smaller parts to create classes with more homogeneous students regarding background knowledge and their teaching level. Others say that more courses are not needed now as they need time to absorb what they have learned so far. In addition, it is said that they have come far enough to build on this knowledge by themselves. Many of the teachers (8) were optimistic because they felt that the course had allowed them to learn relevant things for their classes. E.g., "I think it is essential because we are teaching different subjects, and we have different grade levels. We have other focus areas, so I think it was very nice to have the option of using relevant cases in the class" (T5). When it comes to time usage, teachers say that even though it is challenging to find enough time, more time should be allocated through school and privately, which means that one gets to develop more. Few teachers point to the technical infrastructure, and related challenges that might need a bit more attention than they currently get in the course. E.g., "Yes, we have been lucky, so we have bought everything because the principal is so fond of programming and technology. But we experience technical issues, e.g., updating the firmware etc. It would have been nice if the course focused a little more on technical challenges as well…" (T14). There is a need to support teachers with technical issues, considering that their available infrastructure might vary.

Workload. Many teachers (10) talk about the amount of work in the course and seem to be divided in the middle. Those who think there is too much to do, have not allocated enough time. In addition, it is mentioned that teachers are not aware of how much work it is to learn to program. The others think that the workload is appropriate and emphasize that the flexibility (both in terms of content and times for submitting the exercises) helps to even out the load.

Final Assessment/Collaboration. The examination form in the course is project-based, where several people can collaborate. This seems to many teachers (8) to be positive. E.g., "I really liked the shape of the final assessment, that it is a project. I think it's incredibly significant because there are a lot of people taking exams right now, and I see how stressed they are, and they memorise, memorise and memorise, and go to the exam, then the exam is finished, and they may forget everything shortly after the exam. But here, as I go through the exam, I learn at my leisurely pace, writing and noting every detail, and afterwards, I can use this project as a finished product. And if I want to repeat and come back, then I can come back, print it, and use it right away. I think this is fantastic with this project" (T12). Many teachers felt that collaboration during the course allowed them to use their strengths and support each other as a group. One teacher suggested that an additional activity in the study could be for teachers to share and present the teaching plans that they have created to their peers. Teachers also report on good collaboration between instructors/teaching assistants, and many felt a low threshold for asking questions and got responses quickly.

4.2 Teachers' Perspectives on PD (Part B)

The second part of the interviews was related to teachers' reflections on their future PD in programming after completing the university's programming courses. By completing these courses, they have gained some experience and may have thoughts and opinions about how they envision the subject area's development. We asked teachers what they think is the best way of PD in programming for in-service teachers. Learning programming may be time-consuming, and we asked them how they want this process to be supported. Furthermore, we were interested in knowing their view of the development in the subject area and the students since the learning process can extend over a more extended period.

Teachers' Perspective on How to Learn to Program. All interviewees (16) expressed their opinion on how teachers should learn to program. We have identified 170 code references in transcriptions where teachers discuss different aspects such as the community of practice, collaboration and competency development through participation in courses, seminars, and self-initiatives. Thirteen teachers mention collaboration in the interviews and consider it to be an essential tool in learning programming. They emphasize the importance of collaboration with fellow teachers to share skills and competencies and motivate each other. Thirteen teachers talk about courses or seminars as a meaningful way to learn and teach programming. Many teachers mention also practising and learning by doing as an essential part of the process. Programming is time-consuming, and teachers need to find the required time to do this. One teacher says that programing should be like a 'hobby': "Learning programming should become a hobby. One must prioritise and spend time to learn and understand the concepts well enough" (T9). Nine teachers expressed that a teacher professional environment may be an important arena for learning to program. Educating teachers could happen internally at the school through courses, or that the school would allocate time for teachers to learn together at school.

Time. When we talk about the PD of teachers in programming, we keep coming back to the time issue. It is crucial to allocate enough time to practice and improve programming skills. Fourteen teachers talk about "time" as a constraint when learning to program. E.g., "The fact that schools must give time to learn this, I think, is essential. Because it does not come by itself, and you cannot expect anyone to sit down to do this on Saturday nights because this is too big for it" (T10).

Willingness to Learn More. Sixteen teachers talk about their desire to learn more and cope with the pupil's motivation to learn to program. E.g., "I could well imagine such an additional in-depth course, but the question is what you are allowed to do as an employer. The employer wants to lift now with a minimum of programming skills. So, we get a 15 ECTS credits course now and then we are supposed to know to program" (T1). Some teachers seem to experience that their work colleagues are unwilling or unable to learn to program and report that this applies especially to older teachers who have started planning their retirement rather than learn programming.

Role as a Teacher. Some teachers emphasize the importance of active learning. Pupils need to be guided in the right directions and also learn themselves during this process. One teacher explained that she was anxious about not having as much knowledge about programming as necessary. At the same time, the teacher felt that there might be some advantages because the pupils might become more motivated by knowing that they are more knowledgeable on a subject than the teacher is. E.g., "I like knowing well what I am going to teach. So being a week in front of the students, it's not something I like. But I think it is also a considerable value because if the students discover that they know something that the teachers do not know, they become extra engaged in it and want to show it off and get even more motivated to learn even more. So I feel a little uncomfortable, but I know it may be excellent" (T10). Fourteen teachers talk about their experience when learning to program together with pupils, and those who have not experienced it but reflect on the possibility of it happening in the future. Twelve teachers described that they have been in situations where they have had to learn together with their pupils. E.g., "I think that I will come across that. Well, programming is relatively new for many pupils. Still, eventually, we will work with it throughout primary school. When they get to high school, they will probably have a completely different base of competence than what they have now, so I imagine that it will not be long before learning something with the pupils" (T10). Few teachers mention that they might have unrealistic expectations about what they need to know and put too much pressure on themselves, probably lowering their confidence. E.g., "I think, in the textbook, the author says that when we learn to read and write, none of us expects us to be world-famous as Ibsen or Byron, or who at any time. We learn to read and write because we want to write a letter to Grandma. Why do we look at programming so that if we are to learn to program, then we must be like Steve Jobs? And it helped me lower my shoulders and the way the instructor explains here in this course. Very calm with a good pace, do not expect you to have fantastic prior knowledge, we begin" (T12). Another perspective that teachers (6) have talked about is the willingness of the school administration concerning making necessary time and resources available for teachers. Some schools are extra supportive, while others teachers feel they do not have enough back cover.

5 Discussions

Teachers have shared their experiences and highlighted some essential points from their perspectives on the PD in programming at the university. They seem to be comfortable with how the program is set up, i.e. the first part (7.5 ECTS) focuses on teaching them basic knowledge in programming. In comparison, the second part (7.5 ECTS) focuses on didactics, i.e. how to teach programming. Teachers emphasize the importance of *connecting programming to the relevance of their subject areas* from the outset, even if the focus is on fundamental concepts, such as using relevant examples and assignments from the subject areas in upper secondary schools. This can contribute to increased learning outcomes and more significant engagement.

Another aspect that teachers highlight is *the flexibility* of the program. Since time is a constraint, teachers are concerned with the shortest path to the goal, i.e. learning only what is relevant to their situation in the first place to get started with teaching programming in a short time. This offers a flexible solution where participants can set up their learning paths. Several dimensions can make the program's flexibility valuable: On the personal level, participants vary in terms of their prior knowledge level, attitudes, and motivation; on a pedagogical level, they teach different subjects, school level and focus areas; and in a school context, the support by leadership and technical infrastructure may vary. There is a need to support teachers with *technical issues*, considering that their available infrastructure might vary. This problem could be addressed with (hands-on) tutorials about the more common technologies and troubleshooting sessions.

Teachers also mention the *project-based learning approach and the bridge model* [20] used in this PD program. They develop a teaching plan that they can use directly in their teaching after completing this program. This may increase their self-efficacy in teaching programming. Regarding teachers' PD in programming after completing the university's programming courses, we asked them to reflect on what they think is the best way of PD in programming for in-service teachers. What seems to occupy teachers the most is *establishing a community of practice, collaboration and development through participation in courses, seminars, and self-initiatives*. They emphasize the importance of collaboration with fellow teachers to share skills and competencies and motivate each other. Continuous development through collaboration may be an essential factor that may be facilitated at several levels: between teachers, teachers and pupils, school & teachers internally and externally. Even for teachers who have attended an extensive program, most of them still feel the need to continue their learning process with additional courses, short term workshops, a CoP, collaboration at school etc. It is therefore essential to see any PD as a step in a life-long learning process.

Teachers are concerned with several challenges when learning to program, and the most important one is the *lack of time* that may have several negative aspects. Learning to program can be time-consuming, and in-service teachers need to develop their programming knowledge in parallel with other duties, making it even more challenging. The lack of time can lead to the whole PD becoming complicated and may demotivate the individual. Therefore, teachers must receive the necessary *support from the schools* (both in terms of time and financial support for purchasing equipment, etc.). Good PD is needed to achieve the right level of competence, and that this requires adequate support from the school. Teachers getting the time to attend courses at the university level is beyond what a single school can generally manage. This implies that there is a need for a higher-level commitment. In our case, schools get public support for it. We think it is important to underline that proper qualification of teachers cannot be achieved without a commitment that does not put all the weight on the shoulder and goodwill of individual teachers and schools. Another point is the *expectations of schools* for these types of PD programs. Many schools might think that PD in this area is something to do once due to changes in the national curriculum, rather than as a continuous effort that requires time, dedication, and a clear strategy.

Another practical issue is that PDs that follow the school calendar can create problems for teachers as they already follow their pupils through the same calendar. Thus, the exam may coincide with the exam that the students should have, which complicates the situation for teachers. This reinforces the need for a flexible PD program, but at the same time, we see that more teachers express that there will be challenges in finding out what to do as the scope becomes overwhelming. Instructors must therefore follow them closely. The *role of the teacher* may not be the same anymore. They point to some area that requires changes in the way they think and teach. Teachers might have *unrealistic expectations* about what they need to learn and put too much pressure on themselves, probably lowering their confidence. There might be a need to scaffold teachers' reflection on the competence level that they want/need to achieve. Another point that several teachers emphasize is that active learning should be an essential part of *the teaching process*. Pupils need to get involved, and teachers do not have to worry about not knowing enough about the topic. They will, to a greater extent, learn and develop in the subject together with the students.

6 Conclusion

This paper presented the results from the analysis of sixteen interviews of in-service teachers attending a PD program about learning and teaching programming. Our main objective was to evaluate the PD program offered by the university and asked the research questions: Which strengths and weaknesses of the PD program provided are suited for in-service teachers when learning to program? Learning programming takes time;

how could this long learning process be supported? The main findings seem to be that teachers are comfortable with how the program is set up. They also emphasize the importance of connecting programming to the relevance of their subject areas from the beginning, establishing a community of practice, collaboration and development through participation in courses, seminars, and self-initiatives. They highlight the importance of collaboration with fellow teachers to share skills and competencies and motivate each other. Also, the follow-up initiatives and the need for schools to create supportive environments for cooperative efforts and CoP are essential factors. The role of the teacher may not be the same anymore. They point to some area that requires changes in the way they think and teach. Teachers might have unrealistic expectations about what they need to learn and put too much pressure on themselves, probably lowering their confidence.

The main contribution of this study is the evaluation of the PD program provided by the university in programming. Results may be applicable in a PD program with a similar context but not be easily generalised. All participants are from the same country but teach on different school levels and subjects. The study focuses on teachers who have been given by their school the time to improve their teaching. Further, teachers' perspectives on PD in programming may be seen in the context of the university course already being provided. The findings may be affected by the highly motivated students who participated in this study and were pleased with the course.

As part of future study, it might be interesting to study teachers' perceptions from different institutions in future studies—both teachers who do not participate in any particular PD program and those who do.

Acknowledgments. We thank the teachers who participated in the interviews. The work is partly funded by the Norwegian Directorate for Education and Training (https://www.udir.no) and the Norwegian Center for Excellent IT Education (https://www.ntnu.edu/excited).

References

1. Nouri, J., et al.: Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. Educ. Inq. **11**(1), 1–17 (2020)
2. Galaczi, E., et al.: The Cambridge English Approach to Teacher Professional Development (2017)
3. Qian, Y., et al.: Who needs what: recommendations for designing effective online professional development for computer science teachers. J. Res. Technol. Educ. **50**(2), 164–181 (2018)
4. Ring, B.A., Giordan, J., Ransbottom, J.S.: Problem solving through programming: motivating the non-programmer. J. Comput. Sci. Coll. **23**(3), 61–67 (2008)
5. Mozelius, P., Humble, N.: Lessons learnt from teacher professional development in programming. In: INTED 2020. IATED Academy (2020)
6. Monjelat, N., Lantz-Andersson, A.: Teachers' narrative of learning to program in a professional development effort and the relation to the rhetoric of computational thinking. Educ. Inf. Technol. **25**(3), 2175–2200 (2020). https://doi.org/10.1007/s10639-019-10048-8
7. Hiltunen, T.: Learning and Teaching Programming Skills in Finnish Primary Schools-The Potential of Games. University of Oulu, Oulu (2016)
8. Baxter, D., Simon, B.: Clash of the timelines: lessons learned from the front lines of CS education. In: Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment (2014)

9. Nordén, L.-Å., Mannila, L., Pears, A.: Development of a self-efficacy scale for digital competences in schools. In: 2017 IEEE Frontiers in Education Conference (FIE). IEEE (2017)
10. Menekse, M.: Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014. Comput. Sci. Educ. **25**(4), 325–350 (2015)
11. Darling-Hammond, L., Hyler, M.E., Gardner, M.: Effective teacher professional development. Learning Policy Institute (2017)
12. Cordingley, P., et al.: Developing great teaching: lessons from the international reviews into effective professional development (2015)
13. Desimone, L.M.: Improving impact studies of teachers' professional development: toward better conceptualizations and measures. Educ. Res. **38**(3), 181–199 (2009)
14. Dunst, C.J., Bruder, M.B., Hamby, D.W.: Metasynthesis of in-service professional development research: features associated with positive educator and student outcomes. Educ. Res. Rev. **10**(12), 1731–1744 (2015)
15. Sims, S., Fletcher-Wood, H.: Identifying the characteristics of effective teacher professional development: a critical review. Sch. Eff. Sch. Improv. **32**(1), 47–63 (2021)
16. Reding, T.E., Dorn, B.: Understanding the "teacher experience" in primary and secondary CS professional development. In: Proceedings of the 2017 ACM Conference on International Computing Education Research (2017)
17. Webb, M., et al.: Computer science in K-12 school curricula of the 2lst century: why, what and when? Educ. Inf. Technol. **22**(2), 445–468 (2016). https://doi.org/10.1007/s10639-016-9493-x
18. Karalis, T.: Cascade approach to training: theoretical issues and practical applications in non-formal education. J. Educ. Soc. Policy **3**(2), 104–108 (2016)
19. Gaddis, T., Agarwal, R.: Starting Out with Python. Pearson, London (2015)
20. Rouhani, M., Divitini, M., Olsø, A.: Project-based learning and training of in-service teachers in programming: projects as a bridge between training and practice. In: 2021 IEEE Global Engineering Education Conference (EDUCON). IEEE (2021)

Paper VIII

M. Rouhani, M. Lillebo, V. Farshchian and M. Divitini, ‘Learning to program: An in-service teachers’ perspective,’ in *2022 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2022, pp. 123–132

Paper IX

M. Rouhani, M. Divitini and A. M. Hashemi, 'Computer science in schools: A literature mapping of professional development for in-service teachers,' in *2022 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, 2022, pp. 164–173

Appendix A

Interview Guides

A.1 Study 2

The interview guide was designed based on three main elements:

- Attitudes towards programming in school
- Self-efficacy in teaching programming
- Self-efficacy in programming

For elements 2) and 3), questions were inspired by the Teachers' Sense of Efficacy Scale (TSES), which is a widely used instrument to measure teachers' self-efficacy. The TSES focuses on efficacy for instructional strategies, student engagement, and classroom management. For element 2) on self-efficacy in teaching programming, questions from the TSES on efficacy for instructional strategies were adapted to the context of teaching programming. For element 3) on self-efficacy in programming skills, original questions were created to investigate how teachers feel about their own programming abilities.

The interviews were conducted using a semi-structured format based on this interview guide, allowing teachers to elaborate on their experiences. The theoretical framework draws upon Bandura's concept of self-efficacy and how formal training can impact teachers' beliefs in their capabilities to teach programming. The complete interview guide for study 2 can be found in Thorsnes [120]'s master thesis.

A.2 Study 7

The interview guide was designed based on the following main elements:

- Part A (the PD program): It included questions about the teachers' views on the structure of the PD program at the university, which was divided into two programming subjects - one focusing on learning to program and the other focusing on teaching programming. It also asked about their views on the flexibility of the second programming course.
- Part B (PD in general): It included questions about the teachers' perspectives on the best way for in-service teachers to learn programming in general, their

experiences of learning programming together with students, and how the long learning process of programming could be supported.

The theoretical background for the interview design draws on literature around effective teacher professional development (PD). Specifically, it is informed by research that outlines characteristics of effective PD such as being content-focused, incorporating active learning, supporting collaboration, providing coaching/feedback, and being of sustained duration. The interviews aimed to understand the teachers' perspectives on these aspects as they relate to their PD program in programming. It evaluated the program based on the teachers' views of its strengths and weaknesses in supporting their learning of programming.

The semi-structured interviews allowed an in-depth exploration of the research questions. The original interview guide in Norwegian for Study 7 is included in the following pages. The English version is available on request.

A.2.1 Interview guide P7

***Fargekoder**: IT6203, IT6204*

Vil du delta i forskningsprosjektet

"Lære å programmere: et lærerperspektiv"?

Dette er et spørsmål til deg om å delta i et forskningsprosjekt hvor formålet er å *øke kvaliteten på programmeringsfagene slik at lærere uavhengig av hvilke trinn- eller fag de underviser på norske skoler skal kunne få mest mulig læringsutbytte*. I dette skrivet gir vi deg informasjon om målene for prosjektet og hva deltakelse vil innebære for deg.

Formål
Dette er et forskningsprosjekt med formål om å øke kvaliteten på undervisning i grunnleggende programmering for lærere. Deltakerne i disse programmeringskursene er lærere fra norske skoler (i hovedsak ungdoms- og videregående skoler) som underviser i forskjellige fag og som har forskjellige utgangspunkter når det gjelder programmeringskunnskaper. Vi ønsker å finne ut hvilke utfordringer som er meste gjeldende fra et lærerperspektiv. Spørsmål vi stiller i denne sammenheng er: hvordan oppleves vanskelighetsgraden på kursene? vil et kurs med fleksibelt innhold være passende? Hvilke utfordringer møter lærere når de skal lære programmering? Hvilke utfordringer får lærere når de skal undervise i programmering?

Hvem er ansvarlig for forskningsprosjektet?
Norges teknisk-naturvitenskapelige universitet/Institutt for datateknologi og informatikk er ansvarlig for prosjektet.

Prosjektet vil være i samarbeid med SFU Excited - NTNU.

Hvorfor får du spørsmål om å delta?
Alle som deltar i kursene IT6203 og IT6204 blir bedt om å delta i undersøkelsen. Deltakerne i disse kursene kan sitte med verdifull informasjon som kan brukes til å tilpasse kursene og tilby innhold og form som øker kvaliteten og læringsutbytte.

Deltakerne vil ikke kunne gjenkjennes i publikasjoner som følge av denne forskningen.

Hva innebærer det for deg å delta?
Hvis du velger å delta i prosjektet, innebærer det at du:

- *Blir bedt om å delta i et intervju. Intervjuet kan foregå over Zoom eller personlig oppmøtet dersom det er mulig. Intervjuet vil ta ca. 30-40 minutter. Det vil bli tatt lydopptak som vil bli transkribert og anonymisert. Du vil bli spurt om hvordan du opplever undervisningen, læringsutbytter, utfordringer med å lære programmering selv eller undervise programmering til dine elever*
- *I tillegg samtykker at dine refleksjonsnotater blir anonymiserte og brukt i forskning. Refleksjonsnotater kan være som del av innleveringene eller avslutningsprosjektet.*

Det er frivillig å delta
Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket

tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger
Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

- Opptak vil bli gjort på NTNUs opptaksutstyr, og håndteres av faglærer Majid Rouhani, som også vil være prosjektleder. Etter at intervjuet er blitt transkribert og anonymisert vil opptak slettes fra NTNUs lydopptaker. Prosjektgruppen vil også ha tilgang til anonymisert data fra intervju. Data vil lagres anonymt ved bruk av koder i stedet for kontaktopplysninger. Data blir slettet innen sluttdato på prosjektet (31.12.2023).
- Vi bruker nettskjema.no for spørreskjemaene. Anonymiserte data lagres i prosjekt perioden som er frem til 31.12.2023.
- Alle anonymiserte refleksjonsnotater fra innleveringer lagres på NTNUs systemer og kun tilgjengelig for prosjektgruppen. Disse slettes innen prosjektets sluttdato (31.12.2023)

  All data anonymiseres og lagres på NTNUs interne systemer og tilgang gis kun til prosjektgruppen. Prosjektgruppen vil være faglærer og forskningsassistenter som blir ansatt. I tillegg vil prof. Monica Divitini som er ansatt på Institutt for databehandling og informatikk være prosjektdeltaker.

Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?
Opplysningene anonymiseres når prosjektet avsluttes/oppgaven er godkjent, noe som etter planen er *31.12.2023. Ved prosjektslutt personidentifiserbare opplysninger fjernes, lyd eller bildeopptak slettes*.

Dine rettigheter
Så lenge du kan identifiseres i datamaterialet, har du rett til:
- innsyn i hvilke personopplysninger som er registrert om deg, og å få utlevert en kopi av opplysningene,
- å få rettet personopplysninger om deg,
- å få slettet personopplysninger om deg, og
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

Hva gir oss rett til å behandle personopplysninger om deg?
Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra *Norges teknisk-naturvitenskapelige universitet/Institutt for datateknologi og informatikk* har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

Hvor kan jeg finne ut mer?
Hvis du har spørsmål til studien, eller ønsker å benytte deg av dine rettigheter, ta kontakt med:
- *Norges teknisk-naturvitenskapelige universitet/Institutt for datateknologi og informatikk* ved *Majid Rouhani, epost: majid.rouhani@ntnu.no.*
- Vårt personvernombud: *Thomas Helgesen, epost: thomas.helgesen@ntnu.no*

Hvis du har spørsmål knyttet til NSD sin vurdering av prosjektet, kan du ta kontakt med:
- NSD – Norsk senter for forskningsdata AS på epost (personverntjenester@nsd.no) eller på telefon: 55 58 21 17.

Med vennlig hilsen

Majid Rouhani
(Forsker/veileder)

Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet «*Et opplæringsrammeverk for lærere: Utfordringer og muligheter for læring og undervisning av programmering*», og har fått anledning til å stille spørsmål. Jeg samtykker til:

- ☐ å delta i *spørreskjema*
- ☐ *å delta i analyse av refleksjonsnotater*
- ☐ å delta i intervju

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet

----
(Signert av prosjektdeltaker, dato)

Intervjuinnhold

Bakgrunn

Helt innledningsvis ønsker jeg å vite litt om deg og din bakgrunn i programmering og som lærer.

1. Hvilke fag underviser du i?
    a. Hvilke av disse fagene tror du du kommer til å undervise programmering i?
2. Hvilket trinn underviser du på?
3. Hvilke erfaringer har du med programmering før deltagelse på introduksjonskurset?
    a. Har du undervist i programmering før? Hvis ja:
        i. Hvilket fag?
        ii. Hvordan gjennomførte du økten og hvordan følte du at det gikk?
    b. Har du tatt noen kurs tidligere som inneholdt noen form for programmering?
    c. Har du sett på kompetansepakken til UDIR «programmering og algoritmisk tenkning?»
    d. Er det noen andre erfaringer du ønsker å dele?

What is the most difficult programming concept to learn for you?

- Previous results/literature says…

How do you think the way you have learned programming influence these difficulties?

Programmering

Utfordringer knyttet din læring

1) Hva er det du opplever som utfordrende å lære (inkludert språkkonstruksjoner, programmeringskonsepter, problemer knyttet til prosessen), og hvorfor?
2) Hvor utfordrende tror du det er å lære et annet programmeringsspråk etter at du har lært Python, og hvorfor?

Utfordringer knyttet dine eleveres læring

1) Hvilke deler av programmering tror du er mest utfordrende for elevene dine å lære?

Undervisning i programmering

1) Hvilke utfordringer mener du vil kunne oppstå ved å:

i. lære elevene dine spesifikke programmeringskonsepter

ii. lære elevene dine spesifikk språkkonstruksjon

iii. motivere elevene dine til å lære

2) Andre refleksjoner rundt utfordringer med læring og undervisning i programmering?

****

Utsagn	Helt Uenig	Uenig	Hverken eller	Enig	Helt Enig
Programmering har ikke vært et enkelt emne for meg å lære	☐	☐	☐	☐	☐
Programmering krever riktig forståelse av abstrakte begreper (f.eks. funksjoner/metoder, osv.)	☐	☐	☐	☐	☐
Jeg har læringsproblemer på grunn av fagets natur.	☐	☐	☐	☐	☐
Jeg tror jeg vil ha utfordringer med å utforme leksjoner slik at de vil være gunstige for elevene mine.	☐	☐	☐	☐	☐
Jeg tror det kommer til å bli spennende å undervise i programmeringskurs.	☐	☐	☐	☐	☐
Jeg kunne lite eller ingen programmering FØR kursstart	☐	☐	☐	☐	☐
Jeg føler at jeg har lært nok programmering nå til å kunne undervise mine elever.	☐	☐	☐	☐	☐

	Vanskeli g	Litt vanskeli g	Hverken enkelt\ vanskeli g	Ganske enkelt	Enkelt
Å finne feil i programmer er	☐	☐	☐	☐	☐

Å dele en oppgave i del-problemer (program-funksjoner) er	☐	☐	☐	☐	☐
Å bruke biblioteker i programmer er	☐	☐	☐	☐	☐
Å forstå hva variabel-initialisering er er	☐	☐	☐	☐	☐
Å forstå hva løkker er er	☐	☐	☐	☐	☐
Å forstå hvordan løkker skal anvendes er	☐	☐	☐	☐	☐
Å forstå hva en valg-setning er er	☐	☐	☐	☐	☐
Å forstå hvordan en valg-setning skal anvendes er	☐	☐	☐	☐	☐
Å lage et program for å løse en bestemt oppgave er	☐	☐	☐	☐	☐

Appendix B

Questionnaires

B.1 Study 1

Upon completion of the course, we distributed a questionnaire to the students, informing them that the anonymized survey data could be utilized for subject-related research. Out of the total 22 students, approximately the same number opted not to complete the questionnaire (stopping at the first question), while the remainder did not initiate the questionnaire at all.

Additionally, in conjunction with the aforementioned questionnaire results, we received a reflective note from each student who successfully passed the course as part of their project submission. These reflections encompass two main topics: firstly, how the course has enhanced their understanding of implementing programming in their respective subject areas and their deeper comprehension of programming concepts. Secondly, students have developed their own "teaching activity" as part of the project submission. Some have successfully implemented this program in their classes for their students and have reflected on the outcomes.

The questionnaire was a practical evaluation survey rather than a theoretically designed research instrument. The purpose was to collect student feedback rather than test hypotheses or theories about flexible learning trajectories.

B.2 Study 3

The questionnaire used in the study is designed as a Likert scale survey. A Likert scale asks respondents to indicate their level of agreement or disagreement on a symmetric agree-disagree scale for a series of statements. In this study, the Likert scale ranges from 1 to 5, where 1 means completely disagree and 5 means agree entirely. Students were asked to rate their level of agreement with various statements about the course on this 5-point scale. The statements/questions in the questionnaire covered the following topics:

1. Communication with teachers/assistants
2. Usefulness of webinars

3. Flexibility of course organization
4. Assessment tasks
5. Feasibility study
6. Project implementation
7. Workload
8. Confidence in teaching programming after the course
9. There was also an open-ended question for additional comments.

This study concentrates on questions 1 and 6-9 from the survey. Alongside the survey, we monitored activities in Slack and facilitated student registration in Slack by creating a link on the Blackboard learning platform. The theoretical background for using a Likert scale survey in this study is that Likert scales are commonly used in educational research to measure attitudes, opinions or perceptions. They allow the collection of subjective data in a standardized way that can then be statistically analyzed. By including questions related to different aspects of the course and learning experience, the survey aims to gather both quantitative and qualitative feedback from students to evaluate the effectiveness of using Slack as a communication platform and how it supported the learning process.

B.3 Study 6

In 2020, we put the updated model into practice and conducted new assessments to gauge progress and determine if the changes resulted in improved goal attainment. We collected data through two avenues: (1) post-course questionnaires and (2) reflective notes submitted as part of the project. Upon completion of the course, participants provided the following feedback to the instructors: Q1: The problem identification phase of the project helped me define the issue and was beneficial before commencing the project. Q2: I anticipate that the project will yield a teaching plan that I can use in my classroom. Q3: I have acquired sufficient programming knowledge in this course for teaching programming in my class(es) and feel confident in my abilities. Q4: How did you find the workload in the course?

Questions Q1 and Q2 pertain to the realization phase of the model, aiming to gauge participants' experiences during the problem identification, design, and development processes. Question Q3 addresses overall perceptions of the course and participants' confidence in teaching programming. Question Q4 solicits feedback on the overall workload, with a "neutral" response being preferable, indicating a satisfactory workload level. In addition to these structured questions, there is an open question asking for general comments, and reflection notes from participants were also analyzed qualitatively.

The theoretical framework for the questionnaire is centered around evaluating the "bridge model of programming for in-service teachers" as outlined in the paper, which is rooted in the principles of project-based learning (PjBL).

B.4 Study 8

We utilized a questionnaire to gather both quantitative and qualitative data as part of a mandatory assignment with a pass/fail grading system. Participants had the choice to opt into the research, and data collection was authorized by NSD, the Norwegian agency responsible for overseeing research data collection and management. The quantitative portion of the data comprises a series of statements related to learning and teaching programming. For statements 1-7, responses were rated on a scale from 1=Completely Disagree to 5=Completely Agree, while for statements 8-16, responses ranged from 1=Very Difficult to 5=Very Easy. Out of 178 reflection notes received, 17 were deemed invalid and excluded, resulting in a dataset of 161 reflection notes.

In the quantitative section, we referred to the curriculum of programming courses in this context and reviewed literature to identify common challenges faced by novice programmers. We formulated statements 1-16 to capture teachers' perspectives on these challenges and to address our main research question. Statements 1-3 focused on the challenges of learning to program, while statements 4, 5, and 7 pertained to teachers' self-efficacy in teaching programming. Statements 8-15 addressed specific challenges teachers encounter when learning programming concepts such as variables, libraries, loops, and select-statements, while statement 16 explored the process of creating programs to solve specific problems.

The theoretical background for designing the quantitative statements came from existing literature on common difficulties faced by novice programmers. The qualitative questions were open-ended to get an in-depth understanding of teachers' perspectives, beyond just rating pre-defined statements.

Original questionnaire used in study 1, 3, and 6. English version available on request.

B.5 Questionnaire P1

Sluttevaluering av fag IT6204 Anvendt programmering for lærere

User Information

Name:	Anonymous	**Email:**	Anonymous
Location:	Anonymous	**Company:**	Anonymous
Position:	Anonymous	**IP Address:**	Anonymous
Started:	01.05.2019 14.28.21	**Completed:**	01.05.2019 14.48.14
Time Spent:	0 days, 0 hours, 20 minutes, 1193 seconds, 1193000 milliseconds	**Custom 1:**	Anonymous
Custom 2:	Anonymous	**Custom 3:**	Anonymous

1. Anonymiserte data fra spørreundersøkelsen kan bli brukt i forskning tilknyttet faget. Velg "Yes" og klikk på "Next" for å være med på undersøkelsen.

Yes

2. Organisering av faget (Modul 0)

	Helt uenig	Uenig	Verken/eller	Enig	Helt enig
Jeg har fått nødvendig informasjon om faget før oppstart					
Modulen gir meg en god orversikt over faginnhold					

3. Diskusjonsforum: Slack

	Helt uenig	Uenig	Verken/eller	Enig	Helt enig
Jeg har brukt Slack i dette faget og synes den var nyttig					
Jeg har brukt e-post til å kommunisere med faglærere/Læringsassistenter					

4. Webinarer (1-5)

	Helt uenig	Uenig	Verken/eller	Enig	Helt enig
Det fungerte fint å koble seg på via Skype					
Kvaliteten på lyd/bildet var god					
Jeg hadde utbytte av Webinar 1					
Jeg hadde nytte av opptaket som ble					

publisert på BB for Webinar 1					
Jeg hadde utbytte av Webinar 2					
Jeg hadde nytte av opptaket som ble publisert på BB for Webinar 2					
Jeg hadde utbytte av Webinar 3					
Jeg hadde nytte av opptaket som ble publisert på BB for Webinar 3					
Jeg hadde utbytte av Webinar 4					
Jeg hadde nytte av opptaket som ble publisert på BB for Webinar 4					
Jeg hadde utbytte av Webinar 5					
Jeg hadde nytte av opptaket som ble publisert på BB for Webinar 5					

5. Kursinnhold (Modul 1-6)

	Helt uenig	Uenig	Verken/eller	Enig	Helt enig
Læringsutbyttene har blitt kommunisert på en klar måte					
I dette emnet er det lett å forstå hvilke arbeidskrav som stilles til meg					
Jeg opplever at læringsaktivitetene (forelesninger/ øvinger/gruppeoppgaver) er relevante for mitt arbeid					
Jeg opplever at jeg hadde et godt grunnlag fra før for å nå læringsutbyttene					
Arbeidsmengden på emnet er for stor					
Emnet forsøker å dekke altfor mange temaer					
Jeg synes vi får nok tid til å forstå det vi skal lære					
Emnet er lagt opp slik at jeg jobber jevnt med lærestoffet gjennom hele semesteret					

Jeg synes det er vanskelig å finne ut hva som forventes av meg på dette emnet					
Underviserne gjør det helt klart fra begynnelsen hva som forventes av meg i dette emnet					
Jeg opplever at fjernundervisningen fungerer godt					
Jeg opplever at emnene i modul 1 er relevante for mitt arbeid					
Jeg opplever at emnene i modul 2 er relevante for mitt arbeid					
Jeg fant minst et emne i modul 3 som var relevante for mitt arbeid					
Jeg fant minst et emne i modul 4 som var relevant for mitt arbeid					
Jeg opplever at emnene i modul 5 er relevante for mitt arbeid					
Jeg opplever at emnene i modul 6 er relevante og øker forståelsen av programvare utviklingsprosessen for meg					

6. Øvinger (1-5), forprosjekt og prosjekt

	Helt uenig	Uenig	Verken/eller	Enig	Helt enig
Øving 1 hadde relevant innhold og passe vannskelighetsgrad					
Øving 2 hadde relevant innhold og passe vannskelighetsgrad					
Øving 3 hadde relevant innhold og passe vannskelighetsgrad					
Øving 4 hadde relevant innhold og passe vannskelighetsgrad					
Øving 5 hadde relevant innhold og passe vannskelighetsgrad					
Forprosjektet hjalp meg med å formulere problemstillingen og var nyttig før oppstart av prosjektet					

Jeg synes prosjektet var relevant og hadde pass vannskelighetsgrad

7. Hvilke digitale kommunikasjons- og samarbeidsverktøy (f. eks. Slack, Itslearning) bruker du på jobben i det daglige, og hva brukes de til?

8. Etter å ha brukt Slack dette semesteret, hva tenker du om å bruke lignende verktøy i skolen, både med kollegaer og/eller elever?

9. Føler du at det er nok tilgjengelige ressurser som forklarer hvordan man kan bruke programmering i klasseromsundervisning?

10. Hva synes du kunne blitt gjort annerledes for å gjøre det enklere for lærere i store endringsprosesser, som f. eks. fagfornyelsen?

11. Andre kommentarer?

B.6 Questionnaire P3

Evaluering av forkunnskaper:

Spesifikasjon av spørsmål brukt i spørreskjema for å lære om deltakernes bakgrunn og forventninger i programmering.

Spørsmål	**Type svar**	**Alternativer**
På hvilket trinn underviser du?	Velg ett eller flere alternativer	Videregåendeskole
		Ungdomsskole
		Voksenopplæring
		Barneskole
		Andre (fritekst)
Hvilke(t) fag underviser du i?	Velg ett eller flere alternativer	Realfag
		IKT
		Språkfag
		Elektrofag
		Estetiske fag
		Informasjonsteknologi
		Samfunnsfag
		Teknologi og design
		KRLE
		Programmering
		Media og Informasjon
		SNO
		Kroppsøving
		Andre
Har du undervist i programmering tidligere?	Velt ett alternativ	Ja
		Nei
Undervises det i programmering på din arbeidsplass?	Fritekst svar	Fritekstsvar
Er det flere fra din arbeidsplass som skal ta dette kurset?	Velg ett alternativ	Ja, noen jeg kommer til å samarbeide med i kurset
		Vet ikke
		Ja
		Nei
		Ja, noen jeg kommer til å samarbeide med i kurset, Ja
Hvor interessert er du programmering?	Velg ett alternativ. 1 = ingen interesse, 10 = veldig interessert	1
		2
		3
		4

		5
		6
		7
		8
		9
		10
Er du kjent med noen av følgende platformer/programmeringsspråk for barn/unge?	Velg ett eller flere alternativer	Ingen
		LEGO Mindstorm
		Pascal
		Visual Basic
		C
		Scratch
		C++
		Arduino
		Micro:Bit
		Logo
		Blockly
		Swift Playgrounds
		Python
		Andre
Er du kjent med programmering?	Velg ett alternativ	Ja
		Nei
I hvilken grad er du kjent med Python som programmeringsspråk?	Velg ett alternativ. 1 = ingen interesse, 10 = veldig interessert	1
		2
		3
		4
		5
		6
		7
		8
		9
		10
Er du kjent med andre programmeringsspråk? I så fall, hvilke?	Fritekst svar	Fritekstsvar
Hva er hovedgrunnen til at du meldte deg opp til kurset? Velg alle alternativene som passer.	Velg ett eller flere og/eller fritekst svar	Personlig interesse for temaet
		Ønsker å bli stødigere i temaet
		Mulighet for undervisning i nye fag

		Har behov for å få formalisert eksisterende kompetanse
		Programmering kommer inn i fagene som jeg underviser i
		Behov for å lære om hvordan programmering bør foregå i skolen
		Krav fra arbeidsgiver om videreutdanning
		Har fag hvor programmering kommer inn som del av pensum
		Vil stå bedre rusta nå i fagfornyelsen
		Nye læreplaner krever kompetanse i programmering
		Programmering i fagfornyelse (matte og fysikk).
		Behov for programmering i fag etter fagfornyelsen.
		Være bedre rustet til fagfornyelsen
		Ønsker å bli bedre på didaktikk for programmering
		Andre
I hvor stor grad forventer du å få bruk for kunnskap/ferdigheter fra kurset i din jobb?	Velg ett alternativ. 1 = ingen interesse, 10 = veldig interessert	1
		2
		3
		4
		5
		6
		7
		8
		9
		10
Hvilke forventninger har du til kurset?	Fritekst svar	Fritekstsvar

B.7 Questionnaire P6

End course evaluation (IT6204)

Specification of questions used in end-course questionnaire (P6).

Question	**Type of question**	**Response options**
Communication	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1
		2
		3
		4
		5
Webinars (live)	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1
		2
		3
		4
		5
Webinars (videos)	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1
		2
		3
		4
		5
Modules		
Exercise 1	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1
		2
		3
		4
		5
Exercise 2	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1
		2
		3
		4
		5
Exercise 3	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1
		2
		3
		4
		5
Exercise 4	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1
		2
		3
		4
		5
Problem identification	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1

		2
		3
		4
		5
Project	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1
		2
		3
		4
		5
Workload	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1
		2
		3
		4
		5
Self-Confidency	Likert scale (1 = Strongly Disagree, 5 = Strongly Agree)	1
		2
		3
		4
		5
Other comments	Freetext	Freetext

B.8 Questionnaire P8

REFLEKSJONSNOTAT H20

Besvarelsen (langsvarsdelen) skal være på 1-3 sider, i normal font og skriftstørrelse.

OBS! Merk at det er to spørsmålskateorier: Én med skala-svar og en langsvar.

Leveres på Blackboard under «Arbeidskrav» **innen 19.11**.

Husk å levere dokumentet av-identifisert (se eget dokument med instruksjon).

Dette refleksjonsnotatet ønsker vi å benytte i forskning på utdanning internt hos oss på NTNU. For mer informasjon om prosjektet kan du kontakte Majid Rouhani (majid.rouhani@ntnu.no).

Hvis du godkjenner at dine svar brukes (anonymisert) i denne forskningen kan du krysse av her:

☐ Jeg godkjenner at mine svar benyttes i dette forskningsprosjektet

1. Flervalgsoppgave

Sett et kryss for hvert utsagn, og vurder det med tanke på din egen situasjon/undervisning.

Utsagn	**Helt Uenig**	**Uenig**	**Hverken eller**	**Enig**	**Helt Enig**
Programmering har ikke vært et enkelt emne for meg å lære	☐	☐	☐	☐	☐
Programmering krever riktig forståelse av abstrakte begreper (f.eks. funksjoner/metoder, osv.)	☐	☐	☐	☐	☐
Jeg har læringsproblemer på grunn av fagets natur.	☐	☐	☐	☐	☐
Jeg tror jeg vil ha utfordringer med å utforme leksjoner slik at de vil være gunstige for elevene mine.	☐	☐	☐	☐	☐
Jeg tror det kommer til å bli spennende å undervise i programmeringskurs.	☐	☐	☐	☐	☐
Jeg kunne lite eller ingen programmering FØR kursstart	☐	☐	☐	☐	☐
Jeg føler at jeg har lært nok programmering nå til å kunne undervise mine elever.	☐	☐	☐	☐	☐

	Vanskelig	Litt vanskelig	Hverken enkelt\ vanskelig	Ganske enkelt	Enke lt
Å finne feil i programmer er	☐	☐	☐	☐	☐
Å dele en oppgave i del-problemer (program-funksjoner) er	☐	☐	☐	☐	☐
Å bruke biblioteker i programmer er	☐	☐	☐	☐	☐
Å forstå hva variabel-initialisering er er	☐	☐	☐	☐	☐
Å forstå hva løkker er er	☐	☐	☐	☐	☐
Å forstå hvordan løkker skal anvendes er	☐	☐	☐	☐	☐
Å forstå hva en valg-setning er er	☐	☐	☐	☐	☐
Å forstå hvordan en valg-setning skal anvendes er	☐	☐	☐	☐	☐
Å lage et program for å løse en bestemt oppgave er	☐	☐	☐	☐	☐

2. Langsvarspørsmål

a. Din læring

i. **Hva er det du opplever som utfordrende å lære (inkludert språkkonstruksjoner, programmeringskonsepter, problemer knyttet til prosessen), og hvorfor?**

Svar:

ii. **Hvor utfordrende tror du det er å lære et annet programmeringsspråk etter at du har lært Python, og hvorfor?**

Svar:

b. Dine eleveres læring

i. **Hvilke deler av programmering tror du er mest utfordrende <u>for elevene dine å lære</u>?**

Svar:

c. Undervisning i programmering

Hvilke utfordringer mener du vil kunne oppstå ved å:

a.

i. **lære elevene dine spesifikke programmeringskonsepter**
ii. **lære elevene dine spesifikk språkkonstruksjon**
iii. **motivere elevene dine til å lære**

Svar:

b. **Andre refleksjon er rundt utfordringer med læring og undervisning i programmering?**

Svar:

Appendix C

Program codes/scripts

Code listing C.1: Python code used as macro in excel to search the Internet and download abstracts of papers based on their DOI

```
import sys
import urllib.request
import re

START = "<div class=\"c-article-section__content\" id=\"Abs1-content\">"
END = "/div>"
SEARCH_STR = START + "(.*?)" + END

def cleanhtml(raw_html):
    CLEANR = re.compile('<.*?>')
    cleantext = re.sub(CLEANR, '', raw_html)
    return cleantext

def read_html():
    if len(sys.argv) == 1:
        print("Please specify url of the paper")
        sys.exit(0)

    url = sys.argv[1]
    req = urllib.request.Request(url)
    resp = urllib.request.urlopen(req)
    respData = resp.read()
    html_abstract = re.findall(SEARCH_STR,str(respData))
    abstract = cleanhtml(html_abstract[0])
    return abstract
```

Code listing C.2: Visual Basic script (macro) for highlighting keywords in excel

```
Sub mark_keywords(keyword, thisColor)
        Dim loc As Range
        Dim last As Range
        Worksheets(1).Activate


        With Worksheets(1).Range("C:C")
                Set loc = .Find(what:=keyword, lookat:=xlPart, MatchCase:=False)
                If Not loc Is Nothing Then
                        FirstLoc = loc.Address
                        Do Until loc Is Nothing

                                mark_all loc, keyword, thisColor

                                Set loc = .FindNext(loc)
                                ThisLoc = loc.Address
                                If ThisLoc = FirstLoc Then
                                        GoTo end_search
                                End If
                                Set last = loc
                        Loop
                End If
        End With
        Set loc = Nothing
end_search:
End Sub

Sub mark_all(loc, keyword, thisColor)
        str_pos = InStr(loc, keyword)
```

```
        Do Until str_pos = 0
                If str_pos > 0 Then
                        loc.Characters(str_pos, Len(keyword)).Font.Color = thisColor
                        str_pos = InStr(str_pos + Len(keyword), loc, keyword)
                End If
        Loop
End Sub
```

Appendix D

Examples of reflection notes

Examples of reflection notes from both cources in the program.

D.1 Examples of reflection notes

IT6203 – Example 1

Refleksjonsnotat 1:

Hva sier læreplanen om programmering, og hvordan kan det du har lært til nå i kurset støtte opp under dette?

2: Hvilke læringsmetoder motiverer og engasjerer dine elever, og hvordan kan disse benyttes i forbindelse med læring av programmering?

3: På hvilken måte kan programmering være nyttig for deg i din egen undervisning?

Fra høsten 2020 trer Fagfornyelsen i kraft, og med den mange endringer. Teknologi får en fremtredende plass i den nye overordnede delen av lærerplanen, men frem tid den trer i kraft gjelder den generelle delen av planen. Den generelle delen er videreført fra R94 og L97. Etter at det i Norge (i 2013) ble sett et behov for å legge til rette for at barn og unge ikke bare er i stand til å bruke, men også skape digitalt innhold og digitale tjenester, kommer nå programmering inn i den nye lærerplanens overordnede del. Naturfag skal bli et mer utforskende og praktisk fag, og faget får en tydelig teknologidel. Programmering kommer inn i flere fag. Samfunnsfag får et spesielt ansvar for digitale ferdigheter.

Det jeg har lært i kurset så langt er nok til at jeg kan sette elever i gang på egen hånd, og til dels veilede litt tror jeg. Jeg føler meg ganske på tynn is enda, men jeg vet nok til at jeg kan veilede elever til sider hvor de kan utforske mer på egenhånd og støtte opp med det lille jeg kan.

Jeg underviser i programfaget Service og Samferdsel (fra høsten 2020 – Salg, service og reiseliv), og i faget markedsføring og ledelse. Jeg er en stor «fan» av ungdomsbedrift som pedagogisk metode, og ser at jeg i større grad enn tidligere kan få et økt fokus på teknologi og nye løsninger blant elevene. Jeg holder også på med en masterstudie i multimedia og læringsteknologi, og mener at jeg med dette og programmerings studiet vil bli en verdifull medarbeider på skolen. Det er ikke så mange kolleger som har undervisningskompetanse i programmering, så jeg håper virkelig at jeg får dette til.

I den nye lærerplanen fremmes også tverrfaglighet, og jeg ser at kompetanse innen programmering kan bli nyttig i så måte. Med ungdomsbedrift som metode ligger det til rette for at vi for eksempel kan samarbeide med naturfag- og mattelærere i utvikling av nye produkter som bedriftene kan selge. Jeg føler at programmering i skolen vil føre til utforskende, kreative og nysgjerrige elever, noe som i seg selv fremmer dybdelæring. Jeg tror også at elevene vil oppleve skolehverdagen mer relevant for fremtiden og fremtidens krav om digital kompetanse.

Hvis jeg klarer dette studiet vil jeg være i stand til å lære elevene for eksempel å lage enkle spill, noe vi kan benytte i undervisning og for å fremme læring. Det hadde vært kjempegøy!

IT6203 – Example 2

Innledningsvis vil jeg si at oppgaveformuleringen er noe uklar: Det er ikke oppgitt hvilken læreplan man skulle bruke? Derfor valgte jeg å ta utgangspunkt i *Forsøkslæreplan i programmering og modellering X - programfag i utdanningsprogram for studiespesialisering (PRM1-01)*. Jeg kommer også til å se litt på programmeringen i de nye læreplanene.

1: Hva sier læreplanen om programmering, og hvordan kan det du har lært til nå i kurset støtte opp under dette?

Dagens samfunn er bygget på digital teknologi: her snakker vi om alt fra høyteknologiske systemer til vanlige hverdagsgjenstander. Ifølge læreplanen er programmering en viktig forutsetning for å sikre at framtidige medborgere skal være i stand til å videreutvikle denne teknologien og være med på å kunne sikre samfunnets velferd. Programmering og modellering skal gi innføring i algoritmisk tankegang samt i numeriske og analytiske metoder i realfag. Ikke overraskende står matematikken sentralt i læringsprosessen. Ifølge læreplanen er utforskingen innenfor realfagene et viktig formål med programmeringen. I tillegg skal programmeringen utvikle elevenes kreativitet, kritisk sans og metodeinnsikt.[1]

I de nye læreplanene blir programmering en del av flere fag, f.eks. matematikk, naturfag, kunst og håndverk, musikk og samfunnsfag.

Programmering skal inn i skolen, dette er det ingen tvil om. Det betyr at alle (realfags)lærere trenger grunnleggende programmeringskompetanse. Studiet jeg tar nå gir en innføring i grunnleggende ferdigheter innenfor et programmeringsspråk (Python). Gjennom

[1] https://www.udir.no/kl06/PRM1-01

oppgaveløsning og refleksjon får jeg mulighet til å utvikle algoritmisk tankegang. Jeg er ganske fersk innenfor programmering og merker at det kan til tider være utfordrende å løse oppgavene. Dette, mener jeg, er en viktig erfaring å ta med. Når det kommer til undervisningen i programmering kommer jeg til å møte elever med forskjellige forutsetninger, interesser og forkunnskaper. Erfaringene jeg erverver i løpet av dette studieåret kan hjelpe meg til å forstå elevene bedre, planlegge og tilpasse undervisningen.

2: Hvilke læringsmetoder motiverer og engasjerer dine elever, og hvordan kan disse benyttes i forbindelse med læring av programmering?

Jeg er lærer på en videregående skole, og underviser i medier og kommunikasjon, norsk og matematikk. Selv om MK er blitt et studieforberedende program, skiller elevgruppen seg betraktelig fra dem som tar et vanlig studieforberedende program. MK-elevene er vant til å være i aktivitet, til å jobbe prosjekt. De er kreative, men sliter med å bearbeide store mengder av teori. Ved planlegging av undervisningen skal jeg først og fremst ta utgangspunkt i at alle elevene skal føle mestring. Programmering kan være veldig frustrerende. Derfor er det viktig å legge opp til at alle kan få til noe. Jeg kommer til å starte med korte prosjekter som gir synlige resultater raskt. I oppstarten tenker jeg å bruke fysiske øvelser som tar i bruk algoritmisk tenkning. Jeg har også lyst til å bruke programmering i tverrfaglige prosjekter med MK-fagene slik at elevene ser nytte av denne kompetansen.

3: På hvilken måte kan programmering være nyttig for deg i din egen undervisning?

Som nevnt underviser jeg i medier og kommunikasjon, norsk og matematikk. Programmering kan være nyttig i alle de ovennevnte (og flere andre) fagene.

Hverdagen vår blir stadig mer digitalisert. Vi har tillit til det digitale systemet uten å virkelig forstå det. Derfor mener jeg at grunnleggende kunnskaper i programmering er en viktig del av en allmenkompetanse. Programmering har et stort potensial til å styrke fagene. Den er ikke bare koding, men en tankemåte. Alt vi foretar oss kan sees på som et problem, og alt vi gjør til problemet er løst er programmering. Her sikter jeg mot algoritmisk tenkning. Denne kan brukes uansett fag. Mer enn dette bruker vi algoritmisk tenkning bevisst eller ubevisst i hverdagen vår, f.eks. når vi deler et problem opp i mindre biter, når vi føler regler i spill, eller når vi utarbeider en fremgangsmåte.

Her er noen eksempler på hvordan programmering kan brukes:

- **MK** – se på samfunnsmessige utfordringer og muligheter som er knyttet til vår digitale hverdag; utvikle algoritmer som kan brukes i medieproduksjon.
- **Norsk** – utvikle algoritmer som kan brukes for tekstanalyse.
- **Matematikk** – løse matematiske oppgaver; utvikle bevissthet rundt utfordrende matematiske konsepter, f.eks. bruk av variabler; bruke algoritmer for å løse tekstoppgaver.

Programmering og algoritmisk tenkning er med på å gjøre elevene bevisste og nøyaktige når de jobber med forskjellige oppgaver i de ulike fagene. I tillegg handler programmering om å skape. Det finnes mange veier til en løsning. Derfor er programmering med på å utvikle elevenes kreativitet.

I fagene jeg underviser i kan programmering brukes som verktøy til å utvikle en bedre forståelse av og bevissthet over hvordan algoritmer påvirker hverdagen vår, at det finnes flere veier til en løsning, og ikke minst utvikle elevenes kreativitet.

Kilder:

- Forsøkslæreplan i programmering og modellering X - programfag i utdanningsprogram for studiespesialisering (PRM1-01) https://www.udir.no/kl06/PRM1-01

- Algoritmisk tenkning https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/

IT6203 – Example 3

Refleksjonsnotat

1. Hva sier læreplanen om programmering, og hvordan kan det du har lært til nå i kurset støtte opp under dette?

Blant kjerneelementene i matematikk, er det i tredje utkast blant annet et punkt som heter "Utforsking og problemløsning". I beskrivelsen av dette refereres det til viktigheten av algoritmisk tenking i prosessen for å utvikle strategier og fremgangsmåter for å løse ulike problem. Udirbloggen kobler algoritmisk tenking sammen med programmering, og argumenterer for at opplæring i programmering vil kunne være med å øke forståelsen av algoritmisk tenking.

Det kan være nødvendig å definere nærmere hva algoritmisk tenking er for å koble dette til programmeringskurset. Udir bruker en del nøkkelbegrep for å beskrive den algoritmiske tenkeren.

Et av disse er logikk, prosessen med å analysere å forutse kan trolig kobles til prosessen med å planlegge et program. Det å lage et flytskjema eller en pseudokode kan være en måte man kan koble til det å løse et problem f.eks. i matematikk. Man bryter problemet ned i mindre biter og lager en plan på rekkefølgen disse må løses i.

Det samme gjelder nøkkelbegrepet dekomposisjon. Litt av det samme som nevnt over. Det å lage et program dreier seg blant annet om å spørre seg hvilke kommandoer og funksjoner man må bruke for å lage programmet. I f.eks. et geometrisk problem kan det samme gjelde, hvilke lengder eller vinkler må regnes ut for å komme et steg nærmer målet for oppgaven.

Et tredje begrep er evaluering. Når et program er ferdig, behøver man å prøve det ut for å se om det fungerer som ønsket. Dersom det ikke fungerer må man analysere hvorfor og jobbe systematisk gjennom programmet for å finne eventuelle feil. Det samme vil kunne gjelde i matematikken. Dette er noe som basert på egne erfaringer ofte kan være en manglende ferdighet hos de fleste elever, kanskje det å jobbe med programmering kan ha en overføringsverdi i forhold til dette.

2: Hvilke læringsmetoder motiverer og engasjerer dine elever, og hvordan kan disse benyttes i forbindelse med læring av programmering?

Læringsmetoder vil ofte være individuelt betinget. Det finnes flere jeg på dette tidspunkt har vanskelig for å knytte til programmering.

Noe som kan være bra, er at det ikke nødvendigvis finnes et fasitsvar på hvordan man skal lage et program. Å lage et program kan sammenlignes med en problemløsingsoppgave i matematikk. Det finnes ikke nødvendigvis én riktig fremgangsmåte, men det er mulig å løse oppgaven på ulike måter. Slike oppgaver kan virke motiverende på mange elever.

Programmering kan funke fint som et gruppearbeid. Elevene kan jobbe sammen om å lage et program, utveksle ideer og diskutere løsninger. De kan også forklare sine ideer til hverandre, det å forklare andre er en fin måte å lære på.

Når elevene lærer programmering, kan det sikkert være lurt å gå gjennom en del teoretisk stoff, som ulike kommandoer og funksjoner. For å mestre dette, kreves det øving, det å sitte å programmere er en fin aktivitet der elevene øver på å bruke kunnskapen. I enkelte teoretiske fag er det ikke alltid like lett å få til gode øvinger, noen ganger kan det f.eks. være tekstoppgaver som en del kan synes er kjedelig. Sammenlignet med det, vil programmering være praktisk rettet, elevene gjør noe praktisk for å lære, og det vil jeg tro vil være motiverende. Derimot vil det kanskje også være demotiverende dersom elever får mye feilmeldinger de ikke klarer tolke og ikke opplever mestring. Det kan tenkes at det kan bli en utfordring, som det jo kan være i andre sammenhenger på skolen også.

3: På hvilken måte kan programmering være nyttig for deg i din egen undervisning?

Dette spørsmålet antar jeg er uavhengig av kompetansemål i nye læreplaner, da det jo i tilfelle er programmeringen i seg selv som vil være en del av formålet.

Det kan tenkes at programmering vil kunne øke matematisk forståelse i noen tilfeller jeg har reflektert over. Jeg har blant annet programmert andregradsformelen. Enkelte andregradsligninger har to løsninger, enkelte har én, og enkelte lar seg ikke løse. Dersom man ikke tar hensyn til dette i programmet vil man kunne få ulike feilmeldinger. Kanskje det at man må endre programmet for å få dette riktig, kan være med å øke forståelsen av nettopp hvorfor noen ligninger har to, en eller ingen løsninger.

Man kan lage enkle programmer for ulike matematiske formler, f.eks. arealformler med input av ulike variabler. Kanskje dette også kan være nyttig i å forstå hvordan man setter tall inn i formler, eller forståelsen av å bruke generaliserte formler i seg selv.

En del logiske resonnementer virker å kunne være overførbare til matematisk tenking. Det er litt vanskelig å sette fingeren på, men det har slått meg når jeg har jobbet med if-else oppgaver. Enten skjer dette, eller så skjer dette, eventuelt flere elif. Som nevnt har jeg ikke oppdaget noen direkte overføringsverdi annet den logikken bak tankegangen.

IT6204 – Example 1

Forprosjektet er i grunn to prosjekter, som ikke trengs å kjøres sammen. En kan enten kjøre den delen hvor en innhenter data ved bruk av Micro: bit eller en kan programmere en numerisk løsning av en andre ordens differensiallikning. Det er i grunnen to avhengige prosjekter. Men ved å kjøre prosjektene sammen vil kanskje elevene kunne reflektere litt rundt en ren teoretisk løsning, og hva som skjer i virkeligheten. I virkeligheten er det mange faktorer som påvirker resultatet. Dette kan skape en fin diskusjon, rundt en sammenligning av prosjektet. Men også dersom elevene har en god bakgrunn i programmering vil de kanskje kunne finne en god tilpassing til datasamlingen, og dermed utfra dette anslå luftmotstanden. Jeg er personlig ikke der enda, men vil gjerne jobbe for å finne en måte å anslå dette. De to prosjektene vil jeg tro vil kunne gjennomføres med stor suksess i klasserommet, men de to prosjektene har også veldig ulik vanskelighets grad.

Prosjektet hvor en programmerer Micro:bit til å innhente data fra en svingning er enkelt å gjennomføre (så lenge en får mu til å flashe programmet – der har jeg hatt en del problemer), og kodingen som gjennomføres for å plotte de innhentede data er relativt enkelt. Dersom elevene har litt programmeringsferdigheter vil denne delen være både enkelt og moro å gjennomføre, da de kan se praktisk bruk av programmering. Derimot vil det krevet litt forarbeid dersom elevene ikke har noen programmeringsferdigheter, men jeg anser dette som overkommelig. På sikt vil forhåpentligvis alle ha en del programmeringsferdigheter når de kommer på videregående nivå, og da bør dette være et relativt tidseffektivt praktisk programmeringsarbeid.

Å skrive et program som løser andre ordens differensiallikninger ved bruk av Eulers metode krever mer programmeringsferdigheter og forståelse. Her kreves det at elevene har programmert før. Personlig føler jeg at denne delen var litt vanskelig, men at den var fullt overkommelig etter jeg hadde lest meg opp både på Eulers metode og sett noen alternative metoder for å løse dette i Python.

Dersom elevene er kapable til å løse andre ordens differensiallikninger i Python vil de også ha en del programmeringsferdigheter, og kanskje til og med kunne klare en bedre løsning på kurve tilpassing som jeg nevnte over. Da kan en få en god arbeidsperiode tverrfaglig i matematikk og fysikk, som forhåpentligvis gir god innlæring i både matematikk, fysikk og programmering. Men før en går i gang på med dette undervisningsopplegget er det altså viktig å vite noe om elevenes ferdigheter i programmering. En må kartlegge om elevene kan tenke logisk og har utviklet en algoritmisk tankegang.

IT6204 – Example 2

Valg av HTML som tema

I min erfaring har elevene lite erfaring med hva programmering er og hva det kan brukes til når de starter opp i 8. klasse. Alle er daglig inne på nettsider, så å programmere hvordan en nettside vil se ut kjennes relevant ut samtidig som det gir de en smakebit på å programmere med tekst og ikke blokkene som mange av de har vært innom på mellomtrinnet. Jeg valgte derfor å introdusere HTML i dette opplegget med tanke om at vi kan gå videre med CSS og programmering med JavaScript videre på ungdomsskolen. Siden naturfagsrapporter har en streng oppbygning og struktur blir det naturlig å se sammenhengen med hvordan de kan bruke HTML til å kode en ryddig og oversiktlig rapport.

Tilpasset opplæring

Etter å ha jobbet meg gjennom alle oppgavene viser det seg at dette er en relativt omfattende oppgave som vil løses i ulikt tempo og til ulik grad av elevene. Noen har kanskje jobbet med HTML fra før, andre vil ha utfordringer med å bruke og forstå appen og konseptene rundt programmeringen.

Oppgavene her kan tilpasses på flere ulike måter:

En av grunnene til at jeg valgte å spille inn videoer og bruke flipped classroom er at alle elevene har tilgang på videogjennomgangene og kan se på disse i eget tempo, så mange ganger de trenger, spole, pause og stille spørsmål underveis til læringspartner eller lærer.

De elevene som er selvgående og raske kan etter den første felles introduksjonen gå gjennom hele kursrekken uten å måtte vente, og hvis de har tid til overs er det åpent for å gå inn på w3schools eller Khan Academy og jobbe seg videre i HTML eller CSS på egenhånd.

For noen elever vil oppgavene være såpass omfattende i arbeidsmengde at de ikke vil ha mulighet til å gjennomføre alt på fire skoletimer. For elevene som sliter kan det kanskje være nok å gjennomføre de små kodeoppgavene og ikke kode rapporten. Eller evt. å kode rapporten med lister og tabeller, men uten lenker og bilder. Her har læreren mulighet til å gi individuelle tilpasninger ut fra hva de ser elevene trenger.

Programmering og iPad

Dette prosjektet var en av mine første erfaringer med koding på iPad. Det er ganske klart for meg at koding med tekst på iPad er en god del mer utfordrende enn på PC. Dette kan selvfølgelig være en vanesak, men å skrive kode er mye mer kronglete på iPad fra mitt ståsted. Jeg opplever at man er helt avhengig av å ha et eksternt tastatur til iPaden når man koder. Uten eksternt tastatur blir skjermflaten veldig liten og det blir vanskelig å holde oversikt over koden. En utfordring med våre eksterne tastaturer er at symbolene på tastaturet ikke samstemmer med symbolet du får opp på skjermen når du trykker på tastene. Dette gjør det vanskelig å finne symboler som f.eks. «», <>, = osv. Løsningen som jeg anbefalte elevene mine i videoen er å i stor grad kopiere kode som er tilgjengelig og endre innholdet i taggene.

Når det gjelder appen JS Anywhere synes jeg at den fungerer godt. Den er oversiktlig og enkel å bruke, har ikke for mange distraksjoner i selve appen, kan lenke til nettsider utenfor appen og bilder kan enkelt hentes inn fra kamerarullen og settes inn. En funksjonalitet jeg kunne ha ønsket meg er at den gir automatiske innrykk når kode skrives innenfor en tag. Da er det enklere å holde koden ryddig.

PRIMM-metoden

Det var litt tilfeldig at jeg kom over PRIMM-metoden, og jeg endte opp med å ønske å teste den ut fordi den prøver å sette fokus på tolkning og forståelse av koden før elevene selv skal skrive kode. I oppgavene prøvde jeg å få til dette på litt ulike måter med en kombinasjon av spørsmål og at elevene skulle fylle ut tabeller for å gi litt variasjon for å holde motivasjon oppe. Her må elevene til å tenke gjennom koden før de kaster seg inn og kopierer den, og det har jeg troen på at vil gi bedre forståelse.

Mye av prosjekttiden min har blitt brukt til å tenke ut oppgaver som gir elevene et læringsbytte, men der oppgavene er små nok til at hver oppgave ikke tar lang tid å løse før de kan se et resultat av koden sin. Det har vært en veldig lærerik prosess for meg, og jeg innser at det er mulig at jeg har vært litt ambisiøs i oppgaven med å skrive hele rapporten i HTML, fordi det er veldig mange avsnitt og mye tekst som skal legges inn. Et alternativ som jeg kommer til å vurdere når jeg prøver ut opplegget med elevene er å kun ha fokus på noen deler av rapporten og ikke en hel rapport.

Et punkt jeg reflekterte over underveis er at i de første delene (predict-run-investigate) ligger det en utfordring i at våre elever ikke har engelsk som førstespråk, som vil si at det ikke nødvendigvis er logisk for elevene at p står for paragraph som er det samme som et avsnitt. PRIMM-metoden har i stor grad blitt prøvd ut i engelskspråklige land der dette ikke vil være et problem. For å hjelpe elevene i denne delen lagde jeg ordlister sammen med alle kodebitene som oversetter begrepene i tagene vi skal bruke. Forhåpentligvis kan dette hjelpe elevene til å tolke og forstå koden riktig.

Jeg gleder meg til å prøve ut og videreutvikle opplegget. Jeg tror vi kan få nytte av slike typer prosjekter, gjerne i kombinasjon med simuleringer, når fagfornyelsen er i gang.

IT6204 – Example 3

Faget har gitt oss en dypere forståelse av hva grunnleggende konseptene er, og hvordan de kan anvendes til å løse problemstillinger innen de forskjellige fagområdene. Faget har hatt fokus rettet spesielt mot anvendelse av programmering innen elektronikk og roboter, men vi har også sett på programmering i et bredere perspektiv. Hvorfor skal alle lære seg programmering? Hvilke hindringer vi møter på veien og hvordan på en god måte vi kan overføre denne kunnskapen til våre elever.

Det er mange måter å gripe fatt den pedagogiske fremgangsmåten i dette faget. Men noe vi tror på er at elevene kan trenge en tydelig trinnvis oppdeling i startfasen. Samt at man tilrettelegger og underviser på det nivået eleven befinner seg på. Dette kan i første omgang virke ganske banalt. Men det er fort gjort å undervurdere kompetansen som enkeltelever med stor interesse for faget tar med seg inn i undervisningen. Man bør ha med seg ekstra utfordringer til de elever med en høy faglig modningsgrad. Noe vi legger tydelig merke til i dette faget er at lek og eksperimentering er nyttig og fruktbart og effektivt. Muligens er mindre lærerstyring, og mer egendrevet indre motivasjon noe å prøve ut som pedagogisk fremgangsmøte? Mulig vi har en for tydelig lærerstyrt prosess i vårt opplegg.

Gjennom dette kurset har vi levert inn 5 obligatoriske øvinger. Øvingene har utfordret oss på mange ulike felt. Spesielt nyttig vil jeg trekke frem Øving 1 der vi ble utfordret på å besvare hvorfor alle skal lære seg å programmere. Denne øvingen ga en fin teoretiske innsikt og for meg personlig overbevisning hvorfor programmering blir et viktig fagfelt i skolen i mange år fremover.

Den mest inspirerende øvingen syns jeg var øving 2 der vi laget en fartsmåler for en lekebil. Denne måten å lære hva vi kan bruke en datamaskin til syns jeg er genial. Denne Øvingen satt oss også litt på sporet av hva vi ønsket å fordype oss i eksamensdelen av kurset.

Den delen av kurset jeg tror kommer til å ta videre, er rett og slett samlingen av nettressurser i pensumdelen av kurset. Her var det veldig mye veldig konkrete gode tips som jeg må jobbe videre med etter at kurset er ferdig. Jeg har sett meg ut spesielt lenketipset om AppLab som en ting og ønsker å sette meg mer inn in.

Gode forberedelser til fremtidig undervisningen blir å holde seg oppdatert på gode lenker på nett. Jeg oppdager stadig interessante lenkesamlinger og utfordringen blir kanskje å velge ut gode fremfor å finne godt tilrettelagt undervisningsmateriell. Et lurt sted å starte syns jeg er på Kidsakoder og microbit.org. Her er et mye bra og samtidig godt organisert.

Jeg har også forstått at GitHub er ofte en brukt plattform for programmerere. GitHub må utforske mer nå i etterkant.

Webinarene i kurset syns jeg har fungert fint. En liten negativ tilbakemelding er at jeg syns det flytting av tidpunkt på webinarene virket noe lite gjennomtenkt. Derfor ble det for meg en god løsning å se webinarene i opptak i etterkant. Emnene som er valgt har en fin bredde og en god dybde innen de valgte temane. BRA! Den mest motiverende delen syns jeg var det webinare om Lær Kidsa å kode. Det mest tankevekkende webinare syns jeg var wbinar nr 2. The gender gap and how to address it. Menge tankevekkere for hvorfor det er viktig å få jenter inn i dette faget.

Læreboka Learner-Centered Design of Computing Education: Research on Computing for Everyone syns jeg er en grundig godt forklarende bok. Boka tar for seg sentrale emner på en ryddig måte. Et godt valg som passer fint inn i kurset.

Oppsummering.

Jeg syns kurset har vært fornuftig satt sammen av konkret teori og oppgaver, samt mere reflekterende emner som byr på modning innenfor fagfeltet. Selve eksamensoppgaven har gitt oss et godt og kraftig hjelpemiddel til videre opplæring av programmering i skolen. Tidsbruken jeg har brukt på selve eksamensdelen var for meg veldig meningsfull bruk av tid, da selve eksamensproduktet blir for oss et konkret undervisningsopplegg vi senere vil benytte oss av.

Appendix E

Transcripts / codes for data analysis

Transcripts and codes for data analysis

E.1 Codes used in P4 and P6

Codes used in P4 and P6

Sub-category	Freq	Files	Sample sentence
Teaching programming (147, 539)	**539**	**367**	
Positive towards own teaching	28	23	Og tiden mens elevene jobber, er ganske intens. Elevene sender spørsmål eller besvarelser som skal vurderes, gruppesamtaler skal følges opp, og tidsplanen skal følges. Men jeg erfarte at jeg hadde god kontakt med dem underveis, og jeg synes det var mulig å gi god oppfølging. (Fil PR_10040, kandidatnr. 10040)
Positive towards own project (lesson plan)	119	75	Jeg mener jeg har laget et opplegg som jeg tror kan fenge elevene samtidig som de lærer algoritmisk tenkning, løkker og lærer hvordan de kan lage programmer som løser likninger av første og andre grad. (Fil 10026_refleksjonsnotat, kandidatnr 10026)
Critical towards own teaching skills	4	4	Da vi var ferdig med IT6203, så skal jeg innrømme at jeg fortsatt ikke følte meg kompetent til å undervise i programmering. (Fra fil PR_10005, kandidatnr 10005)
Learning resources	121	69	En annen viktig lærdom er at elever som kan programmering kan være en stor ressurs når man underviser dette i en klasse. (Fra fil 10171_Prosjektoppfave IT6204)
Importance of experience	117	77	Det var fint å få mulighet til å lage et opplegg som vi selv kan ta i bruk ved vår skole neste år. Eksamensoppgaven har på den måten vært svært relevant for min arbeidssituasjon. (10073_Prosjektrapport)
Implementing lesson plan in school was challenging	51	37	Jeg synes også det har vært utfordrende å implementere dette i naturfag- og matteundervisningen. Jeg har måttet tenke helt nytt og på en annen måte enn jeg er vant til når jeg lager undervisningsopplegg, men dette har vært veldig spennende. (Fra fil 10055_Prosjektrapport, kandidatnr 10055)
Looks forwards to teaching programming	77	63	Programmering er ganske nytt for meg personlig også, men jeg gleder meg til å prøve det ut blant elevene. (Fil navn 10073_Prosjektrapport)

Worried about teaching programming in schools	22	19	Da vi var ferdig med IT6203, så skal jeg innrømme at jeg fortsatt ikke følte meg kompetent til å undervise i programmering. (Fra fil 10055_Prosjektrapport, kandidatnr 10055)
Didactical issues (130, 446)	**429**	**259**	
Adapted education	180	101	Men jeg valgte helt bevisst å gå for et opplegg som jeg håpet kom til å treffe ALLE elevene mine, ikke bare de flinkeste. (10092_Prosjektrapport)
Teaching method	116	68	Eg har brukt ACE saman med Viper-roboten når eg har arbeidd med prosjektet og prøvd å laga relevante oppgåver som elevane skal bruke for å komme igong med programmering av roboten. (10071_Prosjektrapport)
Focus of teaching	101	68	Dette for å fokusere bedre på det som er oppgvens kjerne, nemmelig å programmere et enkelt terningspill i JavaScript. (10023_Prosjektrapport)
Assessment	24	15	Jeg har ikke satt opp detaljerte kriterier2 med tanke på hva elevene skal kunne i programmering. Dette har jeg bevisst gjort da det ikke står noe om dette i læreplanen. Men jeg tolker læreplanen slik man kan bruke programmering som et verktøy for å utforske i matematikk. Det står under Underveisvurdering at «Elevane viser og utviklar òg kompetanse når dei utforskar og analyserer reelle datasett, og når dei gjer og argumenterer for funn.» (Utdanningsdirektoratet (2020) Dette er også grunnen til at jeg har valgt at vurderingssituasjonen skal være en presentasjon av det elevene har gjort. (PR_10008)
Gender gap	8	7	Et av mine hovedmål er å få flere jenter interesserte i programmering. I en forundersøkelse viste flere jenter større interesse hvis programmeringen kunne kobles opp mot biotiske faktorer som trening, kosthold og mat. Dette gav meg ideen til å lage et helautomatisk drivhus i kjelleren av skolen hvor lys, gjødsling og vanning ble overvåket av Arduino-enheter. (Personlig prosjektrapport 10041)
Plans for future (115, 237)	**237**	**169**	
Plans for future (coded directly at the top-level code)	32	22	Jeg ser også at både løkker, funksjoner og feilsøking kunne vært inkludert i utvikling av disse programmene, men kommer frem til at det ville vært veldig mye for elevene i det tidsperspektivet jeg hadde satt. Dette kan, med fordel, tas en annen gang som videreutvikling.

Can improve project (lesson plan)	89	64	Dette har ikke vært enkelt og det er stor sjanse for at jeg har bommet på nivået og enten laget det for lett eller for vanskelig. (10055_Prosjektrapport)
Changes to be implemented	40	30	Elevar vert fort lei av noko dei berre halvegs forstår og neste gong skal eg flette litt programmering inn langsmed siste del av presentasjonen, dei kan m.a. programmere dei små eksempelprogramma eg viste. (10038_Prosjektrapport)
Motivated to learn more	76	53	Min bakgrunn innenfor programmering for et år siden, var relativt smal med lite erfaringer, kunnskap og forståelse. Men da jeg fikk mulighet til å søke på videreutdanning i programmering, kjente jeg raskt på en motivasjon for å tilegne meg ny kunnskap og kompetanse som jeg vil få bruk for i fremtiden. (PR_10006)
Programming Skills (99, 220)	**220**	**174**	
Positive towards own learning of programming skills	92	68	Etter å ha brukt mykje tid på programmering det siste året, sit ein inne med grunnleggjande ferdigheiter i programmering, ei forståing av kva programmering er og korleis det skjer, og ein veit litt om kva moglegheiter og utfordringar emnet inneheldt. (10038_Prosjektrapport)
Perceives own programming skills as relatively low	4	4	Jeg syns ikke egen kunnskap her var stor nok til å kunne være en god veileder for elevene, og valgte å ikke ha med dette i undervisningsopplegget jeg har utviklet, selv om det kanskje kunne gjort det mer spennende. (10127.pdf)
Has learned basic programming	38	34	Selv har jeg lært veldig mye den siste tiden. Rent teknisk kunne jeg nok valgt å jobbe mye mer avansert med programmering av avanserte funksjoner, lister, Dictionarys osv. i Python. (10092_Prosjektrapport)
Increased programming skill	31	29	Etter hvert ble jeg litt stødigere i programmeringen og kunne lage egne forenklede eksempler. (PR_10043)
Programming is challenging	55	39	Det har gått mange timer med feilsøking og frustrasjon over ting som først virker fint, for å plutselig ikke fungere mere. I løpet av arbeidet med dette prosjektet har jeg også lært hvor lite som skal til for at ting ikke virker og hvordan det blir hensiktsmessig å koble opp alt sammen.
Attitudes towards programming in school (137, 517)	**509**	**312**	

Positive towards programming in school	92	64	Programmering kan ses på som et nyttig verktøy for å løse ulike problemer og utfordringer som vi står overfor, i tillegg til at det kan gi oss en bedre fagforståelse.
Critical towards programming in school	7	6	Matematikkfaget har gjennom fagfornyelsen fått ansvar for opplæringen av programmering i skolen, og en del kritiske stemmer har vært redde for at programmeringsdelen kan bidra til at det blir mindre tid til det matematikkfaglige. (prosjektrapport)
Positive towards using programming interdisciplinary	167	97	På høyere nivå er programmering relevant for både geografi og samfunnsvitenskap. Dette er særlig aktuelt for kartografi og geografiske informasjonssystemer, men blir også mer og mer aktuelt i samfunnsfagene.
Community and collaboration	153	76	Det har vært et mål at opplegget skal kunne brukes av andre lærere, og det var derfor viktig at beskrivelsen viser både hva vi skal gjøre, men også hvorfor! Begrunnelsen er sentral for at læreren skal kunne bruke opplegget som en bro mellom ulike representasjoner av vekstfart, og dermed hjelpe elevene å øke sin matematiske forståelse.
Norwegian context	90	69	Som jeg nevner i selve undervisningsopplegget, vil de fleste elevene per dags dato ha liten kompetanse i programmering, men det tar nok ikke mange år etter fagfornyelsens inntog før elevene kan dette på et høyere nivå enn meg. (10055_Prosjektrapport)
External challenges (153, 578)	**578**	**379**	
Math in programming	106	57	Elevenes varierende forkunnskaper i matematikk vil også ha betydning for den didaktiske tilnærmingen til opplegget, da en god del av oppgavene inneholder både matematiske algoritmer og -begreper. (PR_10025)
Pupil’s digital competence	122	81	Jeg har ingen kjennskap til hva de kan av programmering fra før. (10026_refleksjonsnotat)
COVID-19	140	95	På grunn at smittesituasjonen av viruset covid-19, så har det vært utfordringer rundt gjennomføringen av undervisningsopplegget. (10129_Prosjektrapport)

Time is an issue	126	77	Det har vært både moro, interessant og frustrerende å gjennomføre kurset. Moro fordi jeg faktisk har fått til en del og lært mye. Interessant fordi jeg lenge har vært nysgjerrig på programmering, hva det egentlig er og hvordan ulike programmer er bygget opp og fungerer. Frustrerende fordi jeg synes deler av programmeringa har vært vanskelig, og jeg har brukt mange timer på å få til enkelte av øvingsoppgavene. (PR_10050)
Technical issues	42	30	Den datamaskinen jeg har tilgjengelig er en chromebook. Jeg kan dermed ikke laste ned Python eller PyCharm. Jeg har derfor valgt å løse de tekstbaserte programmeringsoppgavene i Jupyter. (PR_10044)
School resources	22	20	Elevane kunne ikkje bruke dei micro:bitane vi har på skulen, heller kunne ikkje dei bruke dei datamaskinane eg hadde klargjort. Derfor måtte eg gjere nokre ting på ein anna måte. Heldigvis er det mykje tilgjengelege ressursar og verktøy på internett som eg kunne bruke for å gjennomføre undervisningsopplegget mitt.
Challenges surrounding continuous technological innovation	20	19	Roboten er ny på skulen vår, og vi har ikkje nokon i personalet som har kompetanse i denne roboten.
Course specific reflections (84, 141)	**141**	**104**	
Positive feedback	90	65	Samtidig har det vært utfordrende å skulle studere i tillegg til å jobbe 100%, spesielt de siste ukene når vi har drevet med hjemmeskole, og som har tatt mye tid. Dette har krevd struktur og disiplin fra min side, og jeg er veldig fornøyd med å snart være helt i mål. (PR_10006)
Negative feedback	10	9	Da vi var ferdig med IT6203, så skal jeg innrømme at jeg fortsatt ikke følte meg kompetent til å undervise i programmering. (PR_10005)
Course required a lot of resources	41	30	Kurset har gjort meg betre i stand til å sjå moglegheiter i staden for utfordringar ved å nytte programmering i ulike fag. Eg har no mange idear, ikkje berre til praktiske prosjekt men og meir komplekse program i Python for R-matte elevane mine. (10038_Prosjektrapport)
References (27, 65)	**65**	**45**	

Learner-Centered Education	24	19	Dette spørsmålet dreier seg også rundt et av de mest sentrale begrepene i dette kurset, nemlig «Learner-Centered Design», som enkelt sagt går ut på at man for alvor skal ha den lærende som utgangspunkt for hvilket design man velger i undervisningen. Dette står slik jeg forstår Guzdial (2015) som en motsetning til «Expert-centered design» som i stedenfor å ha den lærendes perspektiver i større grad fokuserer på hvor man vil de skal ende, og hva de MÅ lære seg, sett med eksperters øyne. (PR_10016)
Mark Guzdial, 2015	40	25	I det didaktiske grunnlaget for opplegget, i dokumentet med undervisningsopplegget, henviser jeg hyppig til Guzdial, der jeg finner spesielt det første kapittelet å være fullt av ideer som appellerer til min forståelse av min yrkesutøvelse. Dersom dette hadde vært en mer tradisjonell akademisk oppgave ville det vært naturlig å gå mer i dybden om resten av boka. (PR_10003)
Programmering for alle	1	1	Kurset og arbeidet med denne oppgaven har gitt meg en bredere forståelse av hva «programmering for alle» og «programmering i skolen» egentlig betyr. Det handler om så mye mer enn bare det å kunne koding. Programmering i skolen handler også om likestilling i forhold til kjønn og minoriteter. (10108-Prosjektoppgave)
Pupil's experience (111, 212)	**211**	**142**	
Pupil's positive experience	77	39	Det har vært gøy å få til kodingen, og barna ble svært ivrige både i gangespillet og i reaksjonsspillet.
Pupil's negative experience	21	14	Det viste seg og (ikkje uventa) at nokre vart litt frustrerte over nytt stoff og nye tenkjemåtar når dei skulle prøve sjølv («Kva skal eg no med dette då…»). (10038_Prosjektrapport)
Unable to complete program	113	89	Selv om skolene er åpnet igjen, så er det ikke mulig å gjennomføre realfagsuka nå siden fokuset er å få avsluttet skoleåret og gjennomført nødvendige vurderinger. Siden jeg ikke fikk gjennomført opplegget som planlagt, så må jeg derfor tenke meg til hvordan de ulike delene av undervisningsopplegget kommer til å fungere.
			(10026_refleksjonsnotat)

E.2 Codes used in P5

Codes used in P5

Name	Files	References	Description/Sample
Arguments	0	0	Arguments for why everyone should learn programming, or why we should teach programming in school
Abstract Thinking Capability	40	41	Old enough to grasp abstract concepts. Mostly used as an argument for introducing programming at ungdomsskole or VGS.
Cheaper	9	9	Argues that it is cheaper to introduce programming at a certain education level. Mostly used as an argument to introduce at VGS, but also at ungdomsskole.
Counter Arguments	0	0	Counter arguments to why everyone should learn programming. Note: In general few counter arguments, as the exercise explicitly asks for arguments, not counter arguments. Not as widely used as first imagined.
Expensive	1	1	Argues that it is expensive to teach everyone programming
Not Engaging	0	0	Programming is not engaging
Not Relevant	2	2	Programming is not relevant to future career or education
Too Difficult	2	2	Programming is to difficult to learn
Creativity	15	16	Programming lets you be creative
Innovation	2	2	Programming is innovative, which can lead to new inventions or appeal to certain learners.
Interesting, Fun or Exciting	25	31	Programming is interesting, fun or exciting
Owns Computer	3	3	Argues that because everyone at their chosen education level owns their own computer, it will be cheaper and easier to start learning programming at this education stage.
Prepare For Education	24	26	Learning programming at a certain education stage will prepare learners for what is facing them at the next education level, or may invoke interest to study something involving programming.
Recruiting	31	35	Learning programming in school will recruit more people to a CS education.
Textbook	0	0	Textbook arguments. Guzidal uses mainly six arguments to why everyone should learn programming. Productivity and computational thinking is merged into ComputationalLiteracy, reducing total arguments to four.

Broadening Participation	135	239	[Book] Introducing programming in school will equalise the uneven distribution of programmers. Most programmers are typically white or asian males. The field needs more ethnical minorities and females.
Computational Literacy	150	299	[Book] Being computational literacy is like "learning to read", and an important skill to participate and understand the modern world. A basic understanding of programming, using computers as a form of expression and a way to think of computing.
Computational Thinking	137	194	[Book] Computational thinking (algoritmisk tenkning in Norwegian) involves solving problem, designing systems and understanding human behaviour. May be used to increase productivity in jobs, and also may be beneficial to apply in daily life.
Apply In Daily Life	29	30	[Book] Computational thinking may be beneficial in daily life. (Sub argument of computational thinking).
Interdisciplinary	112	166	[Book] Computational thinking may be applied as a tool to better understand other fields of study. May also be used by students as a tool to better understand subjects like physics, math etc.
Jobs	165	320	[Book] Programming is a good career, and the society needs more programmers. Knowledge of computing is necessary or helpful for other jobs than just professional software developers.
Learn About World	160	290	[Book] Learning about programming is important to understand the world around us. We are learning basic biology, physics and chemistry which is all around us, and it is only natural that we learn about computing and how the digital world around us works.
Productivity	115	139	[Book] Computational literacy may increase productivity when doing experiments and in certain jobs. Can reduce costs.
Tool To Understand	68	92	An argument to learning programming in school, is that is may be used as a tool to get a deeper understanding of other subjects.
Attitude Of Teacher	0	0	The attitude of teachers towards programming for all
Critical	14	25	Reflects on issues or arguments against CS for all, or disagrees with the arguments given by Guzidal (i.e. not relevant to all, does really everyone need it)

Positive	100	133	A subjectively positive statement towards programming, or a strong implicit indication of a positive view towards the subject.

E.3 Transcriptions P7

Transcriptions used in P7

Transkripsjon 1

00:03:38 Intervjuer
Hei der vet du der var du. Velkommen til intervju.

00:03:48 Lærer
Takk.

00:03:50 Intervjuer
Jeg ser at du har har jo signert og fått sendt tilbake samtykkeskjema, så da har du jo sikkert sett litt på infoen der og, men jeg kan jo ta litt kjapt om forskningsprosjektet likevel. Jeg heter Miriam da i forskningsassistent på det prosjektet her som handler om den overgangen fra lærer til å undervise programmering. Og intervjuene skal brukes da til å skrive en forskningsartikkel på den her overgangen da, og så skal det også brukes til å forbedre videreutdannings kursene. Som du sikkert ser oppe i hjørnet på vinduer, så blir intervjuet opp. Men det opptaket slettes så fort intervjuet transkribert da, og da vil også være helt anonymt. Ønsker du å få det transkriberingen tilsendt, så du kan se gjennom?

00:04:47 Lærer
Gjerne det, ja.

00:04:48 Intervjuer
Ja, da skal jeg bare notere meg det her.
Ellers så er jo intervjuet som jeg skrev i mailen firedelt. Og i de to midterste delene så kommer jeg til å vise deg noen ordskyer. Så er du forberedt på det. Har du noen spørsmål før vi begynner?

00:05:13 Lærer
Nei, forsåvidt ikke.

00:05:16 Intervjuer
Nei, det er bare å spørre underveis hvis det er noe du lurer på. Så da tenker jeg vi bare sette i gang.
Og del 1 er jo sånn veldig kort om din bakgrunn.
Så hvilke fag underviser du i?

00:05:28 Lærer
Underviser i matematikk, fysikk og teknologi og forskningslære for tiden.

00:05:32 Intervjuer
Ja, hvilke av de her fagene tror du at du kommer til å undervise programmering i?

00:05:37 Lærer
Jeg er vel pålagt å gjøre det i matematikk, og jeg blir pålagt å gjøre det i fysikk.
I teknologi og forskningslære, der har jeg brukt programmering i flere år allerede.

00:05:49 Intervjuer
Ja. Hvilke trinn underviser du på?

00:05:52 Lærer
Alle videregående trinn.
00:05:54 Intervjuer
Ja. Hvilke erfaringer hadde med programmering før deltakelse på programmeringskurs?

00:06:01 Lærer
Jeg har jo litt programmeringsbakgrunn i form av at jeg er utdannet sivilingeniør fra NTNU. Men den gang jeg tok programmering så var eksamen med papir og blyant, så det er jo den varianten av programmering som jeg har erfaring med.

00:06:18 Intervjuer
Ja, har du kanskje hatt det ITGK-kurset?

00:06:23 Lærer
Jeg har hatt IT-intro som det het den gang i tiden, og jeg har hatt et program, et fag som heter numerikk og programmering.

00:06:27 Intervjuer
Ja.

00:06:34 Lærer
Så jeg har litt programmeringsbakgrunn der, og så har jeg jobbet litt med arduino i teknologi og forskningslære. Og jeg har og brukt en del Lego Mindstorm, for det er jo en lav inngangsterskel for elevene.

00:06:50 Intervjuer
Ja.
Fint. Da tror jeg vi kan gå over til del 2 som handler om utfordringer med å lære programmering. Og nå har jo du lært programmering for en god stund siden sånn for første gang, men hva tror du er den vanskeligste programmeringskonseptene å lære?

00:07:10 Lærer
Det som jeg opplever som... Både som jeg opplevde selv, fordi at vi hopper rett inn på et høyt nivå, og som jeg ser med elever, er det å få med seg alle detaljer i en programmeringsøvelse. Uansett hva det er for noe egentlig. At du må være så detaljert. Og så en del av arbeidet med løkker ser jeg at mange har problemer med, selv om jeg henger noenlunde greit med. Men jeg ser det et problem.

00:07:48 Intervjuer
Ja. Da tenkte jeg at jeg skulle vise den første ordskyen. Ordskyene er laget basert på de refleksjonsnotatene som deltagerne på videreutdanningskurset grunnleggende programmering har levert inn. Og da er ordskyen laget sånn at jo oftere et programmeringskonsept har blitt nevnt som utfordrende, jo større er order og jo oftere forekommer det da. Skal vi se om jeg får opp den her da. Det er jo alltid like spennende.
Sånn. Har du noen tanker om denne her ordskyen? Og da er det snakk om utfordringer for lærere da.

00:08:32 Lærer
Ja. Kode og pseudokode, det pleier å være en utfordring, synes jeg, så det er jo egentlig kjent. Løkker, ser jeg at det står der og if statements. Det er litt av det samme problemet. Så det er mye kjent da. Det å definere variabler vil ikke jeg selv satt så høyt, men kan tenker meg at mange gjør det. Men ellers så ja, mye kjent.

00:09:11 Intervjuer

Ja. Tror du at de her utfordringene kan jo ha noe sammenheng med hvordan kurset er tilrettelagt? I forhold til aktiviteter og varighet og format, og...

00:09:24 Lærer
Det kan nå... Jeg tror det kan ha noe med bakgrunnen til mange av de som tar kurset å gjøre. Så ikke bare hvordan kurset er lagt opp, men det at du får et så stort spenn av faglig bakgrunn og kompetanse.

00:09:42 Intervjuer
Ja, så du har ikke noe tanker om hvordan kurset kunne vært tilrettelagt på en annen måte for å minske utfordringene?

00:09:53 Lærer
Jeg vil jo si at det traff egentlig greit for min del, men jeg har jo faglig bakgrunn innen noe programmering. Så om man skulle hatt en slags innledende øvelse der man ikke måtte kode. Sånn som man gjerne gjør, eller i hvert fall jeg gjør med elever i videregående skole, så har jeg en del innledende øvelser der det å skrive kode ikke er poenget. For å få med seg alle.

00:10:25 Intervjuer
Ja. For å få litt sånn introduksjon før man går over på det tekniske?

00:10:30 Lærer
Ja.

00:10:31 Intervjuer
Ja. Fint. Da er vi over på del 3, som går på utfordringer med å undervise i programmering. Og hvilke deler av programmering tror du er mest utfordrende for elevene dine å lære?

00:10:47 Lærer
Nå har jeg ikke hatt en klasse selv i år, men jeg har jo vært inne som vikar i noen. Og den mest utfordrende timen jeg har hatt, det var når jeg fikk tildelt oppgaven å lære de hva en int, hva en float, og så videre er for noe. Det ble jo i det hele tatt en litt rar øvelse å skulle bruke en time på å prøve å forklare de konseptene. Så jeg tror en del av utfordringene er at mange prøver å gjøre dette her med mye mer tavleundervisning og tradisjonell undervisning. Mens jeg på min side har jo innfallsvinkel fra teknologisiden, så det at jeg vil jo ofte ta ferdige kodesnutter og la de få lov å leke med det og finne ut hva det gjør. Så jeg har en annen innfallsvinkel enn det er mange har. Men tradisjonell tavleundervisning i undervisningssituasjonen med 16 åringer, det fungerer ikke så veldig bra med programmeringskonsepter, synes jeg.

00:11:51 Intervjuer
Ja. Er det noen spesifikke konsepter som du tror er mer utfordrende enn andre da?

00:12:00 Lærer
Hele tankegangen med at de skal lage en kode. De skal sette opp, eller noe skal gjøre. De må være detaljerte. De må tenke igjennom. Jeg har prøvd å sammenligne det med figurtall, fordi med figurtall, der må du å legge en plan og tenke litt igjennom. Og det syns de var mye lettere fordi at da kunne de tegne. Men du kan jo forsåvidt gjøre det og med programmeringen, tenker jeg. At du kan jo tegne det opp, men det er så unaturlig for mange å prøve og lage en struktur på den måten.

00:12:35 Intervjuer
Ja. Så å planlegge strukturen litt med tegning før man går over på faktisk skrive programmene da?

00:12:42 Lærer

Ja. Ha en plan før man starter.

00:12:46 Intervjuer
Ja. Da tenkte jeg skulle vise deg den neste ordskyen. Den er jo da laget på samme måte som den forrige, bare nå basert på utfordringer som elevene kan ha med å lære programmering. Eller hva som er utfordrende å lære for elevene er vel mer riktig. Den ser sånn ut. Har du noen tanker om den?

00:13:12 Lærer
Ja, det er de løkkene. De pleier jo å være et problem da. Men jeg er vel på mer på dette med nye tankeganger og bare det å sette det opp, det pleier å være problemet. Feilmeldinger, det pleier å bety at noen rekker hånda i været og sier jeg skjønner ingenting. Så der hadde det jo vært en fordel om de ofte leste, prøvde og utforsket. Men det er noe med struktur og. Bare det... Jeg husker selv at hvis du mangler en parentes i python, hvis du mangler en innrykk, så får du jo feilmelding. Eller mangler et kolon. Og det er frustrerende for mange. Det å måtte være så tydelig i språket sitt.

00:14:03 Intervjuer
Ja, så syntaksen er litt ekstra utfordrende for elevene?

00:14:09 Lærer
Ikke nødvendigvis syntaks i den forstand, men det å være presis i et språk, uansett om det er å skrive en vanlig tekst eller skrive et matematisk stykke. Jeg driver og innprenter at de må huske likhetstegn når de skriver matematikk forhånd. Det er like mye det. Bare språkbiten, ikke bare syntaksen.

00:14:31 Intervjuer
Ja, så å huske på alle detaljene da?

00:14:34 Lærer
Ja.

00:14:35 Intervjuer
Ja. Datamaskinen er jo litt strengere på det enn menneskene også.

00:14:39 Lærer
Ja den er det.

00:14:42 Intervjuer
Hvis vi sammenligner den her ordskyen med den forrige ordskyen, så ser vi jo at utfordringene for elevene er veldig sånn konsentrert om noen få områder. Mens utfordringene for lærerne var mer sånn jevnt spredt utover de fleste programmeringskonseptene. Har du noen tanker om hvorfor det er sånn?

00:15:02 Lærer
Det vil vi ha noe med de enkeltes bakgrunn å gjøre, vil jeg tro. At som voksen så har man en annen erfaringsbakgrunn enn det 16 til 19 åringer, eller yngre for den saks skyld, har.

00:15:22 Intervjuer
Ja. Så den helhetsforståelsen er kanskje forskjellen?

00:15:26 Lærer
Ja. Og mange har jo ingen programmeringsbakgrunn i voksen alder, mens andre har mye. Så mer variert der, mens elever, de har gjerne noen ting de er veldig frustrert over. Så løkker og feilmeldinger, det kan jeg godt tro at det er stor frustrasjon.

00:15:46 Intervjuer
Ja. Har du noen tanker om hvordan du tenker å adressere de her utfordringene når elevene skal lære programmering i skolen da?

00:16:00 Lærer
Jeg har lyst til å prøve den tilnærmingen som jeg har gjort litt tidligere. Som altså da ikke en sånn klassisk tavleundervisning. For jeg tenker jo at for mine elever, så er dette et verktøy. De skal ikke kunne programmere all verdens, og de skal ikke være en programmerer som sitter og jobber med det. De skal kunne bruke det som et verktøy og få det til å virke. Og da er det viktigere enn at de kan alle konsepter, og de kan sette det opp og så videre. Men bare det at de kan tenke gjennom og få ut – hva er det jeg skal ha dette til å gjøre? Og så forhåpentligvis får det til å gjøre det de ønsker, men ikke nødvendigvis med det at de husker hvordan alle konseptene var. Her er det lov å ha hjelpemidler, tenker jeg. Sitter du først ved PC-en, så har du gjerne hjelpemidler.

00:17:00 Intervjuer
Ja, det er litt sånn at når man lærer å programme, så handler like mye som var om å lære seg å google som å skrive kode.

00:17:07 Lærer
Ja, det er derfor jeg liker å sammenligne med det... Jeg husker jeg hadde eksamen i numerikk og programmering, der vi skulle lage et eller annet JavaScript. Et svært oppslagsverk fikk du lov å ha med seg hjelpemiddel, og ellers var det papir og blyant. Og vi er ikke helt der lenger.

00:17:29 Intervjuer
Absolutt ikke. Det er kanskje litt mer tilgjengelig nå.

00:17:33 Lærer
Ja, det er det det er. Når du først har PC-en nå, og sitter og programmerer, så gjør du ikke det for den øvelsen å sitte å gjøre det med papir og blyant. Du gjør det for å få det til å virke. Og da er ikke poenget nødvendigvis at jeg skal pugge alle konsepter og alle biblioteker alt, da kan jeg slå det opp.

00:17:53 Intervjuer
Ja. Har du noen tanker om hvordan du vil gå fram for å motivere elevene dine til å lære programmering?

00:18:02 Lærer
Jeg har en tendens til å ha litt annen innfallsvinkel enn mange av de andre matematikklærerne på skolen. Med at jeg ikke nødvendigvis bedriver denne tavleundervisningen. Jeg gir de ferdige snutter og ber de finne ut hva dette her gjør istedenfor. Eller lar de få lov å undersøke, og bruker heller litt tid på den lek og lær-biten. Det har vi gjort mye med Lego Mindstorm, at poenget er ikke at jeg skal vise de hva alle disse ikonene gjør. Jeg skal heller gi de noe som fungerer, så får de forteller meg – hva gjør dette her da? Så istedenfor å la de kode, så gir jeg dem en ferdig kode, også spør jeg «hva gjør den?» Jeg tenker at det er viktigere at de leser den og forstå hva den gjør, enn at de husker alt om hvordan de skal sette det opp.

00:18:54 Intervjuer
Ja. Da har jeg den siste ordskyen. For de to siste ordskyene gikk veldig på sånn spesifikke programmeringskonsept eller tekniske deler av å lære programmering, mens den siste ordskyen som jeg skal vise nå, den handler mer om utfordringer knyttet til undervisningen av programmering. Og den ser sånn ut. Er dette her noe du kjenner deg igjen i?

00:19:27 Høyttaler 1
Ja. Hvis vi tenker om matematikk, så både motivasjon og interesseområder er jo kjent. Så noen er veldig motivert. Noen har ikke peiling og absolutt ingen motivasjon for å lære seg det. At det i tillegg kommer i matematikk, som de aldri har skjønt noe av, det gjør det jo bare enda verre for mange. Så du merker veldig stort skille, føler jeg, når jeg har vært inne i klasserom som vikar i praktisk matte og i teoretisk matte. At noen elever er veldig på. Dette hadde gjort før, de er kjempeinteresserte i programmering. De sitter og programmerer spill på egen hånd. Og så har du da den andre motsetningen, som ikke skjønner det at det er et bilde av en trinket de ser på, så de kan altså ikke skrive koden sin rett inn. De må gjøre noe mer. Så det store spennet der er jo det du ser mye av i de ordene som popper opp her.

00:20:39 Intervjuer
Ja. Så stort spenn i alt fra interesse til motivasjon til forkunnskaper da?

00:20:47 Lærer
Ja.

00:20:48 Intervjuer
Ja. Er det noen ting knyttet til undervisning og programmering som du ikke synes står her da? Som du savner?

00:20:59 Lærer
Det er helt sikkert, men står mye rundt her som går inn på programmering.

00:21:06 Intervjuer
Ja. Da kan vi jo gå videre. Det å lære programmering krever jo mye tid og mengdetrening. Og hvilke utfordringer ser du knyttet til den tida elevene har tilgjengelig for å lære programmering?

00:21:27 Lærer
Det har vi ikke nok tid til, så enkelt er det. Hvis de skal lære det ordentlig, så har vi ikke nok tid. Da skulle vi hatt et eget fag.

00:21:36 Intervjuer
Ja. Da er tanken at det er for dårlig tid i matematikk og naturfag og der det skal brukes da, til...?

00:21:46 Lærer
Ja, matematikk, naturfag, fysikk, alle fag der det skal brukes. Og i hvert fall nå de første to årene etter fagfornyelsen, så har vi altfor dårlig tid. For å nå legges det jo opp til at de skal kunne dette er fra grunnskolen. Og det kan det jo ikke, så du må jo løfte de fra helt blank og opp til et visst nivå, og det har du ikke tid til.

00:22:10 Intervjuer
Nei. Det blir en sånn overgangsfase her der man må gape over mer enn det man kanskje trenger om 5-10 år?

00:22:18 Lærer
Ja. Absolutt. Fordelen med det er jo at det gir jo de lærerne som er helt blanke litt tid til å løfte seg selv og, men de har jo heller ikke tid til å drive med alt for mye selvstudier.

00:22:34 Intervjuer

Så tida er en utfordring både for elever og lærere egentlig da?

00:22:38 Lærer
Ja.

00:22:39 Intervjuer
Ja. Supert, da er vi kommet til del 4, som handler om lærerens faglige utvikling. Og da tenker vi på den utviklingen av programmeringskompetansen og undervisningskompetansen. Og først så har jeg et spørsmål om videreutdanningskursene på NTNU. For der er jo programmeringsfagene delt i to fag, der man har et høstfag der man fokuserer på å lære programmering, før man har et vårfag der fokuset er på å undervise programmering. Hva syns du om den oppdelingen?

00:23:18 Lærer
Jeg syns for så vidt at det var en grei oppdeling. Begynte greit på høsten, og så ble det litt vanskeligere etter hvert. Det jeg ser på den undervisningsbiten på våren var at det var jo veldig mange programmeringskonsepter som jeg var innom. Så hvis man kunne hatt et fokus, så kunne man kanskje gått mer i dybden på det.

00:23:44 Intervjuer
Ja, for det vårkurset er jo lagt opp til å være et sånt fleksibelt, veldig valgfritt kurs. Og hva synes du om å lære programmering et kurs der du må på en måte velge kursen selv?

00:23:58 Lærer
Det er mye det jeg har gjort tidligere, så for meg er det for så vidt greit.

00:24:06 Intervjuer
Ja. Hva synes du om at kursene gir 7,5 studiepoeng i forhold til aktiviteter, varighet, format og?

00:24:17 Lærer
Jeg tenker at det er en passende studiebelastning.

00:24:21 Intervjuer
Ja. Hadde du kunne tenkt deg kortere kurs som har gitt færre studiepoeng, eller lengre kurs som har gitt flere da?

00:24:32 Lærer
For egen del, så kunne det godt vært mer periodisert. At man hadde kanskje til og med delt i 4, men det er absolutt ikke nødvendig.

00:24:43 Intervjuer
Nei. Så du synes arbeidsbelastningen har vært grei i forhold til de studiepoengene det har gitt?

00:24:49 Lærer
Ja, men jeg ser jo de... Hvis man da tenker at man var helt blank i utgangspunktet, så har du vel arbeidsbelastningen, i hvert fall på høsten, vært stor.

00:24:59 Intervjuer
Ja. Tror du at det hadde vært nødvendig eller nyttig med flere kurs som bygger videre på de her to kursene da?

00:25:08 Lærer

Jeg kunne forsåvidt godt tenkt meg sånn fordypningskurs, men det spørs jo hva man får lov til å arbeidsgiver. De ønsker jo seg bare å løfte nå med minimum på programmeringskompetanse, i kompetanse for kvalitet. Så der ser jeg arbeidsgiver gir 15 studiepoeng, for da kan du programmere. Det er riktignok hakket bedre når jeg hadde en diskusjon med noen rektorer som sa det at nå har vi gitt dere en hel dag til å lære dere programmering, så nå er dere forberedt.

00:25:31 Intervjuer
Jøss. Har du noen tanker om hva du kunne ønske deg fordype deg mer i?

00:25:53 Lærer
Nei, nå har jeg jo fått meg et nytt programmeringsspråk, så for min del så ville jeg jo sett med på python i matematikk.

00:26:02 Intervjuer
Ja. Fått litt mer innføring i hvordan man kan bruke python i matematikken?

00:26:07 Lærer
Kanskje sånn at jeg hadde hatt mer kompetanse enn det jeg føler at elevene skal sitte igjen med når de går ut videregående, for jeg liker jo å kunne litt mer enn det de skal kunne.

00:26:22 Intervjuer
Ja. Videreutdanningskurs er jo bare en av mange måter å lære programmering på. Men hva tror du er den beste måten å lære programmering på for lærere som er i full jobb?

00:26:33 Lærer
Jeg vil tro det avhenger av hvert enkelt individ på hva som er det beste. Men jeg ser det er mange som kjører kursing av lærere over perioder. Andre sender de på mer fordypende kurs, så det at det er jo litt variert. Men det viktigste er vel at alle som underviser faget der det kommer programmeringen nå faktisk får en videreutdanning av noe slag. Det sitter nemlig mange rundt omkring som ja... Har nektet å bruke GeoGebra i det hele tatt i matematikk. Og det begynner jo å bli noen år siden vi skulle over på digitale hjelpemidler, og jeg kan ikke se for meg at de også skal lære seg programmering og bruke det.

00:27:22 Intervjuer
Ja. Så handler litt om viljen til å lære selv og da?

00:27:29 Lærer
Ja, det er... Hvis du må ta initiativet selv, så er det veldig mange som ikke vil gjøre det.

00:27:33 Intervjuer
Ja. Har du noen gang vært i en situasjon der du måtte ha lært noe programmeringskonsepter eller noen elementer innen programmering sammen med elevene dine?

00:27:45 Lærer
Ja, flere ganger. Men det har også noe med at jeg underviser teknologi og forskningslære, så plutselig så sitter du der med en gruppe som har funnet ut... Kan ta et eksempel fra noen år tilbake. Det var noen som hadde funnet ut at det gikk an å programmere Lego Mindstorm i C. Det var jo spennende, for det hadde ikke jeg vært borti. Så da må man jo bare snu seg rundt og prøve å finne ut av dette sammen med elevene da.

00:28:15 Intervjuer
Ja. Hvordan du det påvirker din rolle som lærer, da når man må på en måte lære litt sammen?

00:28:21 Lærer
Det er noe som jeg føler er noe av konseptet i teknologi og forskningslære mye. Altså du kan noe, men du kan ikke alt, og det er også en av grunnene til at mange lærer ikke vil underviser det faget. Nettopp på det at du kan jo bli kastet på dypt vann på absolutt alle mulige rare varianter. Men jeg syns det er gøy. Elevene syns jo det er gøy når jeg sier at dette kan jeg ingenting om, så da får vi prøve å finne ut av. Men det gjør jo at det er begrenset med lærere som vil ha faget.

00:28:55 Intervjuer
Ja. Da har jeg et siste spørsmål. Og det går på det at å lære programmering tar jo lang tid, tid langt utover de videreutdanningskursene på NTNU. Og det finnes jo mange måter å støtte den lange læringsprosessen, gjennom for eksempel korte kurs, lengre kurs, seminarer, fagforum, deling av læringsmateriell. Listen er lang over måter å støtte en læringsprosess. Men hvordan tror du at den lange læringsprosessen kan støttes på best mulig måte?

00:29:37 Lærer
Det er nok kanskje et vanskelig spørsmål, men det må i hvert fall settes av tid til det. Det kan ikke baseres på at alle skal sitte hjemme og gjøre dette for seg selv. Det er også noe med å samhandle, dele opplegg, diskutere med andre, eventuelt videreutvikle opplegg av hva man har. Som man ikke kan gjøre med å sitte hjemme for seg selv.

00:30:05 Intervjuer
Ja, så det viktigste er sånn en kompetansedeling og materialedeling?

00.30.13 Lærer
Ja, og en samhandling. Ikke bare det at du deler med noen, men at du snakker med noen. En kan ha en idé, og en annen kan tenke et mye bedre konsept for den ideen, så det å kunne snakke sammen og ikke bare sitte hjemme for seg selv. Det tror jeg er viktig i undervisningsbiten med programmering.

00:30:34 Intervjuer
Ja. Fint, da har jeg egentlig fått svar på alle spørsmålene mine, men har du noen kommentarer eller refleksjoner eller tilbakemeldinger som du ikke føler at du har fått sagt?

00:30:50 Lærer
Nei, jeg har vel vært innom det meste nå, har jeg ikke det?

00:30:55 Intervjuer
Jo, du har i hvert fall svart på det jeg lurte på.

00:30:58 Lærer
Ja.

00:31:00 Intervjuer
Da er det egentlig bare for meg å si tusen takk for at du tok deg tiden til å stille opp på intervju. Det er jo veldig verdifullt å få høre rett fra kilden hvilke utfordringer som man ser knyttet til både å lære og undervise programmering. Så det setter vi veldig pris på da.

00:31:19 Lærer

Nå er vel ikke jeg den gjennomsnittlige matematikklærer som tar programmeringskurs, men ellers så er det jo hyggelig å få uttale seg.

00:31:29 Intervjuer
Ja, det er viktig med ulike perspektiv også på... At man får litt de forskjellige vinklingene både på videreutdanningskursene og utfordringer knyttet til å ta det her videre inn i skolen. Så litt utenfor gjennomsnittet kanskje, men veldig nyttig.
Men da får du bare ha en fin dag videre og tusen takk for at du stilte opp.

00:31:54 Lærer
Bare hyggelig. Ha det bra.

00:31:54 Intervjuer
Ha det.

Transkripsjon 2

00:02:57 Lærer
Hallo.

00:02:59 Intervjuer
Hallo hei, der var du. Hører du meg godt?

00:03:03 Lærer
Ja, hører du meg?

00:03:03 Intervjuer
Ja.
Velkommen til intervju.

00:03:08 Lærer
Ja, takk.

00:03:10 Intervjuer
Har det allerede vært en lang dag, eller er det?

00:03:16 Lærer
Nei, altså, det har vært en start med møte, så jeg akkurat ferdig på første møte, men. Ja.

00:03:24 Intervjuer
Ja. Da er du klar for å bli stilt litt spørsmål.

00:03:26 Lærer
Ja.

00:03:27 Intervjuer
Som du sikkert ser opp i hjørnet her, så blir intervjuet her tatt opp. Men
det opptaket blir slettet så fort intervjuet transkribert da. Og så ja, blir alt anonymisert, så at det er ingenting som kan knyttes tilbake til deg. Ønsker du at jeg skal sende deg en kopi av
det transkripsjonen når det er ferdig?

00:03:56 Lærer
Ikke så nøye det. Jeg tenker det at så lenge det dere kan finne noe fornuftig i det, så er det greit for meg.

00:04:01 Intervjuer
Ja. Så er det sånn at hvis du på et eller annet tidspunkt finner ut at du ønsker å trekke deg fra forskningsprosjektet igjen, så er det bare å gi meg en beskjed, og så sletter vi alt av transkripsjoner. Sånn som kommer fra det intervjuet her.

00:04:02 Lærer
Mm.

00:04:19 Intervjuer
Så som jeg skrev i mail, så jeg intervjuet sånn delt i 4 deler. Og i de 2 midterste delene så kommer jeg til å vise deg noen ord skyer. Så er du forberedt på det.
Har du noen spørsmål før vi begynner?
00:04:35 Lærer
Nei, ikke egentlig.

00:04:37 Intervjuer
Nei. Da tenker jeg vi bare sette i gang. Da er første delen litt sånn kort om din bakgrunn.
Hvilke fag underviser du i?

00:04:49 Lærer
Matte og naturfag.

00:04:50 Intervjuer
Matte og naturfag. Og tror du at du kommer til å undervise programmering i begge de her fagene?

00:04:57 Lærer
Ja, altså, det ligger jo i fagplanen, så har ikke så mye valg om det.
Ja, så det er jo det egentlig så var min en sånn motivasjon for å ta dette fag og finne ut hvordan jeg kan gjøre det.

00:05:08 Intervjuer
Ja. Hvilke trinn underviser du på?

00:05:13 Lærer
Videregående første klasse.

00:05:16 Intervjuer
Ja. Og hvilke erfaringer med programmering hadde du før deltakelsen på NTNU kursene?

00:05:25 Lærer
Svært lite. Jeg hadde skjønt det var noe som het input og output. Vi var der liksom. Så det... Det var sånn at jeg... Det var ikke vanskelig å begynne å være med på studiet, men jeg kunne veldig lite.

00:05:39 Intervjuer
Ja. Så da har du kanskje ikke undervist i programmering før heller?

00:05:43 Lærer
Nei.

00:05:46 Intervjuer
Nei. Ja, da var det unnagjort. Del 2 handler jo da om utfordringer med å lære programmering for deg. Under NTNU-kursene. Så da lurer jeg på, hva tror du at det er de vanskeligste programmeringskonseptene å lære?

00:06:05 Lærer
Tenker du for meg personlig, eller?

00:06:07 Intervjuer
Ja. Ja, eller for lærere. Hva er det som er utfordrende å lære?

00:06:12 Lærer
Altså her... Sånn som for meg, så var det vel... Dette python-kurset gikk veldig greit. Akkurat det med definisjoner som kommer inn på slutten. Det slet jeg veldig med å forstå, og jeg forsto det vel egentlig ikke så mye da, men det har jo begynt å demre litt etter hvert. Dette kurset nå i vår er jo veldig sånn... Delt, og du velger forskjellige ting du skal ta. Og da merker jeg at jeg synes veldig mye av året veldig vanskelig, men jeg har jo lagt meg der som jeg tenker at jeg kommer til å bruke det, og jeg har lagt meg veldig på modellering, fordi at det vet jeg at jeg kommer til å bruke i matte og naturfag. Og egentlig veldig mye python, litt micro:bit, men jeg ser jo at når den undervisningen som har vært i python, med, altså... Det med å importere bibliotek, da faller jeg litt av rett og slett det. Litt fordi at det kanskje føles litt lite relevant for meg akkurat nå. For det at jeg ser at elevene kan jo veldig lite, så vi må jo begynne helt i fra basic med elevene. Så jeg tenker jo litt at... Ja, nå er vi på en plass der elevene kan lite, og vi kan heller ikke så mye, men vi begynner der, og så må vi jo tilegne oss noe mer kunnskap etterhvert. Men den kunnskapen vil nok være lettere å tilegne seg når vi har jobbet med det en stund, ser jeg for meg.

00:07:37 Intervjuer
Ja. Det blir litt enklere å vite hva man skal fokusere på?

00:07:40 Lærer
Og så tenker jeg... Jeg personlig har ikke noen veldig stor personlig interesse for programmering, sånn at det hjelper nå kanskje ikke heller når ting begynner å bli innviklet. Ja.

00:07:54 Intervjuer
Ja. Da tenkte jeg at jeg skulle vise deg en ordsky. Det er en ordsky som er laget basert på refleksjonsnotatene fra det høst-faget grunnleggende programmering, der flere deltakere på kurset da har reflektert litt over hvilke konsepter som er vanskelig å lære da. Skal vi se om jeg får den opp her...
Der. Ser du den nå?

00:08:24 Lærer
Ja.

00:08:25 Intervjuer
Ja, har du noen tanker om den her? Noe som burde vært større, eller mindre, eller noe du savner?

00:08:32 Lærer
Men hva er på en måte spørsmålet her? På en måte, er dette hva som var vanskelig, er det det? Eller er det på en måte?

00:08:37 Intervjuer
Ja. Det er konsepter som lærere skrev om i refleksjonsnotatene sine, som de fant utfordrende. Så om du har noen tanker om du er enig eller uenig i om ting har vært utfordrende eller ja.

00:08:55 Lærer
Den største med kode og pseudokode. Jeg syns jo kanskje det må skrive pseudokoder er nok litt vanskelig for det at, altså, hele greia med koding handler jo om problemløsning. Og da må du løse et problem for å skrive den pseudokoden. Så det er jo på en måte en sånn... Du vil jo helst gå rett på koden, men jeg ser jo etter hvert at det er en fornuftig måte å lære elevene å tenke på. Så jeg kan være enig i det med pseudo koden. Ja. Ny måte å tenke på? Ja, kanskje, men jeg vet ikke... Bibliotek-greiene er jeg enig i at var litt utfordrende. Ja. Variablene. Jeg synes... Nei, ikke variablene. Jeg synes jo som sagt det med funksjoner, det tok meg veldig lang tid før jeg grep om det. Det er ikke sånn at jeg tenker at det jeg ser her er på en måte veldig vrient, men det er kanskje dem pseudokode og med funksjoner og bibliotek som var mest utfordrende for min del.

00:10:10 Intervjuer
Ja. Tror du at noen av de her utfordringene kan ha noen sammenheng med hvordan kurset er tilrettelagt? Du var jo litt inne på det med at det var veldig stor valgfrihet i det vår-kurset.

00:10:27 Lærer
Ja. Ja, hvis jeg kunne komme med tilbakemelding til kurset, så har jeg jo litt jeg vil si. Jeg synes høstkurset. For min del var det kanskje det letteste, men samtidig så var det det minst... Ja, jeg savnet det å ha faktiske forelesninger. Det var en tilbakemelding om at det blir ikke forelesninger her, fordi folk er så på forskjellige nivå. Men det er jo på en måte der vi jobber i skolen. Det er ingen er på samme nivå. Så jeg synes jo at det å si det til en gjeng med lærere blir litt feil. Og det var jo undervisningsfilmer, og de var bra, men jeg kunne nok tenkt meg at litt sånn... Gjerne sånn som vi jobber skolen. Nå går vi gjennom et nytt tema, og så de som synes det er lett, de får bare hoppe over det, og de som synes det er vanskelig, de følger med.
Og eksamen. Jeg tror vi satt som et spørsmålstegn hele gjengen. For det første, det at vi skulle være ei gruppe der vi måtte skaffe oss ei gruppe blant navn på Slack var litt spesielt. Det gikk jo helt fint, men det at vi visste egentlig ikke... Altså, det var så veldig svevende hva skulle gjøre. Så vi satt jo egentlig hele tiden med «gjør vi det vi skal nå». Og det synes jeg i vår. Han hadde jo en hel times forelesning med «hva dere skal gjøre». Så jeg føler at i vår vet jeg hva jeg jobber med. I høst viste vi egentlig ikke hva vi jobbet med. Det var litt den følelsen der.

00:11:51 Intervjuer
Ja.

00:11:54 Lærer
Ja, så jeg føler at i dette vår-semesteret har vi på en måte kanskje blitt fulgt opp litt mer.

00:12:01 Intervjuer
Ja. Det skal jeg ta med, og gi beskjed om til Majid. Men da var opplevelsen bedre på våren enn på høsten da? Sånn at vi kan ta litt lærdom.

00:12:13 Lærer
Ja, han kan veldig godt få en positiv tilbakemelding. For han å være flink å følge opp, og det er på en måte var litt sånn lavterskel, føler jeg. For å få svar på spørsmål.

00:12:24 Intervjuer
Ja, men det er bra.
Da har vi vært litt inne på utfordringene med å lære programmering og utfordringene i kurset, så da kan vi jo gå videre til utfordringen med å undervise programmering. Og hvilke deler av programmeringen tror du blir mest utfordrende for elevene dine og lær?

00:12:46 Lærer

Ja. Altså, vi sitter jo nå i en fagfornyelse, der det står i den nye læreplanen at vi skal programmere. Men ingen vet egentlig hva vi skal programmere. Så det er jo den største utfordringen, men det ligger jo hos lærere, vi aner jo ikke noen ting. Det kommer noen løse eksamensforslag i fra direktoratet, og det virker ikke som de heller egentlig vet. Så største utfordring er – hvor mye skal vi lære dem? Og hva skal vi egentlig lære dem? Vi vet ikke.
Så vi har jo begynt helt basic på python, men jeg merker at jeg har fokusert mest på if-betingelser og løkker. Helt enkelt. Vi har vel vært innom så vidt å tegne grafer, men det har på en måte ikke... Tendensen i eksamensoppgavene har ikke vært der, så jeg ikke fokusert så mye på det. Utfordringen i naturfag har nok vært det at jeg ser vi skal programmere, men jeg skjønner ikke helt hvordan vi skal få til noe relevant programmering, fordi at jeg ikke kunnskap nok og elevene, ja... Så jeg har fulgt noe sånn ferdig opplegg i fra forlaget. Der elevene spør «hvorfor vi skal med dette her?» Og jeg sier at det at vi fordi vi må, rett og slett. Fordi at jeg har ikke noe bedre opplegg. Litt for å krysse av på lista, og det føler jeg blir litt feil. Så det er jo det jeg jobber med eksamensprosjektet, å få et eller annet som er litt mer relevant. Men uansett, problemet, største utfordringen er at ting henger og svever, vi vet ikke hvor vi skal undervise i. Så det er den største utfordringen for oss lærere. Og klart det ikke er utfordringer for elevene fordi de... Jeg synes de henger med ganske greit. De på en måte skjønner det ganske greit, men jeg har en veldig liten klasse da, så det har ikke vært noe problem. Ja.

00:14:41 Intervjuer
Ja. Skjønner. Da har jeg jo en ordsky til, som går litt på samme tema som vi vil snakket om nå. Så kan jeg ta opp den, og så kan du få se om du har noen tanker om den.

00:14:51 Lærer
Ja.

00:15:00 Intervjuer
Her er jo en ordsky som representerer litt sånn utfordringene med det å bruke programmeringen i undervisningen da. Er dette her noe du kjenner deg igjen i, eller er det noe som burde ha vært større eller mindre?

00:15:14 Lærer
Denne relation to other subjects, det er jo den jeg slet litt med. Spesielt i naturfag, for å finne noe som faktisk er relevant. Så den er jeg jo enig i at skal være stor. Ja, motivasjonen ser jeg, men motivasjonen ligger nok i at vi kanskje ikke helt forteller de... Altså, vi kan fortelle de hvorfor det er lurt å kunne noe om programmering, men ja, som sagt, ting henger og svever så veldig at det er veldig vanskelig.

00:15:46 Intervjuer
Ja.

00:15:47 Lærer
Der ser jeg en creating good teaching plans. Det er jo vanskelig for oss å vite hvordan vi skal legge opp dette opplegget. Å lage gode, fordi at jeg vil jo tro at i Norge så er det veldig få... Det er en veldig liten andel lærere som føler seg trygge på dette her.
Ja, nei, jeg vet ikke helt hva mer jeg skal si her.

00:16:20 Intervjuer
Nei, nei. Vi kan gå videre også. For det her er jo litt sånn generelle utfordringer med å ta i bruk programmering i undervisningen. Men hvis vi tenker på litt mer sånn spesifikke deler av programmering som elevene kan synes er vanskelig, altså spesifikke konsepter. Har du noen tanker om hvilke elementer, som f.eks. løkker eller funksjoner, eller...?

00:16:51 Lærer
Jeg tror det å lære seg en syntaks er litt utfordrende. Jeg husker at det var en av kommentarene elevene kom med i høst. Det at, å ja, dette er jo et språk. Dette er et nytt språk. Og så er det engelske ord. Så det er på en måte litt massivt, og så er vi ennå der at elevene er redde for GeoGebra. Det har jo vært i læreplanene i en del år allerede. Så blir det nå enda en ting vi skal inn med. Ja. Men jeg tror nok kanskje det med å lære seg syntaksen, for jeg ser jo at når de da har lært seg den, så er det lettere å sette opp resten. Men det er jo det å klare og bryte opp dette problemet og kode det. Og vite hvilke ting de skal bruke. For-løkke, while-løkke. Men klart jeg skjønner jo utfordringen, for jeg synes jo det er vanskelig selv enda.

00:17:49 Intervjuer
Ja.

00:17:49 Lærer
Så jeg tenker vel språk først og fremst. Og så det med... Det ligger nok kanskje det å kunne lage gode pseudokoder eller algoritmer på problemet, men ja...

00:18:02 Intervjuer
Ja. Da har jeg en siste ordsky som handler om akkurat det vi snakker om nå. Så kan vi jo se om du er enig i den, eller om det burde ha vært gjort noen endringer i den. Den ser sånn ut da. Med utfordrende programmeringskonsepter for elever.

00:18:25 Lærer
Ja okei. Ja, jeg vet ikke om løkken i seg selv er problemet, men det som er problemet her synes jeg er at det er kanskje vanskelig å vite akkurat hvilken løkke de skal bruke.
Jeg tenker nok dette her med pseudokode og problemløsning, først og fremst.
Og det med feil... Det på en måte er de vant med å få i GeoGebra hele tiden. Det er standard, og etter hvert så lærer de seg det at det er på en måte punktum, ikke komma. Og det er litt sånn kolon her og kolon der.
Men altså. Jeg er ikke et kjempegodt sammenligningsgrunnlag. Jeg har 3 elever i matte dette året, sånn at det er tre veldig oppegående elever. Altså det er ikke... Ja, jeg er jo ikke representativ sånn.

00:19:22 Intervjuer
Ja, så det kan oppstå litt annen utfordringer hvis man får en litt annen elevgruppe?

00:19:28 Lærer
Ja, jeg har jo en kollega her, som på en måte, det største problemet i starten var en elev som spurte «hvor er kolon hen på tastaturet?» Vi var der, og når du da skal begynne å kode... Så ja, da må du begynne veldig basic.

00:19:42 Intervjuer
Ja.
Hvis vi sammenligner den her ordskyen med ordskyen for lærerne, så ser vi jo at elevordskyen er mye mer sånn konsentrert på noen områder, mens lærerskyen er kanskje litt mer sånn jevnt spredt utover de fleste programmeringskonseptene. Har du noen tanker om hvorfor det er sånn?

00:20:04 Lærer
Jeg er overrasket over at loopen er så stor. Jeg på en måte ville ikke tenkt at det var det største problemet. Men det er jo litt interessant, for det viser jo kanskje litt hva veldig mange tenker er problemet.

Hva var det egentlig du spurte om her? Om jeg hadde hatt tanker om hvorfor det var forskjellig?

00:20:33 Intervjuer
Ja, hvorfor det er mer konsentrert for elevene, mens kanskje litt mer jevnt spredt for lærerne?

00:20:40 Lærere
Jeg vet ikke. Elevene lærer kanskje ikke like mange programmeringskonsept. Jeg vil jo tro at... Jeg har fokusert på betingelser og løkker, og det er kanskje andre også som har gjort det da. Ja, og der er løkkene vanskeligere enn betingelsene. Så det har kanskje noe med det å gjøre.

00:20:59 Intervjuer
Ja. Det høres jo fornuftig ut.

00:21:03 Lærer
Og så har vi jo ikke veldig mange... Altså, når du ikke så god på programmering selv, så vil du gi elevene ferdige oppgaver, og det er ikke veldig mange gode oppgaver å bruke. Så de får på en måte ikke den mengdetreningen som de nok burde hatt.

00:21:18 Intervjuer
Ja. I forhold til den mengdetreningen, så vet vi jo at det krever jo enormt mye tid å lære seg programmering. Hvilke utfordringer ser du med den tiden elevene har tilgjengelig for å lære programmering?

00:21:35 Lærer
Der er jeg igjen ikke så representativ, for hadde jeg hatt klasse en på 25, så ville tiden vært mer knapp. Nå har jeg hatt ganske god tid til å følge opp elevene. Jeg føler ikke det har vært noe problem å legge inn programmering i matten. I naturfag har jeg vel også hatt tid, men der har vel hatt mer problemer med hva jeg skal gjøre. Men jeg føler på en måte du kan bake den programmering ganske godt inn. Vi begynte året med å ha et «innføring i programmering»-kurs med en 3-4 skoletimer, som er på en måte... Det er altså de fikk lett innføring i python, det å kunne skrive inn og ja, vite hva programmering er da. Så jeg følte vi fikk en start der da, sånn at de var ikke helt ferske når vi skulle begynne i timene etterpå.

00:22:26 Intervjuer
Ja, da tror jeg egentlig at jeg har fått svar på... Nei, et spørsmål til. Vi har jo snakket litt om at det kan være utfordrende for elevene å lære å skrive syntaks, og pseudokode, og å bruke de ulike programmeringskonseptene. Har du noen tanker om hvordan du vil gå frem for å klare å lære bort det som du... Spesielt det du selv synes er utfordrende?

00:23:02 Lærer
Jeg tenker syntaks... Den er jo... Det er jo et nytt språk det. Men de må på en måte kanskje få se det at det har nytteverdi. Men så sliter jeg litt med det at... Vi skal ha Excel, vi skal ha GeoGebra, vi skal ha med kode i python. Det er veldig mange ting de må forholde seg til, og da er det veldig vanskelig å legge... Bruker veldig mye tid på hver av de delene der. Men jeg tenker jo at det viktige er jo at det er gode oppgaver tilgjengelig, for vi som er såpass ferske som lærer på dette her er ikke er ikke flinke i det. Mange av oss. Så vi trenger på en måte en litt sånn bank å hente i fra. Så vi kan utvikle oss litt mer også, sammen med elevene.

00:23:58 Intervjuer
Ja. Opplever du at det er utfordrende med tilgangen på norsk materiale?

00:24:06 Lærer
Ja, jeg tenker jo at... Vi har jo vi har jo lærebøker, og vi har der vi har tilgang på veldig mye, eller læreverk og sånt. Men det er veldig varierende hva som finnes av programmeringsopplæring. Og det er jo som jeg sa tidligere, at problemet vårt er at vi vet egentlig ikke hvordan vi forventer til eksamen, hva de skal kunne. Så da er det jo vanskelig å vite litt hva vi skal bruke.

00:24:30 Intervjuer
Ja. Det er litt sånn rask overgang nå. Fra ingen programmering til masse.

00:24:33 Lærer
Ja, men vi var forberedt... Ja, det var jo sagt at det skulle komme, men jeg hadde forventet litt mer konkretisering akkurat rundt hva vi skal kunne av programmering, for hvis vi bare skal kunne programmering, så kan det være veldig, veldig varierende hva man ender ut med.

00:24:52 Intervjuer
Ja. Så kan det jo være litt utfordrende å få til at programmeringen blir med i undervisningen som en god del, og at det ikke bare blir en ting på toppen av alt det andre for elevene.

00:25:05 Lærer
Ja. Det er akkurat det.

00:25:09 Intervjuer
Fint. Da har jeg kommet til den siste delen, som går på lærerens faglige utvikling. Som går litt på det der med å utvikle programmeringskompetansen sin over tid da. Vi var litt inne på tidligere at det er høst-kurs som heter grunnleggende programmering. Og et vår-kurs som heter anvendt programmering. Der på høsten, så er det fokus på å lære programmering, mens på våren så er det mer fokus på hvordan å bruke programmering i undervisningen. Hva synes du om denne oppdelingen?

00:25:43 Lærer
Jeg synes den var grei. Men klart når da eksamen på høstsemesteret handler om egentlig bruk av programmering. Så har du på en måte ikke hatt det fokuset før. Og det blir litt sånn, vel, overveldende hva du skal lage i den eksamen der.

00:26:07 Intervjuer
Ja. Så det ble litt sånn blanding?

00:26:10 Lærer
Ja. Og så er jeg altså... Jeg synes jo det var veldig fint at vi har anvendt programmering, at det vi egentlig jobber mot er et produkt som vi selv kan bruke. Så vil jo det være veldig forskjellig i fra... Ja, mellom oss lærere. Hva vi trenger og interessene våre. Og der har jeg jo uttalt tidligere til foreleser at jeg er litt kritisk til der timekravet på eksamen. For å sette ett, på en måte, bastant timekrav, at det skal du bruke... For det første, så er det ingen lærere vant med å ha 40 timer tilgjengelig for å lage 5 undervisningstimer. Det er jo helt absurd. Og for det andre som jobber vi så forskjellig. Vi har helt forskjellige opplegg. Noen vil kanskje kreve mindre tid, andre vil kreve mer tid. Så jeg, på en måte tenker, at å sette en sånn grense der du egentlig blir så veldig opphengt i timetallet, mer enn det du skal gjøre, synes jeg blir litt feil. Men sånn ellers så synes jeg jo eksamen var bra, eller sånn, at det er på en måte... Du skal få et produkt som du selv skal bruke, så er jo det veldig motiverende.

00:27:18 Intervjuer
Ja. Det blir litt nytteverdi i aktivitetene.

00:27:21 Lærer
Ja.

00:27:26 Intervjuer
Vi var litt inne på i sted at anvendt programmering er et sånn veldig fleksibelt kurs, der man kan se webinaret når man vil, og velger hvilke oppgaver som man synes passer til sitt undervisnings nivå og sånn. Hva tenker du om å lære programmering og bruke programmering i et kurs som det her som gir 7,5 studiepoeng?

00:27:53 Lærer
Jeg synes det er viktig, fordi at når læreplanene plutselig nå sier at vi skal programmere for alle. Så er jo det et veldig stort mål når en veldig liten andel av lærerstaben i Norge har kompetanse. Så jeg synes jo det er bra at det blir lagt til rette, men jeg må jo si at det er jo ikke... Jeg er veldig glad for muligheten til å ta det, men det er veldig få... Altså, jeg vet om veldig mange som ikke får muligheten.
00:28:20 Intervjuer
Ja.

00:28:20 Lærer
Ja, og da synes jeg det litt ambisiøst at alle skal bruke det.

00:28:27 Intervjuer
Ja, så det blir et krav om at man skal kan det, men så får ikke alle en lik mulighet til å lære seg det?

00:28:33 Lærer
Ja.

00:28:37 Intervjuer
Tor du at det hadde vært nødvendig eller nyttig med flere kurs som bygger videre på de kursene som du har hatt?

00:28:47 Lærer
Ikke for min del nå, tenker jeg, fordi at jeg har nok kommet meg litt videre. Men altså jeg... Jeg føler ikke jeg kan mye, men jeg kan mer. Så nå tenker jeg at nå må jeg på en måte testet det litt ut og komme meg litt videre derifra. Men jeg vet ikke, noe sånn videregående kurs om noen år, kanskje, hadde vært nyttig. For da vil jo elevene nødvendigvis kunne mer, de som kommer fra ungdomsskole.

00:29:17 Intervjuer
Ja. Da har du litt mer grunnlag for å finne ut hva det er du trenger å lære?

00:29:21 Lærer
Ja.

00:29:24 Intervjuer
Ja. Hvis vi tenker på dem som ikke har muligheten til å ta kurs, eller både dem som har muligheten til å ha kurs, men også dem som ikke har muligheten. Så er jo det å lære programmering noe de aller fleste må nå. Hva tenker du at det er den beste måten å lære programmering på for lærere som er i full jobb?

00:29:47 Lærer
Nei, vi må jo bare... Jeg vet ikke jeg. Lese, og prøve og feile. Det er det jeg hadde gjort hvis jeg ikke hadde hatt det kurset er.

00:30:00 Intervjuer
Ja.

00:30:04 Lærer
Det er jo litt sånn man måtte lære GeoGebra en gang i tiden. Det var jo prøving og feiling. Men jeg ser jo at det er en utfordring, for spesielt de her som er litt eldre. Det er jo folk som velger å gå av med pensjon akkurat nå, fordi at så slipper de å forholde seg til dette her.

00:30:20 Intervjuer
Ja. Tror du at de som har muligheten til å gå på sånne kurs som NTNU-kursene vil ha en fordel når det kommer til å lære programmering?

00:30:32 Lærer
Jeg tenker det. Fordi at det å faktisk få muligheten til å bake det inn i arbeidstiden og å bruke tid på det gjør jo at jeg kanskje får brukt mer tid på det enn hvis jeg måtte gjøre alt på kvelden.

00:30:47 Intervjuer
Ja, så det er litt tiden tilgjengelig som spiller en rolle også, i forhold til om man får mulighet til videreutdanning, og ikke?

00:30:54 Lærer
Jeg tenker jo det. For jeg vet jo at, spesielt lærere i ungdomsskolen, de har jo ikke tid til å ikke tid til å planlegge timer som det er, og hvis de da skal lære seg noe helt nytt i tillegg så blir jo kvaliteten deretter, for du har ikke tid.

00:31:08 Intervjuer
Ja. Du nevnte jo at du har prøvd og programmert litt sammen med elevene dine allerede. Har du noen gang vært i en situasjon der du måtte ha lært noen programmeringskonsepter, eller at du må ta lært noe nytt innen programmering sammen med elevene dine?

00:31:29 Lærer
Ikke foreløpig. Fordi at jeg har jo på en måte hatt en progresjon som har ligget ti steg foran elevene, så foreløpig ikke. Men jeg tenker at det er jo en del av det med utforskende læring. At jeg tenker jo at... Poenget her er ikke at læreren kan alt. Det må man jo... Etter hvert, så vil noen elever kunne mye, andre kunne lite, og vi må på en måte være litt veiledere kanskje, mer enn... Ja, vi kan ikke alt. Sånn er det bare.

00:31:59 Intervjuer
Ja. Da har jeg et siste spørsmål. Og det handler om det at å lære programmering tar jo lang tid. Og den læringsprosessen går langt utover dem videreutdanningskursene som man har på NTNU. Og den lange læringsprosessen kan jo støttes av nye kurs, eller seminarer, eller deling av læringsmateriale, organisering av sånne CoT, altså fagforum... Listen er lang over ting som kan støtte en sånn læringsprosess. Hva tror du er den beste måten å støtte opp en sånn lang og kontinuerlig læringsprosess?

00:32:42 Lærer
Jeg tenker jo det med delinger av erfaring og opplegg. Det tenker jeg vil være nyttig. Så tror jeg kanskje det med kurs innimellom... Jeg har vært på matematikk-konferansen i Trondheim noen ganger. Og sånn som der, der går du for å få faglig påfyll. Jeg tenker det er litt viktig å ha påfyll av det som er med programmering og, som jeg har der. Og selvfølgelig andre, sånne fylkesvise konferanser også. Ja litt ja.. Egentlig kurs og erfaringsdelingen og deling av opplegg.

00:33:19 Intervjuer

Ja. Da har jeg egentlig fått svar på alle spørsmålene mine. Men har du noen andre kommentarer eller tilbakemeldinger eller refleksjoner som du ønsker å dele?

00:33:33 Lærer
Nei, jeg tror egentlig jeg er delt det jeg har tenkt på.

00:33:35 Intervjuer
Ja. Men da er det bare for meg å si tusen takk for at du stilte opp. Det er jo veldig nyttig for fagstaben å ha muligheten til å forbedre oppleggene basert på å snakke med dem det faktisk gjelder. Så vi setter veldig pris på at du tok deg tid til å ta en prat med oss da.

00:33:54 Lærer
Ja, jeg tenker jo det viktig at dere faktisk også tar tilbakemeldinger, sånn som nå, og at dere bryr deg om tilbakemeldinger, for det vil jo gjøre kursen bedre, kanskje etter hvert. Så det er jo det er veldig bra.
00:34:06 Intervjuer
Det er jo noe som vi er veldig opptatt av å bruke tid på, så...

00:34:10 Lærer
Ja, men det er bra.

00:34:13 Intervjuer
Ja, men tusen takk igjen, så får du ha en fin mandag videre.

00:34:18 Lærer
Takk i like måte.

00:34:19 Intervjuer
Ha det bra.

Transkripsjon 3

00:03:46 Lærer
Hei hei!
00:03:47 Intervjuer
Hei sann!
00:03:48 Lærer
Ja. Da var du der. Hører du meg?
00:03:53 Intervjuer
Det gjør jeg. Hører du meg godt?
00:03:55 Lærer
Det er veldig bra.
00:03:58 Intervjuer
Jeg kan jo starte med å introdusere meg selv, sånn at du vet hvem det er du snakker med. Jeg heter Miriam. Jeg er masterstudent på informatikk på NTNU, og så har jeg en deltidsjobb som forskerassistent for Majid på det her prosjektet som handler om fra å lære til å undervise programmering. Så det er jeg som gjennomfører intervjuene. Og som du sikkert legger merke til så blir den samtalen her tatt opp, og det er også jeg som skal transkribere intervjuet da. Så opptaket blir slettet med en gang det har blitt skrevet ned på papiret.

Ønsker du at jeg skal sende den transkripsjonen når det er ferdig, sånn at du kan få se gjennom?
00:04:44 Lærer
Betyr det på en måte hva jeg har ment med..?
00:04:49 Intervjuer
Ja, om du ønsker jo sjekke at det stemmer det som står i transkripsjonen da.
00:04:53 Lærer
Ja, det kan jo være greit ja.
00:04:59 Intervjuer
Ja. Da noterer jeg det. Så sender jeg det til deg. Jeg er ikke helt sikker på når det blir ferdig transkribert, fordi at jeg har en masteroppgavefrist selv, så det kan ta litt tid. *ler*
00:05:01 Lærer
Ja. *ler*. Ja, vi har alle frister i disse tider.
00:05:17 Intervjuer
Ja, det er veldig mange frister i den her den her tida ja.
Som jeg skrev i mailen, så består intervjuet av fire deler, og i de to midterste delene så kommer jeg til å vise deg noen ordskyer, så er du forberedt på det.
Har du noen spørsmål før vi starter?
00:05:36 Lærer
Nei.

00:05:38 Intervjuer
Nei. Da starter jeg littegranne med en del om din bakgrunn, knyttet til programmering da. Hvilke fag underviser du i?
00:05:49 Lærer
Matematikk og naturfag. Og i matematikk, da har jeg egentlig alle kodene bortsett fra R1 og R2 i år. Altså, ja, ikke i år. Men altså jeg har ikke undervist R1 og R2, men ellers alle matematikk-kodene på videregående. Både på yrkesfag og på studieforberedende. Og på.. Ja. Så på 1P, 2P, 2PY, 1PY, 1T, S1 og S2 på matematikken. Og naturfag også, der også er det studieforberedende og yrkesfag. Og yrkesfag, både elektro og service og samferdsel.
00:06:37 Intervjuer
Ja, og tror du at du kommer til å undervise programmering i alle de her fagene?
00:06:43 Lærer
Det jeg har prøvd i år, det er iallfall 1P, og naturfag både på forberedende og på elektro. Ja. Ja, også i S1 i år, men det er ikke nye læreplaner, så der har jeg ikke blandet inn noe. Jeg har jo ikke hatt det før heller.
00:07:03 Intervjuer
Ja. Og da underviser du på alle trinn på videregående da, fra første til tredje?
00:07:11 Lærer
Ja, det gjør jeg.
00:07:13 Intervjuer
Hvilke erfaringer med programmering hadde før du deltok på NTNU-kursene?
00:07:21 Lærer
Veldig gammel kunnskap. 30 års gammel kunnskap, så det er derfor jeg frisker opp igjen da med python, fordi at selv om konseptene er like, så er det jo et nytt språk. Og det er litt sånn veldig lenge siden sist og ja.

00:07:40 Intervjuer
Ja. Har du undervist i programmering før du tok kursene?
00:07:47 Lærer
Nei, det har jeg ikke.
00:07:51 Intervjuer
Takk. Da går jeg over på neste del, som er utfordringer du har opplevd med å lære programmering. Og da tenker jeg mest på de kursene du har hatt på NTNU nå da.
00:08:03 Lærer
Ja, med å lære selv ja.
00:08:05 Intervjuer
Ja, det som du synes har vært utfordrende å lære selv ja. Hva tror du at er de vanskeligste programmeringskonseptene å lære?
00:08:19 Lærer
Jeg ser i alle fall når det gjelder python at når vi har så mange sånne syntakser som må være på plass, så synes jeg på en måte når jeg begynte med funksjonene og alt som måtte defineres og alle de tingene som på en måte... I matematikken er unødvendig, på en måte. Og alt det må være på plass for at funksjonen skulle skrives ut, det synes jeg på en måte... Da begynte jeg å liksom... OK, det må være på plass, og det må være på plass og sånn. Og sånn tenker jeg også litt for elevene sin del også, alt sammen på en måte. Når det blir store koder for en liten ting. Ja. Også nå har jeg prøvd litt på raspberry med tanke på en anvendt programmering, og der også er det ganske mye som må hentes opp av eksterne bibliotek. Og å vite hva slags bibliotek som inneholder de forskjellige kodene og hvor vi skal finne tak i det synes jeg har vært utfordrende da. På en måte, hvor er det jeg henter informasjonen for å på en måte gjennomføre programmeringen. Jeg vet hva jeg vil, men jeg vet ikke på en måte kodene... Jeg kan ikke på en måte en kode som skal til for å gjennomføre det.
00:09:42 Intervjuer
Ja. Det blir kanskje litt ekstra utfordrende når man skal bruke biblioteker som innebærer at man ikke får se koden som blir brukt.
00:09:50 Lærer
Ja, og jeg vet ikke om de bibliotekene også. Jeg spurte kollegaen min som har litt ferskere erfaring, og liksom hvor finner du dem? Og så sier han «Nei, det er bare å google». Men ikke sant, du må jo vite hva du skal google etter.
00:10:05 Intervjuer
Ja, ikke sant. Ja, det er veldig sant. Det hjelper ikke bare å google, du må logge etter de riktige tingene også.
00:10:08 Lærer
Ja, og da må du vite hva du skal google etter ja. Og som sagt, jeg har såpass gammel erfaring at jeg er jo kanskje av den generasjonen som slo opp det som vi skulle ha av koder. Vi har ikke google *ler*. Vi hadde ikke internett, sant.
00:10:34 Intervjuer
Ja, litt andre utfordringer den gangen.
00:10:34 Lærer
Ja, vi bladde opp og slo opp i en bok, der vi fant på en måte forskjellige koder og konsepter og sånt. Men det er jo ikke... Så det er på en måte å kombinere programmeringen med hvor du skal finne faktisk det du skal gjøre. Det syns jeg er utfordrende ja.

00:10:55 Intervjuer
Ja, det kjenner jeg igjen jeg også. Å innhente den informasjonen man trenger kan være utfordrende til tider.
00:11:03 Lærer
Ja.
00:11:05 Intervjuer
Da tenkte jeg skulle vise deg en første ordsky. Den ordskyen er laget basert på refleksjonsnotatene fra høst-kurset grunnleggende programmering for lærere, der lærerne har reflektert litt over ting som de synes har vært utfordrende da. Så skal vi se om vi får opp det her da, det er jo alltid like spennende. Der. Ser du den nå?
00:11:43 Lærer
Ja.
00:11:45 Intervjuer
Ja. Har du noen tanker om den her ordskyen? Noe som burde ha vært større, eller mindre, eller noe du savner?
00:11:50 Lærer
Ja, jeg ser at de har skrivet writing code and pseudocode. Selve koden synes jeg ikke på en måte var det vanskeligste, så jeg hadde kanskje laget den litt mindre.
00:12:03 Intervjuer
Ja.
00:12:05 Lærer
Men det med feilmeldinger, det synes ikke enkelt for meg. Altså, disse er på en måte... Det er bare å tolke koden, ikke sant. Eller når du får feilmelding så er det bare å tolke den feilmeldingen, og så rette opp. Jeg synes ikke nødvendigvis at det har vært så enkelt, fordi at selv om det står en beskjed om at den feilen ligger på den og den linja, så ikke nødvendigvis at en som er fersk vet hva det betyr, den tilbakemeldingen som står i en sånn tilbakemelding da. Og så er en annen utfordring, det når vi er litt... Vet ikke hva jeg skal si, men gammel kunnskap i forhold til engelsk. Dette her er jo egentlig engelsk på alt sammen. Ja, jeg synes på en måte vært en utfordring. Ser jo at du gir ord skyer til og med på engelsk til meg. *ler*. Så det er sånn. Jeg synes ikke loops, altså løkker, har vært utfordrende. Det er fordi at det er på en måte et gammelt konsept. På en måte, det konseptet har jeg vært gjennom så lenge, sånn at jeg ser ikke den som den samme utfordringen, fordi der vet jeg på en måte hvordan programmet skal bygges opp for å få til en løkke.
00:13:24 Intervjuer
Ja.
00:13:25 Lærer
Ja. Lister var nytt for meg. At du kunne plassere ting i lister. Så på en måte, jeg hadde nok kanskje uthevet den litt større.
00:13:35 Intervjuer
Ja. Det er det flere som sier. Der vet jeg ikke hva som har skjedd, fordi det er flere som sier at lister er blant det mest utfordrende, men den er fortsatt liten.
00:13:47 Lærer
Ja, men det kan være... Jeg tenker grunnen til at den er såpass liten på oversiden kan være faktisk at vi glemmer i det helheten da, fordi at vi har bruker det lite.
00:13:59 Intervjuer
Ja, ikke sant.

00:14:02 Lærer
Men å øve nok, står det practicing enough. Den ser jeg utfordringer, både med når jeg skulle lære selv og ikke minst når jeg hatt elever nå altså. Ja, for det du skal overføre den kunnskapen, så må de ha interesse for å sitte lenge med det.
00:14:16 Intervjuer
Ja. Du ser utfordringer med den mengdetreningen som kreves?
00:14:27 Lærer
Mengdetreninga ja. Både å ha tid til det, og at elevene på en måte skal se at det at det hjelper.
00:14:39 Intervjuer
Ja. Tror du at noen av dem utfordringene med å lære programmering som lærere og du opplever kan ha noen ting med hvordan kurset er tilrettelagt?
00:14:52 Lærer
Jeg snakker du om meg nå, eller snakker du om elevene mine?
00:14:56 Intervjuer
Deg.
00:14:58 Lærer
Meg ja. Jeg synes det var helt greit sånn som det ble lagt opp. Føler kanskje det at jeg trengte kanskje faktisk mer støtte i høst, når jeg på en måte tok selve programmeringsbiten. For da skal jo alle disse konsepter inn. De som vi ikke kan. Og så får du en del lese-lekser, for meg da, i en engelsk bok som på en måte gir en veldig stor kneik. Og måtte lese det på engelsk. Og selv om at jeg leste på engelsk selv når jeg studerte for en del år tilbake, så er det gammelt. Og da trenger jeg egentlig mer støtte for selve konseptet. Det var en del sånne videoer ble lagt ut, men de var på det enkle nivået. Altså, hvis du skjønner? Det er litt samme utfordringene som elevene, ikke sant. Du gir en introduksjon til tema, og så ber jeg elevene gå videre. Og så sitter vi litt på egen hånd. Ja. Men jeg syns det var kjekt å kunne bruke det gruppeprosjekter på slutten, for da fikk vi egentlig bruke våre styrker og svakheter på en måte, og hjelpe hverandre da.
00:16:01 Intervjuer
Ja. Forstår jeg det riktig at du tenker på at det at det mangler litt norsk materiale på de litt mer videregående programmeringselementene som litt utfordrende?
00:16:25 Lærer
Ja, jeg tenker også det på en måte... For nå er det på en måte... Vi er jo lærere som på en måte tar etterutdanning. Og vi har høy utdanning fra før av og sånne ting. Men det er ikke nødvendigvis sånn at all kunnskap er fersk. Og spesielt ikke på språk. Så jeg kjenner at det var nesten en sånn ny kneik for meg, så jeg har faktisk brukt en del av en 1T-matteboka bakerst innimellom, og brukt det som et oppslagsverk for meg selv. Mens jeg hadde den engelske varianten ved siden av. Ja, så brukte jeg den som på en måte var enkel å forstå. Hente inn... Hva var konseptet? Hvor skulle kolon stå? Hvor skal det være innrykk? Hvor skal jeg gjøre av disse små tingene som gjør at koden fungerer? Så et sånt enkelt norsk oppslagsverk for kodene, det hadde hjulpet meg da.
00:17:28 Intervjuer
Ja. Og da tenkte jeg vi kunne gå videre på del tre. Og det handler om utfordringer med å undervise i programmering. Så nå vil det gå litt mer på utfordringer som elevene har, eller kommer til å ha. Og da lurer jeg på hvilke deler av programmering tror du er mest utfordrende for elevene dine å lær?

00:17:54 Lærer
Jeg tror det er mange ting. Jeg kan jo si litt erfaringer. Det som jeg prøvde på i år da, og jeg har hatt... Spesielt 1P har jeg prøvd, og elektro. Eller, 1P prøvde jeg i høst. Og jeg prøvde også nå elektro gjengen litt sånn systematisk nå på våren med tanke på at jeg holdt på med den anvendte programmeringen da. Og 1P, vi var jo veldig usikker på læreplanene. hvor mye programmering som skal inn, men vi satte i gang med en gang for å på en måte være i forkant hvis det kommer opp eksamen. Og da synes jeg kanskje det som var utfordrende egentlig, var å få elevene mine motivert til å forstå at de kan brukes dette til noe fornuftig. Fordi programmene blir så enkle på det nivået.

00:18:43 Intervjuer
Ja.
00:18:47 Lærer
Altså. på en måte. at det er faktisk enklere å slå inn programmet på en kalkulator enn å programmere oppgaver, sant. Fordi at du på en måte må starte så enkelt. Og da er motivasjonen veldig laber, og det forstår jeg. Ikke sant, du skal bruke veldig masse på kolon og punktum og innrykk, og alt sammen, og kalle det med riktig navn og alt sammen. Få på plass variablene. Og elevene ser ikke hensikten med det. Altså, det er ikke noe mening i programmering enda. Fordi at det er for enkelt, på en måte, den matematikken de skal utføre med programmeringen.
00:19:25 Intervjuer
Ja, så det med å skrive syntaks blir liksom en sånn kompliserende del av da. At det ikke oppleves som nytteverdi?
00:19:31 Lærer
Ja, det kompliserer i stedet for å forenkle. Ja, det er tall. Det tar litt tid før de i matematikken er kommet dit at, på en måte, for eksempel en løkke kunne vært nyttig, sant.
00:19:34 Intervjuer
Ja.
00:19:43 Lærer
Ja, og de har jo regneark for før av, som har brukt før. Så de ser kanskje ikke programmering som noe hensiktsmessig enda på det nivået, tenker jeg. Nå endrer vi... Vi hadde jo bare et halvt år med den, og så endret jo UDIR og sa på en måte at det blir ikke en del av en kommende eksamen. Så vi la det litt på hylla. Men vi bruker jo det litte granne i sånn anvendt i naturfag med samme denne gjengen da. Så på en måte, de har... Vi har jo kunnet tatt i bruk sånne praktiske program litt, men kanskje ikke samme opplæringen innenfor naturfag da, der rekker de ikke det så veldig mye.
Ja, men nå har jeg jo kurset på anvendt. Ja, anvendt programmering. Og da tenkte jeg at nå skal jeg prøve elektro-gjengen som enda ikke hadde prøvd noe. Og på den praktiske retningen, sant. Det var egentlig i 1P at jeg tok den teoretiske retninger inn. Så tenkte jeg at jeg nå tar den praktiske retningen, og tester ut den. Og de også var blanke fra ungdomsskolen. Det var en elev som har hatt valgfag med programmering. Eller så hadde de hatt null niks fra før av. Så det jeg gjorde der var å gå veien via micro:bit. Ja, så de fikk på en måte blokkprogrammere litt via microbiten på egen hånd. Og tok det konseptet veldig fort. Og begynte å se veldig masse hva de kunne finne på. Altså, ikke sant, den der nysgjerrigheten. Ja, der kan vi sende meldinger etter hverandre, så kan vi gjøre sånn, og litt sånn. Og så begynte de å prøve det ut og teste ut litt. Det synes jeg det måtte være en mye mer motiverende retning å gå for elevene. At de fikk gå via den praktiske biten og teste ut

litt – hva er det jeg kan bruke dette her til? Ja, så jeg prøvde liksom å teste ut da på en måte å gå fra den blokkprogrammering over til python. Og den utfordringen synes jeg fortsatt er hard altså. Altså, på en måte, de de har for lite bakgrunnskunnskap, og for lite tid. Elevene treffer det hver 14. dag, og vi skal ha det sammen med naturfag, sånn at det på en måte... Det blir bare et lite drypp hver gang. Ja, så den er mengdetreningen som de trenger for å på en måte lære seg det skikkelig. Det har ikke vi i videregående skole, det bare drypp inne i de ulike andre fagene. Og ikke programmering som fag, vet du, så ja.

00:22:16 Intervjuer

Så det er litt vanskelig å få til den mer kontinuerlige læringen som programmering egentlig krever da?

00:22:22 Lærer

Ja. Jeg tenker at det på en måte... Men jeg hadde jeg kunne fått valgt, så hadde jeg nok gått videre praktiske sånn. For nå er det jo interessert, denne elektrogjengen min. I motsetning til 1P, som på en måte gjerne kunne tenkt seg at vi la vekk programmering for alltid *ler*.

00:22:40 Intervjuer

Ja. *ler*. Vi får håpe at programmeringen blir en naturlig del av matematikken og etter hvert da. Det er jo sikkert en sånn overgangsfase.

00:22:48 Lærer

Ja, jeg håper jo på en måte at de kommer med litt mer grunnkunnskaper, ikke sant. At akkurat det som jeg gjort med elektrogjengen på en måte kan ligge litt i det, og at de har holdt på litt med blokkprogrammering, på en måte, fra før av.

00:23:02 Intervjuer

Når du tok i bruk programmering både i 1P-klassen, men også elektroklassen, opplevde du at det var noen programmeringskonsepter de hadde mer utfordringer med? Som for eksempel løkker, eller du nevnte jo variabler, at det vanskelig å få til at variablene stemte?

00:23:22 Lærer

Ja, jeg opplevde en ny utfordring når jeg begynte med elevene nå, spesielt med elektro da, som jeg egentlig aldri har tenkt over. Men kan vi snakker om punktum, komma, innrykk, alt det der, som vi må være nøye med. Elever i dag, de er ikke nøye de. *ler*. Om det er en dobbel konsonant, en enkel konsonant, om en stor bokstav eller en liten bokstav, litt sånn. Og dette ødelegger jo for hele gjennomføringen av en kode. Så, de er på en måte ikke opplært til at ja... Du mister ikke sekseren din når du har glemt store bokstav i en skriveoppgave, men nå plutselig mister de hele programmet sitt for en bokstav. Det er jo flinke elever også, som har skrevet alt riktig. Men jeg skal være essens med C stedet for med s, ikke sant? Altså, en plass har de gjort det, og det var liksom nok til at du... «Åh, ja, ja men, ja». Så er det bare sånn litt sånn sukk. Sånne utfordringer er det som frustrerer en 17-åring. Det er på en måte aldri blitt pirka på det.

00:24:37 Intervjuer

Det er kanskje ikke er samme behovet for å være så streng på rettskrivingen i andre fag eller i andre elementer i programmering?

00:24:46 Lærer

Ja, og det er jo en... Altså, det er på en måte en ting at ved masse skrivefeil i norskoppgaver, så trekkes det selvfølgelig karakterer. Men her mister jo det nesten... Her blir det jo nesten stryk, for å si det enkelt, når du når du mangler sånne småting. Jeg synes det er småting da. Det er nå... Men ellers konsepter, sånn i programmering, så tenker jeg på en måte at de klarer fint å lage formlene. De det klarer det enkle... Altså, elevene som på en måte svakere klarer å lage formler etter hvert, de må på en måte bare øve seg på det. Men å tenke nye tanker, at du kan sette ting inn i ei løkke for at det skal gjenta igjen og igjen. At på en måte her legger du til en, og så kjører løkka en gang til. Også kjørte løkka en gang til. Det syns jeg er vanskelig å overføre da, av kunnskap. Ja.

00:25:45 Intervjuer

Ja. Da har jeg en ordsky til, som er tilsvarende som den i sted, bare at nå går det på det som er utfordrende for elevene å lære da. Skal vi se her. Da har du jo allerede vært litt inne på enkelte ting her, men har du noen tanker om den ordskyen her?

00:26:08 Lærer

Ja, jeg er helt enig i at loops skal være med i den ordskyen så ganske stort ja. For det er jo det som på en måte, som vi ser det i forhold til 1P da, så det er det som er den lille nytteverdien av programmering, som de nesten ikke forstår. Og derfor så har den på en måte ingen nytte. Da er det jo egentlig bare en, da er det jo bare formler som skal settes inn i et vanskeligere program. Ikke sant. CAS som er i GeoGebra gjør jo like i like mye, eller like enklere. Der kan du på en måte sett inn formel, og så kan du bare legge inn variabelen etterpå. Markere og trykke på x er lik, og alt løser seg uten noen utfordringer. Så det er jo loop i hvert fall på den....

Feil ja, altså, det er veldig... Det var det jeg snakker på en måte de kaller det for småfeil, men for python så blir jo alt feil. Ikke sant. Hvis du akkurat... Skriver et komma eller et punktum eller kolon, eller mangler et kolon eller et eller annet sånt. Og de forstår heller ikke i starten feilkodene som kommer, altså på en måte å tolke kodene.

00:27:30 Intervjuer

Ja.

00:27:33 Lærer

Ja, og altså, det å har ny tanker med tanke på variablene. Det tror jeg også. De har jo nok utfordring faktisk på papiret, og med formler. Og hvordan sammenhengen mellom 2 størrelser. Og nå så bruker de jo GeoGeobra, de bruker regneark, og nå skal på en måte programmering også på en måte bare ha en ny... Så det er veldig mange konsepter... Hvis du tenker på en 1P elev, så skal de skrive formelen på egen hånd i GeoGebra, i et regneark og i programmering.

00:28:16 Intervjuer

Ja.

00:28:17 Lærer

Og når de nesten knapt vet hvordan en formel skal skrives i utgangspunktet. At du faktisk må et likhetstegn mellom en sammenheng. Altså, på en måte har noe på venstre side og på høyre side, så dette her er kjempestor kneik for svake elever i hvert fall. Så det er jo... Det er de elevene som ser nytte med det her, som kan nå langt, tenker jeg. Jeg må ikke glemme de altså oppi dette her, men vi skal jo på en måte undervise alle. Vi underviser jo ikke dem som da har veldig lyst på programmering. Vi underviser alle nå, og det er... Vi må på en måte ha en annen tilnærming. Så det jeg gjorde med prosjektet også nå, det var jo det at jeg på en måte lagde en utviding. Jeg lagde en innfallsport som gikk an å bruke for alle, men på en måte så tipset jeg om litt om utviding. Altså som folk på en måte som kan komme lenger kan få ta bruke av.

00:29:20 Intervjuer

Ja. Så tilpasser litt til nivå?

00:29:25 Lærer

Tilpasse litt, men ikke sant, det handler om interesse, hvis ikke sitter det ikke i den programmeringen heller.

00:29:31 Intervjuer
Nei, absolutt ikke.
00:29:33 Lærer
Nei, det det må det. Så det er på en måte... Jeg har nok fått litt erfaring i år da, på hvilken retning jeg har lyst å fortsette med, på en måte, når jeg skal undervise da. På en måte skal ha noen hensikt med det.
00:29:48 Intervjuer
Absolutt. Nå har vi snakket litt om utfordringer som, eller ting som kan være utfordrende for elevene. Har du noen tanker om hvordan du ønsker å gå fram for å for å adressere dem utfordringene i undervisningen? Som for eksempel, hvordan skal du gå fram for å lære elevene at skrivefeil, eller syntaksen er veldig viktig?
00:30:18 Lærer
Det vet jeg nesten ikke hvordan jeg skal klare å motivere for i denne sammenhengen. Uansett så tenker jeg at jeg hvis de kom inn på nytteverdi. Altså at den har lyst å lære... Den der motivasjonsbiten, hvis den er der, da er det enklere å lære de andre tingene også. Det er på en måte... Du setter ikke i gang med noen ting som du sier sukk til før du har begynt. Litt sånn, da kommer du ikke i gang. Da vegrer du rett og slett for å starte med det. Ja, så jeg tenker at på en måte med hvert fall innfallsporten, å bare se nytteverdi først. Ja, sånn at de har et konsept der, så tror jeg på en måte det blir enklere å ta i bruk de ulike konseptene. Men så ser jeg også utfordringer nå selv, at på en måte for å få noen nytteverdi... For å se den der «oi, nå skjer det, nå skjer det», så er det jo allerede avanserte program for eleven.
00:31:28 Intervjuer
Ja.
00:31:28 Lærer
Ja, da er det allerede kommet på et nivå som er avansert for dem. Og det er ganske mange ting som må forklares i det som de da har gjort, for at de skal på en måte forstå konseptet de har vært gjennom. Men jeg tror kanskje via blokk... Jeg tror det er riktig veien å gå via blokk, for da ser de det litt synlige og kan, på en måte, klippe og lime koden selv.
00:31:54 Intervjuer
Ja. Tenker du da at det er fornuftig å bruke blokkprogrammering for at de skal lære konseptet først, og så går videre på den skrivingen?
00:32:03 Lærer
Ja, ikke sant. Det var jo det som vi på en måte hadde håpet at elevene våre har vært borti før de kommer til oss. Og i år, vet du, så er det på en måte sånn at vi starter med python, mens de ikke har vært borti blokk, ganske mange. Ja, og det er der på en måte utfordringen ligger, ikke sant. De får ikke kjørende program. De får ikke til programmet. De forstår ikke matematikken bak programmet. Og de har heller ikke sett noen nytteverdi på forhånd. Og de har på en måte ikke sett den. Så forhåpentligvis blir det jo bedre med årene da. At de på en måte kommer fra barne- og ungdomsskole med litt mer, i alle fall blokkprogrammeringskunnskaper.
00:32:45 Intervjuer
Ja. Du nevnte jo på utfordringer for deg selv, at mangelen på norsk læringsmateriale opplever du som utfordrende. Ser du at det kan være knyttet til utfordringer for elevene også?
00:33:05 Lærer
Det kan det nok. Men jeg synes nok kanskje elevene er flinkere i engelsk enn meg. Altså, de er jo på en måte i en annen verden, ikke sant. De er mer i spillverdenen. Så de lærer seg

engelsk nesten samtidig som norsk. Om de kan den riktige engelske ordene, det er en annen sak. Men de har jo gode engelskkunnskaper, ungdommer i dag.

00:33:29 Intervjuer

Ja. Det er jo mye mer tilgang på film og spill og alt på engelsk nå kanskje?

00:33:35 Lærer

Ja. De starter med spill ganske tidlig også. Og det er jo... For å få de gøye spillene, så er det jo engelsk med en gang. Og så kommuniserer de med hele verden. Så fort de får lov til å kommunisere, så kommuniserer de med hele verden med på engelsk.

00:33:52 Intervjuer

Ja. Det kan du være nyttig. Det er jo mye av den terminologien som brukes i spillverden, som også blir veldig nyttig programmeringsverdenen.

00:34:05 Lærer

Ja, det er det. Og jeg har en datter som tok datateknologi, og hun har også i havnet i spillverdenen. *ler*.

00:34:15 Intervjuer

Å ja! De er nært knytta.

00:34:17 Lærer

Ja da. Ja, har brukt henne som sparringspartner innimellom ja. Ja, men hun også sier det at alt går jo på engelsk, så da må du bare lære deg engelsk.

00:34:28 Intervjuer

Ja. Det husker jeg at jeg også synses var veldig utfordrende første året. Jeg var heller ikke så sterk i engelsk, men det kommer seg fort da. Man blir jo tvunget til det når man går inn for å lære programmering.

00:34:39 Lærer

Ja, men ikke sant, for oss så er det jo samme som lærere. Dette her er bare en del av fagene som vi underviser, sant. Jeg underviser i matematikk og naturfag, jeg undervisning ikke i programmering, så ikke sant. Men det er den der biten – hvor mye skal du putte inn for? Samtidig skal du gjennom det pensumet som du da faktisk har å komme gjennom i andre deler av faget.

00:34:48 Intervjuer

Ja, ikke sant. Jeg glemte forresten å spørre om en ting på den forrige ordskyen, så da må jeg bare gå tilbake til det. Hvis man sammenligner de to ordskyene... Altså, den første var jo utfordringer som lærere opplever selv, og den andre var utfordringen lærerne ser at elevene kan ha. Og elevskyen er mye mer konsentrert om noen få områder, mens for lærere så er det mer sånn jevnt spredt utover flere områder. Har du noen tanker om hvorfor det kan være sånn?

00:35:41 Lærer

Ja. Jeg tror egentlig at vi ikke har kommet lenger med elevene, så vi har på en måte... Ja, vi har ikke kommet... Altså, elevene har kanskje litt om funksjoner inn på naturfaget, da vi på en måte har programmert lungefunksjonen, og strålingsintensiteten og moskus-populasjoner og litt sånne ting. Ellers så har vi på en måte ikke kommet... Vi kommer ikke så langt inn med elevene våre, så det er jo bare noen få konsepter elevene våre har vært borti. Så det er naturlig... Det er på en måte at når du kommer til den kneika med elevene, ikke sant. Sånn som med løkker, så er det det som kanskje er faktisk toppnivået på det, og så langt vi kom. Kanskje noen har vært borti litt funksjoner, men det er jo egentlig variabler og litt for-løkker, while-løkker, og kanskje noen har vært borti litt funksjoner når de går over på naturfaget. Ja, det er vel egentlig så langt jeg tenker at elevene våre kommet i løpet av VG1 hvis de i det heletatt har prøvd seg på noe. Og derfor så blir jo alt mye større, fordi vi som er lærere, vi har vært gjennom på en måte ganske mye nå. Og da ser vi på en måte mange flere konsepter som kan tas i bruk. Ja, men så langt er ikke egentlig elevene våre kommet

00:37:07 Intervjuer

Så grunnen til at for eksempel lister og strukturerte datatyper ikke er med i elevskyene, er fordi at man rett og slett ikke har kommet til det punktet at det er inkludert?

00:37:17 Lærer

Ja, det tror jeg nok ja.

00:37:19 Intervjuer

Ja, men det tror jeg nok er et veldig godt poeng.

En utfordring med at programmeringen brukes som en liten del av fag, er jo at man ikke får den kontinuerlige læringen som vi snakket om i stad. Og da vil det jo ofte oppstå den utfordringen at elevene lærer å skrive en løkke i en time, og så når de kommer tilbake neste gang, så husker de ikke hvordan en løkke fungerer. Hvordan vil du gå fram for å sørge for at eleven lærer programmeringskonseptene over tid?

00:38:01 Lærer

Det synes jeg egentlig er en utfordring. Fordi at, som sagt, de har det i ulike fag. De har de med ulike lærere. Du også vet at det er forskjellige måter å bygge opp et program på.

00:38:19 Intervjuer

Absolutt.

00:38:19 Lærer

Nå sier vi at det er veldig nøye med syntakser. Men ikke sant, hvis da på en måte en lærer har en måte å si «sånn må du gjøre», mens en annen sier «sånn må du gjøre», så har de forskjellig læreverk. Sånn som man på en måte er slår opp i. Som har ulike måter å gjøre ting på, så er det ikke rart at de ikke husker. Det er ikke en gjentagende kode dem har, men dem har noen som sier noen ting om ei løkke. Altså det... Så på en måte går de inn i et klasserom, så sier en lærer en ting. Og så går de inn i et annet klasserom, og så sa en annen lærer... Også blir det sånn «Ja, men det sa ikke den læreren», sant. Så jeg tror at Norge egentlig har bomma litt i forhold til det med opplæring av dem. De burde heller ha et lite 2 timers fag som på en måte konsentrerte seg om programmering. Sånn at det blir lettere på en måte, sånn at det da kan tas i bruk i de andre fagene. At alle sammen fikk en lik opplæring da. Jeg tror det blir problematisk nå. Det blir bare sånn at det er ingen som har ansvaret for å lære opp elevene, ikke sant. For det er ikke nødvendigvis at det er samme lærer i matematikk og naturfag.

00:39:40 Intervjuer

Nei, så det blir litt lite konsekvent da? Måten man lærer bort konseptene?

00:39:41 Lærer

Nei. Veldig lite konsekvent, ja. Og spør etterpå om «hva du har lært dem?». Vi tok det, vi rakk ikke det, liksom. Og så er det i forhold til fagets art, ikke sant, og hvor masse skal du ha om det. Det blir mer sånn tilfeldig om du kommer inn på, som du sier, om det funksjonsstrukturer, om det er løkker og sånne ting. Så programmering blir litt mer sånn en del som noen lærere tar seg av, og andre legger litt til sides. For elevene så blir ikke det skikkelig opplæring, synes jeg.

00:40:16 Intervjuer

Nei. Da har jeg en siste ordsky. De to forrige har jo gått veldig på sånne spesifikke programmeringskonsepter, mens den siste ordskyen som jeg skal vise deg nå, det går litt

mer på utfordringer knyttet til å undervise programmering. Skal vi se om vi får opp det. Der. Ser du noen ting her som du kjenner deg igjen i? Og er det noe du savner?
00:40:56 Lærer
Ja, dette er ordskyen for utfordringer jeg kan ha når jeg underviser?
00:41:00 Intervjuer
Ja, utfordringer med å undervise da. Å inkludere programmering i undervisningen. Hvis det er noe som er uklart, så må du bare spørre. De er ikke alltid like lett å forstå, alle disse setningene. *ler*.
00:41:15 Lærer
Ja. Men progresjon... Progresjon har jeg jo snakka om da. Men pre-existing knowledge, altså det er forkunnskaper?
00:41:23 Intervjuer
Ja.
00:41:26 Lærer
Ja, de har jo ingen forkunnskap. Altså, det er de som har hatt et valgfag på ungdomsskolen som foreløpig nå har litt kunnskap. Og når det nevnes også motivasjon, så handler det om på en måte at vi må gi de en grunn til å lære programmering. Det tenker jeg også at er... Ja, sånn som 1P ble, så tenker jeg liksom... Sånn neste år, så tror jeg kommer til å ta fram microbiten, og vise dem hensikten med at ting kan programmeres, og viser dem programmer som på en måte kan få deg til å gjøre noe synlig. Ja, i stedet for å begynne teoretisk, sånn som det var lagt opp til matematikken. For en vet selv at man må bruke litt tid på programmering for å lære seg det, og hvis de da ikke har lyst, så er på en måte... Da er løpet allerede kjørt, tenker jeg, før vi har startet, ja. Så motivasjonen der må på en måte være en faktor som vi må legge inn før vi setter i gang, ikke på en måte å introduserer de for enkle konseptet hele tiden, enkle konseptet som vi skal få til, hvis de ikke ser hensikten med det.
For det står liksom interest and effort, og de prøver jo ikke. Det har ikke noe hensikt, en kan slå inn på kalkulator det vi lærer dem. Eller mobilen.
00:43:07 Intervjuer
Ja.
00:43:08 Lærer
Creating good teaching plans ja. Jeg tror vi må tenke på forhånd, alle sammen. Det har jo vært et prøveår i forhold til programmering i forhold til nye læreplaner. Veldig masse sånn prøve år i år, og så kommer den koronaen i tillegg som bare topper det hele.
00:43:22 Intervjuer
Ja. Det har jeg forståelse for. Jeg skriver faktisk masteroppgave om den digitale undervisningen under koronaperioden, så jeg forstår jo at det å slenge opp på litt ekstra programmering oppå en allerede krevende situasjon... Jeg tror det er et helt ekstremt prøver for dere lærer. Det har jeg full forståelse for.
00:43:46 Lærer
Ja, nei, det er... Vi er glad vi kommer til endes i år. *ler*
00:43:56 Intervjuer
Ja. Du har jo allerede vært litt inne på det med tid. Fordi programmering krever jo enormt mye mengdetrening over tid. Hvilke utfordringer ser du med den tida elevene har tilgjengelig for å lære programmering?
00:44:14 Lærer
Den er ikke eksisterende i videregående, rett og slett. Altså, så lenge de ikke velger et fag

som heter IT og skal se på programmering, så vil ikke en allmenn-elev, på en måte som ikke har valgt seg faget, de vil ikke lære seg det på videregående. Det er på en måte... Det er rett og slett fordi at det blir bare drypp. Og nå har vi kjørt til og med et prøveprosjekt på skolen med litt blokkundervisning i år i tillegg og alt mulig. Dette er jo fordi det er mye fordypning i de nye læreplanene. Derfor på den måten også, så blir det jo lengre i mellom hver økt, fordi at vi kjører flere timer i slengen. Og da kan det gå en uke til neste gang eller en og en halv uke til neste gang vi er... Og da skal elevene holde på med andre fag.

00:45:11 Intervjuer

Ja.

00:45:12 Lærer

Ja, så det er ikke den der jevne jobbingen. Nei, jeg tror på en måte at skal vi lære videregående elever skikkelig på en måte tekstbasert programmering, så må de få øvelse minst en gang i uka, og de bør ha et fag som er dedikert til deg.

00:45:38 Intervjuer

Ja, for du tenker at dem det må være tilrettelagt for at de skal gjøre det på skolen, for de kommer til å gjøre det hjemme?

00:45:46 Lærer

Det må i hvert fall være tilrettelagt sånn at de får drypp minst en gang i uka her på skolen. Du kan ikke... En 16-åring som på en måte ikke kan noen ting, den gjør ingenting i de 14 dagene på en måte. Nei, du må du må mate dem. Du må mate dem på en måte med interesse og forståelse for at du... På en måte, blir de stående fast, så slutter de der.

00:46:12 Intervjuer

Ja, men det gjelder jo for de fleste innenfor de aller meste områdene. *ler*. Hvis du bare står og stanger, så blir det ikke noe av.

00:46:22 Lærer

Ja, og så er det på en måte som du sier, hvis da lærer også sier at, ikke sant, «du skal ha prøve i kompetansemålene dine, mens programmering er tillegg», så legger elevene det også vekk. Så det kommer an på hva du vektlegger også.

00:46:37 Intervjuer

Ja, så du tenker at litt holdningene til lærerne knyttet til programmering og så kommer til å påvirke hvilke tanker elevene har om programmering?

00:46:45 Lærer

Ja, jeg ser jo dette store forskjeller på oss nå, ikke sant. Nå tok jeg tak og tenker at nå er det gammel kunnskap om programmering. Vi skal lære programmering. Ok, da må jeg lære opp meg selv på en måte for å på en måte ha kompetansen. Og det er jo andre lærere på huset som på en måte underviser på samme måten, men som ikke har kompetansen, altså som ikke har tatt noe ekstra i år. Sant. For dette her er jo frivillig. Altså, det er frivillig på en måte å oppdatere seg på en måte, og det er heller ikke alle som får det. Jeg tar det faktisk utenfor de 100% mine, fordi at jeg fikk ikke på vikarordninger. Jeg fikk bare stipendordningen. Så derfor så har jeg på en måte jeg tatt det som et tillegg, men det er jo ikke alle lærere som vil gjøre, og da vil det være ulikt hvordan lærerne prioriterer, ikke sant. Hvis du ikke behersker programmering så godt selv, så vektlegger en ikke det så masse i læreplanene heller. For dem ligger jo som på en måte overordnet del både i naturfag og 1P. Det legger ikke i kompetansemål. Så ja, det blir forskjell. Det blir forskjell på oss.

00:47:57 Intervjuer

Ja. Det er jo sikkert en overgangsfase på hvert fall noen år før at man får bedring innenfor det her da, i forhold til forkunnskaper både på elever og lærere.

00:48:11 Lærer
Ja.
00:48:14 Intervjuer
Da har vi kommet til siste del det av intervjuet som handler om lærerens faglige utvikling. Og da handler det litt om den der utviklingen av programmeringskompetansen over tid da. Og så er den delen delt i to sånne underdeler der den første går mest på NTNU-kursene. For du har hatt både det høst-kurset grunnleggende programmering, og vår-kurset anvendt programmering?
00:48:43 Lærer
Ja.
00:48:45 Intervjuer
Ja. De kursene er delt opp sånn at den høstdelen har fokus på å lære programmering, mens i vårdelen så handler det mer om hvordan du kan bruke programmeringen i undervisningen. Hva tenker du om den oppdelingen?
00:48:59 Lærer
Jeg synes det var en veldig fin oppdeling. Ja, det var helt greit. Veldig viktig å egentlig kunne den programmeringsbiten før en begynner med forskjellige duppedingser som skal programmeres. Ja.
00:49:13 Intervjuer
Greit å få på plass litt grunnleggende forståelse først, og så litt mer bruk av det senere?
00:49:16 Lærer
Ja.
00:49:22 Intervjuer
Det vårkurset anvendt programmering er jo lagt opp til å være et sånn veldig fleksibelt kurs med mye valgfrihet og muligheter til å gjøre ting når det passer for seg. Og hva tenker du om å lære programmering i et kurs som det her?
00:49:37 Lærer
Jeg syns det var både utfordrende og lurt på en måte. Fordi jeg så jo på en måte alt det han Majid la ut i forkant, på en måte, hva som vi kunne bruke og sånne ting. Og hva som kunne kjøpes inn. Å tenke liksom hva kjøper du inn i en sånne sammenhenger sånt? Men det jeg gjorde var å spørre han som da står for litt innkjøp på skolen, hva har vi? Hva har vi på forhånd? Og det som overrasket meg... Det var jo det at vi altså, vi har jo den microbiten, den har jo vært i bruk før. Men rasperry'en? Nei, det visste jeg ikke at vi hadde. Jo, vi har jo klassesett, men det ligger en pappeske ned i kjelleren som ingen har tatt i bruk. *ler*. Litt sånn typisk skolen, sant, ja. Men det har jo vi, men det er ingen som kan å bruke det, så derfor så ikke tatt i bruk. Ja, og det å kunne bruke ting som faktisk er utstyr som vi har. Vi får jo penger til både å kjøpe inn utstyr og alt mulig sånn. Men det er en terskel for lærere også på å ta i bruk nye ting som de ikke har på en måte kompetanse på å ta i bruk, ikke sant. Bruke litt tid på å sette seg inn i hvordan det fungerer og sånne ting. Og det syns jeg på en måte har vært smart med dette kurset her da, at nå har jeg fått tid til å sette meg inn i sånne ting som på en måte lå på huset.
00:51:07 Intervjuer
Ja. Synes du arbeidsmengden har vært grei i forhold til at kurset gir 7,5 studiepoeng da?
00:51:17 Lærer
Ja, jeg synes det. Også synes jeg at det var veldig greit at det var litt fleksibelt med innlevering også, forhold til... Jeg har prøvd å være i forkant hele tiden, siden jeg har såpass mye utenom. Ja, både 100% jobb, og vi er fosterforeldre og vi er liksom sånn... Det mer ting

som skjer enn bare skolen og studiene også. Derfor synes jeg på en måte at det har vært greit at jeg har fått jobba i mitt tempo og fått tatt det på den tida der da det passer meg. Jeg tror jeg bare har fått med meg to direkte på en måte møter i hele... For det passer aldri dessverre på torsdagene. Det er liksom bare tilfeldigvis at på hver tredje torsdag så kan det passe. Så spørs det om den torsdagen er den da torsdagen det er noe møte på.
00:52:02 Intervjuer
Ja.
00:52:06 Lærer
Ja. Men veldig greit at det ligger på nett, så jeg kan se gjennom. Slack er veldig greit å bruke til å spørre når du... Og når du har den, så har du elevassistenten som på en måte du kan spørre om når det er noen ting du lurer på. Så jeg synes på en måte oppbyggingen har vært helt grei. Jeg kunne kanskje tenkt meg litt mer sånn... Siden altså, jeg sa det på en måte at jeg har funnet fram rasperry'en på en måte som lå i en pappeske at det lå litt mer... Sånn som det lå i høst, så da det lå litt sånne introduksjoner til de forskjellige konseptene med lister og løkker, og litt sånne ting. At det kanskje kunne ligge noe også innenfor for eksempel micro:bit og rasperperry'en, arduinoen og... Litt sånn små videoer som kunne ha gitt litt starthjelp på en måte, til komme i gang.
00:53:02 Intervjuer
Ja, en sånn introduksjon til hvordan man bruker en raspberry pi for eksempel?
00:53:02 Lærer
Ja. Rett og slett alle ting er på en måte foreslått nå, det er jo er jo grunnen til at det er en terskel å ta i bruk, for jeg vet ikke jeg opp og ned og fram og bak eller noen ting.
00:53:16 Intervjuer
Ja, da kan det jo være litt vanskelig å begynne å programmere da, hvis man ikke vet...
00:53:21 Lærer
Ja, når du ikke vet hvordan det fungerer. Og der også må jeg på en måte søket på nett, ikke sant. Og da tenker jeg kanskje at disse NTNU-kursene kunne ha laget en introduksjon for oss på disse her praktiske tingene. Sånn at vi lettere kom i gang med det som vi har lyst å holde på med programmering og sånt.
00:53:45 Intervju
Ja, sånn at terskelen blir lavere for å faktisk få programmert, fordi man ikke bruker så mye tid på å sette opp?
00:53:52 Lærer
Ja.
00:53:56 Intervjuer
Tror du at det hadde vært nødvendig eller nyttig med enda flere kurs som bygger videre på de 2 første kursene?
00:54:06 Lærer
Ja, da tenker jeg kanskje at den... Hvis det skal være anvendt, at vi følger opp det vi startet opp med noe på en måte... At hvis du har et sånn minikurs om rasperry, minikurs om microbit, om arduino, og disse her forskjellige elektronikk bitene som finnes som vi tar i bruk i skolen. Om en kunne ha på en måte gitt en egen opplæring til det da. For det er noe med at for oss lærer også, å se hva vi kan bruke det til. Og der tenker jeg også på en måte, at med en gang du begynner å se nytten med noe. Da er det mye kjekkere å komme i gang med det også. Sånn «Åja, man kan bruker det til det ja», ikke sant. Som regel så er lærere kreative, men vi må på en måte vite hva vi kan bruke ting til, før på en måte setter i gang også da. Selv om tida er knapp og sånn så.

00:55:01 Intervjuer
Ja. Så kanskje det hadde vært nyttig med et enda et kurs da, som viser enda mer mulighetene innenfor sånn microbit og rasperry pi og... Sånn at man skiller det ut fra det litt mer grunnleggende? Eller bygge videre på da?
00:55:21 Lærer
Oi, det var plutselig helt stille nå.
00:55:24 Intervjuer
Og ja, beklager.
00:55:28 Lærer
Ja, bildet ditt ble hengende.
00:55:31 Intervjuer
Ja, det jeg holdt på å si, når internettet mitt koblet litt ut her, var at det hadde kanskje vært nyttig med enda et kurs da, som viser mulighetene videre til å bruke micro:bit, rasperry pi og sånt, utover det grunnleggende man fikk gjennom anvendt programmering?
00:55:32 Lærer
Ja, eller om anvendt programmering på en måte har tatt opp disse der introduksjonen og nytteverdiene. Altså jeg vet ikke... Ikke sant, nå fikk vi jo mange valg. Vi fikk veldig mange valg som vi på en måte kunne plukke blant og sånne ting, så jeg vet ikke om det er for mye å legge det inn i det kurset som allerede er. Men slike introduksjoner til de forskjellige delene en kan velge. For jeg så jo også det, ikke sant, det ble jo lagt ut tidlig før vi startet kurset om hva vi kunne velge i. Men det er veldig vanskelig å velge når vi ikke vet hva de konseptet faktisk inneholder. Altså hva du kan gjøre med de konseptene. Som du egentlig ble bedt om å... Du kan kjøpe en sånn, og du kan kjøpe sånn. Jeg så at veldig mange spurte akkurat om det er på slacken. På en måte, hva har dere på skolen, og hva har dere på skolen? Men enda, så vet vi jo ikke hva vi kan bruke det til. Så det er jo på en måte en dings som vi lurer på hva vi kan bruke den til. Så jeg vet jeg selv lagde til noen sånne... Vi har selv valgt emne i naturfag, så lager jeg til noen sånne inspirasjonsfilmer dem. For i ulike emner for på en måte for å vise hva de kunne velge blant. Og da handler det på en måte om litt ideer på en måte innenfor den og den og den og den. Og ja, «dette liker jeg», ikke sant. Så hvis du da på en måte lager noe sånt i forkant av å velge, sant, at det på en måte «Rasperry'en kan brukes til sånn, micro:biten kan bruke sånn, og arduiono sånn», så vet vi liksom hva vi velger i.
00:57:29 Intervjuer
Ja, det er jo ikke så lett å velge når du ikke vet hva det er.
00:57:30 Lærer
Nei, det er ikke det. Og spesielt når en begynner å snakke om å kjøpe inn. Du vet nå økonomien på videregående. Det er ikke bare å kjøpe noe som er... Altså, kanskje ikke blir brukt til noe annet enn at du skal ta et fag på NTNU liksom. Og at det ikke blir tatt i bruk senere, så det... Det er veldig viktig at det er nytteverdi i det som blir handlet inn også.
00:57:55 Intervjuer
Ja, det er nytteverdien som er nøkkelen i det meste som har med programmering å gjøre, høres det ut som?
00:58:01 Lærer
Ja, og da ser også elevene den der nytteverdien, og har lyst til å fortsette, ja.
00:58:10 Intervjuer
Hvis man tar et steg tilbake, og ser på å lære programmering sånn generelt. Hva tror du er den beste måten å lære programmering på for lærere som er i full jobb?

00:58:23 Lærer
Nå gjelder det meg?
00:58:25 Intervjuer
Ja.
00:58:29 Lærer
Tid en stor faktor her. For jeg synes på en måte... Altså, når jeg har sånn en halvtime-time til rådighet om ettermiddagen... Da ble det litt sånn stykkevis og delt. Men når jeg på en måte fikk satt meg ned noen timer og holde på, og så komme meg videre, og får ting til å fungere, alt sammen i samme økta. Da ga det mer motivasjon til å prøve nye ting. Det er merket jeg selv også da. Og det er jo egentlig det vi setter elevene med også, på en måte, at de får bare drypp og drypp. Da kommer aldri til den der oi. For da må vi pakke sammen. For det er liksom sånn at da må vi faktisk pakke sammen. Så det tidsaspektet tror jeg er veldig nyttig. Også tror jeg det er nyttig på en måte å ha noen å sparre med. Ha noen å spørre. Altså, hele tiden når du... Nå brukte jeg litt slack, men jeg brukte også dattera mi underveis, når en på en måte stopper opp. For det å ha noen som på en måte faktisk klarer å svare deg... Jeg er jo det selv med mine elever. Jeg svarer dem utenom skoletid. Det er ikke alle som har lyst å gjøre det, men på en måte det når du sitter og jobber med noen ting, og sitter fast – så er det veldig frustrerende å måtte vente til neste dag med å få svar på det du lurer på. Ja.
01:00:00 Intervjuer
Nesten vanskelig å legge seg på kvelden, fordi du blir bare liggende og tenke på det du ikke har fått til.
01:00:04 Lærer
Ja, eller så gir du opp. Ja, det gidder jeg ikke, vi utsetter det til en annen dag. Så det å på en måte ha noen å sparre med. Det er viktig. Så den der kanalen på slack, den har jeg likt veldig godt. Og så har jeg likt veldig datteren min også i vinter *ler*. Det har vært på en måte en sånn, når alt sammen stopper opp, og jeg skjønner ikke hva jeg gjør feil, og liksom sånn for å få det til så... Og det tenker jeg også er viktig for elever da, som skal lære noe, ikke sant. At de har noen å sparre med, og det har ikke videregående elever på samme måte.
01:00:46 Intervjuer
Nei.
01:00:46 Lærer
Og heller kanskje ikke så mange jevnaldrende som kan så mye enda, sånn at det... Og det er en terskel for å spørre lærerne også. Det er det. Uavhengig av hvor åpne vi er, så er det ofte en terskel.
01:01:04 Intervjuer
Ja så den beste måten å lære programmering på, det er liksom å få tid til den her mengdetreningen og litt interaksjon?
01:01:12 Lærer
Ja. Bruke tid, og å ha noen å sparre med på en måte, hele veien. På en måte det som heter scaffolding, heter det ikke det? Det å ha noen til å på en måte følge opp hele veien. Som på en måte dytter deg framover når det på en måte det stopper opp for deg da.
01:01:28 Intervjuer
Ja. Når du har undervist i programmering, har du noen gang vært i en situasjon der du har måttet lære deg noe nytt, eller nye programmeringskonsepter sammen med elevene dine?
01:01:50 Lærer
Ja. I høst ja, da jeg holdt på med 1P. Selv om det var på enkle ting, så var det liksom som å... Jeg måtte ... Før jeg hadde noe undervisning, så måtte jeg være sikker på at jeg kunne det

selv. Også møter du nye utfordringer, ikke sant. I det elevene kjøre program, og ikke får det til å kjøre, sant. Så jeg blir flinkere og flinkere til å lese feilkoder jeg også på en måte. Etter hvert som du... Så det å lese feilkoder, en lærer jo ikke det før det kommer en feil, sant.
01:02:21 Intervjuer
Nei, det er sant.
01:02:22 Lærer
Ikke sant, når du sitter og programmer enkle ting her hjemme, og du får det til, så er det ikke utfordringer. Men det er ikke før feilkodene kommer blant elevene at du blir god på det da.
01:02:35 Intervjuer
Ja. Hvordan synes du det har påvirket din rolle som lærer, når du må sitte sammen, altså det samspillet med elevene for å finne ut av feilmeldinger og lære noe nytt?
01:02:46 Lærer
Ja, men det gjør vi hele tiden. Det syns jeg bare kjekt. Ja da. Det er sånn det er det. Og så er det sånn når det er en oppgave en elev på en måte spør om, og er på en måte plutselig «Nei, nå må vi se på at, nei...» Så blir de så irritert, vet du *ler*. Så ja, men nå lærer vi. *ler***.** Jeg pleier å si til elever som blir irritert når jeg ikke gir dem hele svaret også sånn – «Ja, men nå lærer vi.» Nei, nå er hjernecellene i gang. Så nei, vi må ikke bli motløse fordi vi ikke får det til i første omgang
01:03:24 Intervjuer
Jeg tenker at det er jo sikkert litt interessant for elevene også. For ofte så er det jo sånn at læreren er jo den som sitter med svaret på det meste, og så plutselig så er man i en situasjon der man faktisk må samarbeide om å finne ut av programmeringsfeil, og...
01:03:39 Lærer
Ja da. Men det er noe med at på en måte det er da vi må følge opp, ikke sant. Ha tid til å følge opp, og på en måte for dem å hvert fall ikke sitte igjen med et program som aldri fungerer. Altså på en måte, hvis vi da ikke får det til i klasserommet, så må vi komme tilbake igjen, og hjelpe de på en måte, må følge opp. Fordi hvis ikke, så mister de motivasjonen. Det er bare... Ja.
01:04:07 Intervjuer
Ja. Da har jeg et siste spørsmål, og det går på at å lære programmering tar lang tid. Og den læringsprosessen med programmering, den går jo langt forbi de NTNU-kursene som vi drifter. Men hvordan kan den her lange, kontinuerlig læringsprosessen støttes, for eksempel gjennom kurs, seminarer eller læringsmateriale? Når du er ferdig med NTNU-kurset, hva kan vi bidra med for at den videre læringsprosessen din innenfor programmering støttes?
01:04:46 Lærer
At jeg skal utvikle meg videre, eller at den skal vedlikeholdes? Hva er det?
01:04:52 Intervjuer
En kombinasjon kanskje. Det kommer jo nye ting hele tiden, ikke sant. Så som de to kursene som vi tilbyr... Vi tror jo ikke at dere blir utlært *ler*.
01:05:09 Lærer
Nei. *ler*. For min egen del, så hadde i hvert fall et oppslagsverk på norsk vært til god hjelp ja, for å på en måte hvert fall vedlikeholde. Fordi at det er jo en del konsepter som jeg kan sånn delvis selv. Jeg føler selv også, på en måte både det med lister, tupler og alt det andre som skal plasseres. Og så tenker jeg også det at det som vi kommer til å bruke noe i klasserommet, det kommer vi til å bli gode på. Mens det som på en måte... Disse to ordskyen mellom elev og lærer, så ser du på en måte hvor langt elevene er kommet, og det spørs hvor langt videre de kommer også. Hvis det blir en så liten del av undervisningen vår

som skal inneholde programmer. Og da vil det nok dabbe av med den avanserte kunnskapen som vi nå har lært oss. Så at i hvert fall om det da kunne vært noe oppslagsverk, i forhold til... Så vi på en måte ser hva vi kan ta i bruk. Det tror jeg hadde vært nyttig. For at vi kommer kanskje ikke til å ta i bruk så veldig masse blant elevene våre. Jeg ser, altså... Neste år, så får jeg ny førsteklasse, ikke sant. Som har også hatt gamle læreplaner på ungdomsskolen, så de kommer sikkert med like dårlig bakgrunnskunnskap, så vi starter på scratch igjen neste år. *ler*
01:06:42 Intervjuer
Ja, det blir sikkert sånn i noen år ja. *ler*
01:06:46 Lærer
Ja, det blir ikke lister og tupler og funksjoner sånn med det første. De kan ikke matematikken til den gang, så... Så får vi se da, hvor mye programmering det blir inn i S1 og S2 etter hvert. Om de klarer å få lagt inn noe der, for da kan det kanskje gå videre med noe der da, i hvert fall.
01:07:09 Intervjuer
Ja. Ja, da tror jeg at jeg har fått svar på alle mine spørsmål. Eller jeg har fått svar på alle mine spørsmål! Men har du noe annet kommentarer eller refleksjoner om ting som jeg ikke har spurt om?
01:07:26 Lærer
Nei, nei, jeg tenker ikke det. Jeg har fått med det meste.
01:07:31 Intervjuer
Ja. Ja, men da bare for meg å si tusen takk for at du stilte opp. Det har vært veldig interessant å høre på det du har å si. Det er veldig spennende å høre på en måte rett fra kilden hva som oppleves som utfordrende. Så det gir veldig nyttig informasjon i det forskningsprosjektet, så tusen takk for det.
01:07:51 Lærer
Ja, men det er bra.
01:07:53 Intervjuer
Så får du ha en fin dag videre.
01:07:56 Lærer
Takk for det, ha det godt.
01:07:58 Intervjuer
Ha det bra.

E.4 Codes used in P8

Codes used in P8

Frequency	Name
96	Learning Prog - Teacher
60	Challenges learning prog
2	Connecting theory and practice
4	creating a mental model
5	Debugging
4	Functions and Libraries
2	if-statements
5	Lists and structured datatypes (tupler)
5	Loops
5	New way of thinking
5	Other concept or constructs
5	Practicing enough
3	Steep Learning Curve
10	writing code and pseudocode
35	Learning a new language
12	Learning new language is easy
2	Mixing Syntax
5	New Learning
15	Transferability
89	Learning prog - Pupils
10	Challenging for pupils
3	Connecting theory and practice
4	creating a mental model
8	Debugging
5	Functions and Libraries
4	if-statements
3	Lists and structured datatypes (tupler)
12	Loops
2	Mixing concepts
9	New way of thinking
3	Other concept or constructs
2	Time
2	Transition from block-based
7	Variables
10	writing code and pseudocode

5	Teaching Programming
11	General academic skills
9	Interest and effort
11	Pre-existing knowledge
3	vunerability of pupils
4	Applied programming
2	Collaboration
3	Controlling physical objects
8	Other approaches to motivate
13	Pupils general motivation
2	Teaching language Constructs
5	Logic of the language
8	Adapted education
3	Block-based programming
9	Creating good teaching plans
8	Creating Learning resources
6	Guidance from teacher
8	Lack of time
3	Language Learning
3	Learning by doing
8	Practicing enough
2	Predicted advantages
9	Progression in learning
15	Relation to other subjects
7	Teachers knowledge
3	Teaching in school is challenging
2	teaching programming concepts
5	Concepts and complexity
4	Logic
34	Perceived learning
2	Negative own learning
8	Positive own learning
12	Pre-knowledge
8	Programming is challenging
5	Programming is time consuming
32	Course specific
7	Assignments

2	Course was time consuming
6	negative feedback
2	Participation in learning activities
3	Pensum relevance
3	positive feedback
5	Progression
4	Suitability for beginners
6	Learning Resources
4	Community and collaboration
9	Information Sources
8	Attitude towards future learning
3	Motivated to learn more
3	Positive towards programming in schools

E.5 Data for calculating quantitative figures

Statements

STMT 1	Programming has not been an easy subject for me to learn
STMT 2	Programming requires proper understanding of abstract concepts (eg functions / methods, etc.)
STMT 3	I have learning difficulties due to the nature of the subject
STMT 4	I think I will have challenges in designing lessons so that they will be beneficial to my students.
STMT 5	I think it will be exciting to teach programming courses.
STMT 6	I knew little or no programming BEFORE starting the course
STMT 7	I feel like I've learned enough programming now to be able to teach my students.
STMT 8	Finding bugs in programs is
STMT 9	Dividing a problem into sub-problems (program functions) is
STMT 10	Using libraries in programs is
STMT 11	Understanding variable initialization is
STMT 12	Understanding loops is
STMT 13	Understanding how loops should be used is
STMT 14	Understanding a select-statement is
STMT 15	Understanding how to use a select-statement is
STMT 16	Creating a program to solve a specific problem is

Scale

		STMT 1-7	STMT 8-16
1	Helt Uenig	Completely Disagree	Very Difficault
2	Uenig	Disagree	Difficault
3	Hverken eller	Neutral	Neutral

4	Enig	Agree	Easy
5	Helt Enig	Completely Agree	Very Easy

Responses

	STMT 1	STMT 2	STMT 3	STMT 4	STMT 5	STMT 6	STMT 7	STMT 8	STMT 9	STMT 10	STMT 11	STMT 12	STMT 13	STMT 14	STMT 15	STMT 16	Invalid
10001	4	4	2	2	4	4	3	3	2	3	3	2	2	2	2	2	0
10002																	0
10003	2	5	1	1	4	1	4	3	4	4	5	5	5	5	5	3	0
10004	2	5	1	2	5	2	4	2	3	3	4	4	2	4	3	3	0
10005	4	5	1	2	4	4	4	3	3	3	4	4	3	3	3	3	0
10006	2	5	1	4	4	2	4	2	3	4	4	2	2	3	3	1	0
10007	4	5	3	4	4	4	2	2	3	2	2	4	2	4	2	2	0
10008	3	4	2	1	4	4	4	2	2	2	4	2	2	4	4	3	0
10009	3	5	1	3	5	5	4	3	3	3	5	5	4	5	4	4	0
10010	1	5	1	4	3	4	2	2	3	2	4	4	4	4	4	3	0
10011	5	4	2	4	3	5	1	1	1	2	1	3	3	1	1	1	0
10012	3	4	2	2	5	4	3	2	4	3	4	4	4	4	4	2	0
10013	2	4	1	4	4	3	4	2	1	4	5	5	5	5	5	4	0
10014	2	4	2	2	5	4	4	3	4	4	4	4	4	4	4	3	0
10015																	0
10016	3	5	2	3	5	4	3	2	2	2	4	4	2	4	3	3	0
10017	2	4	3	4	4	5	3	4	3	4	4	3	2	3	2	3	0
10018	3	5	3	4	4	5	4	2	2	2	3	2	1	2	2	2	0
10019	1	5	1	2	5	2	5	3	4	4	5	5	5	5	5	4	0
10020	4	5	2	3	4	4	3	2	2	2	4	3	2	4	4	2	0
10021	2	5	1	2	4	1	4	3	4	3	5	5	4	4	4	3	0
10022	5	5	3	4	5	5	3	1	1	1	2	2	2	2	2	1	0

10023	2	4	1	3	4	1	4	4	4	4	4	4	4	4	4	4	0
10024	1	2	1	2	5	3	4	4	4	3	4	4	4	4	3	4	0
10025	1	4	1	1	5	2	4	4	4	5	5	5	5	5	5	3	0
10026	5	4	1	4	3	4	1	2	2	2	3	4	4	4	3	2	0
10027	2	4	1	3	4	2	2	1	2	3	4	4	2	4	2	3	0
10028	3	4	1	1	5	2	3	3	2	4	4	4	4	4	4	4	0
10029	2	4	1	1	5	4	4	4	4	5	5	4	2	5	4	3	0
10030	4	5	2	4	3	4	4	2	3	2	1	4	3	4	4	2	0
10031	4	4	1	3	4	4	2	Invalid	2	1	2	4	3	2	2	3	1
10032	1	4	1	3	4	1	4	3	4	3	5	5	5	5	5	3	0
10033	2	4	1	2	5	2	4	4	4	4	5	5	4	5	4	4	0
10034																	0
10035	2	4	2	3	4	2	3	3	2	4	4	5	4	5	4	3	0
10036	2	4	2	2	4	4	2	2	2	4	4	4	2	4	4	3	0
10037																	0
10038	3	5	2	3	4	1	4	2	3	5	5	4	3	3	3	3	0
10039	3	4	1	2	5	1	2	2	4	3	5	5	5	5	5	5	0
10040	3	5	2	2	5	2	4	4	2	3	5	4	3	4	4	3	0
10041	4	5	2	2	5	2	4	3	3	Invalid	4	4	4	4	4	3	1
10042	4	5	2	4	4	5	1	2	2	Invalid	Invalid	Invalid	2	Invalid	2	2	4
10043	2	4	1	2	4	3	3	2	4	3	5	3	3	4	4	3	0
10044																	0
10045	3	4	2	Invalid	5	2	4	2	3	4	4	4	2	4	4	2	1
10046	2	4	3	4	4	5	2	2	2	2	4	4	2	2	2	4	0
10047	3	4	2	3	3	4	2	3	3	2	4	3	3	4	4	3	0
10048	3	4	2	2	4	5	2	4	3	4	4	3	3	4	4	3	0
10049	3	4	2	3	3	5	3	2	4	2	4	4	2	4	2	2	0
10050	2	4	2	2	4	2	2	4	3	4	4	4	2	3	3	3	0
10051	3	4	2	3	5	5	2	2	4	3	4	4	3	3	3	3	0

10052	3	4	3	3	5	3	2	2	3	4	4	4	4	4	4	3	0
10053	3	4	1	2	4	5	3	2	1	3	3	3	2	2	2	2	0
10054	3	4	2	2	5	5	5	3	Invalid	4	4	3	3	4	4	3	1
10055	3	4	1	2	4	2	4	3	4	4	4	5	4	4	4	3	0
10056	2	4	1	2	5	2	4	2	4	4	Invalid	5	4	4	4	4	1
10057	2	5	2	3	4	5	2	2	3	4	5	5	3	5	5	2	0
10058	3	4	2	2	5	2	4	3	3	4	4	2	2	3	3	3	0
10059	4	5	4	4	5	5	3	1	2	2	2	2	2	2	2	2	0
10060	4	4	2	4	4	4	3	4	3	2	2	2	1	2	1	1	0
10061	3	4	3	4	4	4	3	2	2	3	2	3	3	2	2	2	0
10062	2	4	3	4	3	2	3	2	Invalid	2	4	2	2	4	3	2	1
10063	2	5	1	2	4	3	5	2	2	4	3	3	3	4	4	2	0
10064	4	4	4	4	4	5	2	1	2	4	3	3	3	3	3	2	0
10065	2	4	2	2	3	2	4	1	1	2	4	4	2	4	4	4	0
10066																	0
10067																	0
10068	1	5	1	2	4	1	5	4	5	4	5	5	5	5	5	4	0
10069	2	4	1	2	4	5	Invalid	4	3	2	5	4	4	4	4	2	1
10070	2	4	1	Invalid	Invalid	2	4	2	4	4	Invalid	4	Invalid	4	2	3	4
10071	3	4	2	3	4	2	2	2	3	2	4	4	4	3	3	2	0
10072	4	4	3	3	4	5	3	2	2	3	3	2	2	2	2	2	0
10073	3	4	3	2	4	2	2	3	4	2	5	4	4	4	4	2	0
10074	1	2	1	1	5	2	5	4	4	5	5	4	4	5	5	Invalid	1
10075	3	4	1	4	3	2	2	3	3	2	4	4	4	4	4	3	0
10076	4	4	2	5	3	5	2	2	1	3	2	3	2	3	2	2	0
10077	2	3	2	2	4	4	3	3	4	3	4	4	3	4	4	3	0
10078	2	2	2	Invalid	5	1	4	4	3	1	5	5	5	5	5	4	1
10079	4	4	1	4	5	4	4	3	4	4	4	4	3	3	3	3	0
10080	4	5	1	3	4	5	3	2	2	Invalid	3	4	4	4	3	3	1

10081	1	4	1	4	3	1	4	3	4	4	4	5	5	5	5	4	0
10082	3	4	2	2	5	4	3	2	1	2	4	4	3	4	3	2	0
10083																	0
10084	1	5	1	1	5	1	5	5	5	5	5	5	5	5	5	5	0
10085	2	5	1	2	4	2	4	2	4	4	5	5	Invalid	4	4	3	1
10086	3	5	2	2	4	4	4	3	3	4	4	5	5	5	5	4	0
10087	1	4	1	2	5	3	4	2	3	3	4	4	4	4	4	3	0
10088	3	5	2	2	4	5	3	2	3	4	3	4	4	4	4	3	0
10089	2	5	1	3	5	2	4	3	Invalid	5	5	5	5	5	4	3	1
10090	4	4	2	3	4	5	2	2	2	2	4	4	2	4	2	2	0
10091	3	4	2	2	4	5	4	2	2	3	4	4	2	4	3	2	0
10092	3	5	3	4	4	5	4	2	1	2	4	4	4	3	3	2	0
10093	4	5	2	2	5	5	4	4	4	4	4	4	2	4	4	2	0
10094	3	5	1	4	4	4	3	2	Invalid	3	5	4	2	4	2	2	1
10095	4	4	1	2	4	2	5	3	3	2	5	5	5	5	5	3	0
10096	2	4	1	2	4	2	3	3	4	4	5	5	4	5	4	4	0
10097	4	4	2	4	4	4	2	1	2	1	2	2	2	2	2	2	0
10098	4	5	3	4	3	5	2	3	2	2	5	4	3	4	4	2	0
10099	2	4	2	3	4	1	3	4	3	4	4	4	4	4	4	4	0
10100	4	5	4	3	3	4	2	2	2	2	3	3	3	2	2	2	0
10101	3	4	2	2	4	5	1	2	3	4	5	2	2	4	4	3	0
10102	2	4	1	2	4	1	5	5	5	5	5	5	4	5	5	4	0
10103	2	4	1	2	5	5	3	2	4	2	4	4	4	4	4	2	0
10104	2	5	1	1	3	2	2	2	2	3	4	4	4	4	4	3	0
10105	4	5	4	4	2	5	3	1	2	3	4	3	2	3	2	2	0
10106																	0
10107																	0
10108	4	5	4	5	3	5	3	1	2	2	4	4	4	4	4	2	0
10109																	0

10110	4	4	2	5	3	3	2	3	2	4	4	4	2	4	2	2	0
10111	3	5	2	2	1	5	3	2	2	2	3	2	3	2	2	2	0
10112	4	4	3	2	4	5	3	2	2	4	4	2	2	Invalid	3	Invalid	2
10113	3	4	1	4	4	5	4	2	3	3	5	4	3	3	3	2	0
10114	1	4	2	3	4	5	1	2	3	5	5	2	2	5	5	4	0
10115	2	4	1	3	5	2	4	2	2	4	4	4	4	4	4	3	0
10116	4	4	4	4	4	4	2	4	1	2	4	4	3	4	4	2	0
10117	2	3	1	3	4	3	4	4	4	4	5	4	3	4	4	4	0
10118	1	5	2	3	5	1	2	3	4	5	5	5	4	5	5	3	0
10119	3	4	2	3	4	5	5	3	1	2	4	4	4	4	4	2	0
10120	4	5	1	2	3	5	2	2	4	4	5	4	3	5	4	3	0
10121	1	4	1	4	5	1	5	2	2	4	5	4	2	5	4	2	0
10122	3	4	Invalid	4	3	5	2	2	2	2	2	2	2	2	2	2	1
10123	4	5	2	3	4	5	3	2	2	2	4	5	2	5	2	Invalid	1
10124	2	4	1	2	4	5	2	2	3	4	4	5	4	4	4	3	0
10125	1	Invalid	1	1	4	1	4	4	3	Invalid	5	5	5	5	5	4	2
10126	3	4	3	4	4	4	3	2	2	2	3	3	3	2	2	1	0
10127	3	5	3	1	5	3	4	1	2	3	Invalid	4	2	3	2	2	1
10128	1	4	1	2	4	1	4	3	4	2	5	5	5	5	5	4	0
10129	2	4	1	1	4	2	1	2	3	2	4	4	4	4	4	3	0
10130	4	4	2	2	4	4	4	4	4	4	4	4	4	4	4	3	0
10131	2	4	1	2	5	3	4	2	3	5	4	5	5	5	5	Invalid	1
10132	2	4	1	2	4	5	4	2	4	4	5	5	4	5	4	Invalid	1
10133																	0
10134																	0
10135	5	5	3	2	4	5	1	2	2	4	2	2	2	2	2	1	0
10136	4	4	1	1	Invalid	4	3	2	3	2	2	3	3	2	2	1	1
10137	5	5	3	4	4	3	1	1	1	1	1	2	1	1	1	1	0
10138	1	5	1	1	5	1	3	4	3	5	5	5	5	5	5	5	0

10139																	0
10140	4	4	3	4	4	5	2	2	2	2	3	4	4	2	2	1	0
10141	4	5	2	1	5	4	3	2	Invalid	4	5	5	2	5	4	2	1
10142	3	4	2	2	4	5	5	1	2	3	4	4	4	4	4	2	0
10143	4	5	1	2	5	5	5	1	2	3	4	3	1	3	1	1	0
10144	4	5	3	2	4	4	2	2	2	2	3	2	2	2	2	2	0
10145	2	4	1	2	4	2	4	3	3	5	5	5	5	5	5	3	0
10146	4	5	3	2	3	3	2	2	3	3	3	4	2	4	2	1	0
10147	1	5	2	3	5	2	2	3	3	3	4	3	3	4	3	3	0
10148	2	4	1	3	4	5	4	4	2	2	5	5	2	5	4	3	0
10149	4	4	3	2	5	5	3	1	2	2	3	1	2	2	2	2	0
10150	4	4	3	4	4	5	2	1	1	4	3	4	2	2	2	2	0
10151	3	5	3	4	4	5	2	2	2	2	4	3	2	1	1	2	0
10152	3	4	1	2	5	5	3	3	Invalid	3	2	4	4	3	3	2	1
10153	3	5	2	2	4	5	4	2	3	4	4	4	2	4	2	2	0
10154	2	4	2	3	4	4	4	4	2	3	4	4	4	3	4	2	0
10155	3	4	1	1	4	3	4	3	3	4	4	4	2	4	2	3	0
10156	5	5	3	4	4	4	2	2	4	2	4	4	4	4	3	2	0
10157	4	4	2	2	4	2	4	1	4	2	4	5	4	4	4	2	0
10158																	0
10159	2	4	1	Invalid	4	4	Invalid	2	4	2	4	Invalid	Invalid	4	4	3	4
10160	2	5	1	2	5	3	5	3	5	4	4	5	Invalid	4	4	2	1
10161	2	4	1	2	4	3	4	4	4	4	5	5	5	5	5	4	0
10162	4	4	2	4	3	5	4	2	2	3	4	4	2	3	3	3	0
10163	Invalid	4	2	4	4	5	4	2	2	2	2	2	2	2	2	2	1
10164	4	4	1	4	3	3	2	3	4	2	4	4	3	4	4	3	0
10165	2	4	1	1	5	3	4	2	2	4	5	5	4	3	3	4	0
10166	2	4	1	2	5	1	4	4	4	5	5	5	4	4	4	2	0
10167	2	3	1	1	5	4	3	4	4	4	5	5	5	5	5	4	0

10168	1	4	1	1	5	1	5	4	5	4	5	5	5	5	5	5	0
10169	5	3	5	4	3	5	3	1	2	3	2	2	1	2	1	3	0
10170																	0
10171	3	5	3	2	5	4	Invalid	1	3	3	2	5	2	1	1	2	1
10172	2	5	1	1	5	3	4	4	2	3	4	5	4	4	3	3	0
10173	3	4	2	2	4	5	4	3	2	3	3	3	3	3	3	3	0
10174	2	4	1	2	5	3	4	2	2	3	4	4	4	5	5	2	0
10175	4	4	2	2	4	5	4	2	3	3	4	4	4	3	3	3	0
10176	1	3	1	2	5	3	3	3	4	4	5	4	4	4	4	3	0
10177																	0
10178	2	4	2	2	5	4	4	3	4	4	4	4	4	4	4	3	0
Invalid	1	1	1	4	2	0	3	1	6	4	4	2	4	2	0	5	40

ISBN 978-82-326-7846-4 (printed ver.)
ISBN 978-82-326-7845-7 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU

Norwegian University of
Science and Technology