Master's thesis

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

Vegard Hellem

The Effect of Mandatory Assignments on Students' Learning Outcome and Motivation in Introductory Programming Courses

Master's thesis in Computer Science
Supervisor: Guttorm Sindre, Co-supervisor: Madeleine Loraas
May 2019

NTNU
Norwegian University of Science and Technology

Vegard Hellem

The Effect of Mandatory Assignments on Students' Learning Outcome and Motivation in Introductory Programming Courses

Master's thesis in Computer Science
Supervisor: Guttorm Sindre, Co-supervisor: Madeleine Loraas
May 2019

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

NTNU
Norwegian University of
Science and Technology

Abstract

In a world in high demand of computer scientists, computer science education should be effective and quality conscious. A better understanding of what type of activities that are best suited for improving students' learning could enable further improvements for students taking computer science courses.

In this thesis, the effect of mandatory assignments on students' learning outcome and motivation in introductory programming courses is explored through a research project on students taking the course TDT4100 Object-oriented programming at NTNU. The experiment involved a quasi-experimental research method, with students volunteering into one of two groups, the experimental and the control group. The control group followed a weekly set of assignments that have been the norm for the course the last years, while the experimental group had biweekly sessions with a teaching assistant, monitoring their accomplishment of the learning goals of the course.

The results were measured using a pretest and a posttest given to the students. The results indicated there was no statistically significant difference in neither learning outcome nor motivation between the two groups. The results indicated that the experimental group achieved the same learning outcome while spending fewer hours per week on the course compared to the control group.

The results of this thesis can be used as a starting point for further research into creating the best computer science education. Students have individual learning styles and learn to program in different ways. Therefore, computer science education should tailor to students' requirements, offering them a way of learning that is best suited for their learning style. Further research into what type of learning activity that contributes to the best learning environment for students is needed to create tomorrows computer science education.

Sammendrag

I en verden med høy etterspørsel etter teknologer, bør datateknologi og informatikk-utdanning være effektiv og kvalitetsbevisst. En bedre forståelse av hva slags aktiviter som passer best for å forbedre studentenes læring kan muliggjøre ytterligere forbedringer for studenter som tar kurs i datateknologi.

I denne oppgaven blir effekten av obligatoriske oppgaver på studentens læringsutbytte og motivasjon i introkurs til programmering utforsket gjennom et forskningsprosjekt på studenter som tar kurset TDT4100 Objektorientert programmering ved NTNU. Forsøket involverte en kvasi-eksperimentell forskningsmetode, med studenter som frivillig deltok en av to grupper, den eksperimentelle og kontrollgruppen. Kontrollgruppen fulgte et ukentlig sett med oppgaver som har vært normen for kurset de siste årene, mens eksperimentgruppen hadde økter annenhver uke med en studentassistent. Hensikten med møtet var å gå gjennom oppnåelsen av kursets læringsmål og hva de hadde gjort i forhold til det.

Resultatene ble målt ved hjelp av en pretest og en posttest gitt til studentene. Resultatene indikerte at det ikke var noen statistisk signifikant forskjell i verken læringsutbytte eller motivasjon mellom de to gruppene. Resultatene indikerte at eksperimentell gruppe oppnådde samme læringsutbytte mens de brukte færre timer per uke på kurset sammenlignet med kontrollgruppen.

Resultatene av denne oppgaven kan brukes som utgangspunkt for videre forskning i å skape den beste informatikkutdanningen. Studenter har forskjellige læringsstiler og lærer å programmere på ulike måter. Derfor bør informatikkutdanning tilpasse seg studenters behov, og gi dem en måte å lære som passer best for deres læringsstil. Videre forskning på hvilken type læringsaktivitet som gir den beste veiledningen for studenter med ulik læringsstil, er nødvendig for å skape morgendagens informatikkutdanning.

Acknowledgements

I want to extend my gratitude to my supervisor, Guttorm Sindre, for valuable insight and feedback in the process of writing this thesis. His insightful commentary and knowledge of previous work have been very helpful for the writing of this thesis.

I also want to extend my gratitude to my co-supervisor Madeleine Loraas. Her weekly feedback has been invaluable during the writing of this thesis, as well as her knowledge of pedagogy and methodology for designing the experiment.

I would also like to thanks all the students in TDT4100 who contributed to this project, both in the experimental and the control group. Without them, this thesis would never have had happened, and their commentaries and work have given me very insightful data for use in this thesis and the future.

A big thanks also go out to Abakus, my student association, for a constant supply of coffee at their office, as well as excellent opportunities for procrastination and good conversation.

Contents

List of Tables

List of Figures

Abbreviations

NGD	=	Non-Equivalent Groups Design
STD	=	Self-Determination Theory
ANCOVA	=	Analysis of Covariance
CS	=	Computer Science
US	=	United States
NOKUT	=	Norwegian Agency for Quality Assurance in Education
NTNU	=	Norwegian University of Science and Technology
IT	=	Information Technology

Chapter 1

Introduction

The following chapter gives a brief overview of the problem that this thesis aims to help solve. It describes the motivation behind the thesis and the research goals that it seeks to answer. Lastly, an outline of the rest of the thesis is provided for the reader.

1.1 Motivation

Computer Science has had and will continue to have a huge impact on modern society. Digitization is in increasing appeal, which results in ever-increasing demand in computer scientists and developers. In Norway, $\frac{1}{4}$ of all positions for developers will be unfilled by 2030 [1]. The same goes for the US, where the number of jobs far surpasses the number of educated developers [2]. In order to supply society with educated people equipped for the new challenges, computer science education should be of top priority and top quality in universities.

Luckily, at the same time as demand for computer scientists are at an all-time high, enrollments in computer science courses and educations are also at an all-time high according to a report made by the National Academies in the US [2]. However, this increasing demand also puts a high strain on the resources needed to teach these courses, as course enrollment often rises without teaching staff or resources following. This is a challenge in Norway, especially for undergraduate courses which have to be taught in Norwegian, a constraint limiting the pool of available applicants by removing those that do not master a Scandinavian language. This puts further pressure on personnel, infrastructure, and accentuates the need for new education styles that handles these issues and keeps up with the high enrollment numbers in universities.

These high enrollment numbers are visualized in Figure 1.1 for CS degrees in the US, and Figure 1.2 for Norway, and there is no evidence that the numbers are going to go lower any time soon. There has been exaggerated optimism about the state of computer science in the past, as the periods around 1995 and 2008 in Figure 1.1 shows, but one can not rely on that happening again.

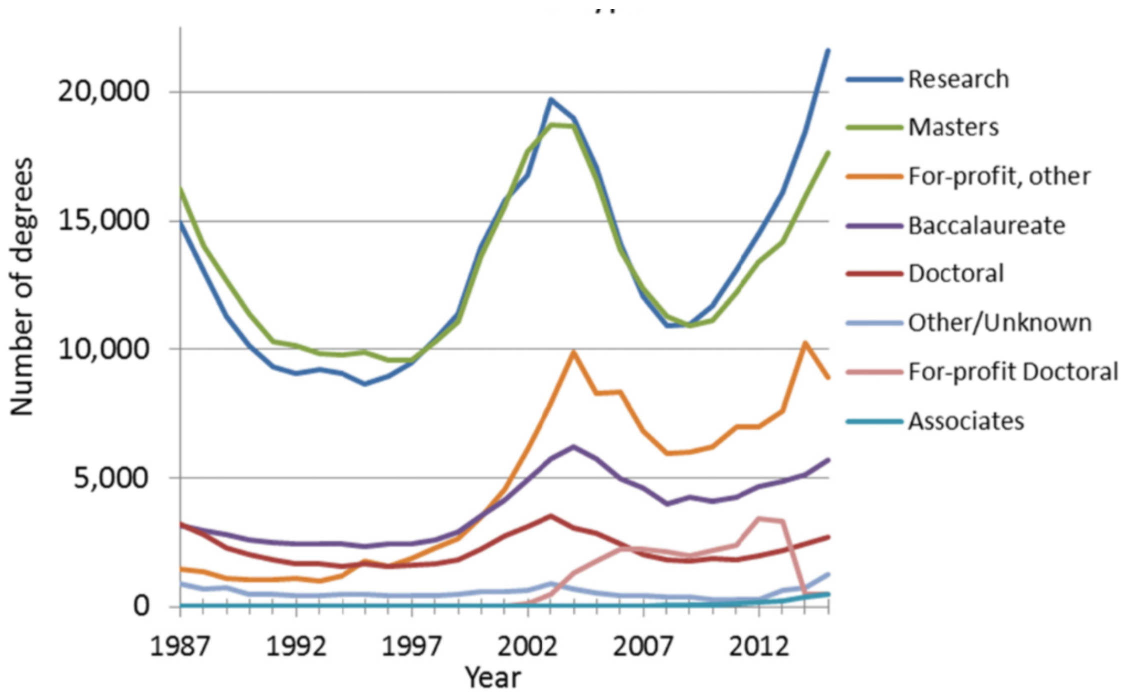


Figure 1.1: Number of CS Degrees in the US. The graphs show previous declines, but so far there is no evidence that this is going to happen any time soon [2].

The need for even higher enrollment numbers is also evident in Norway, due to the demand described by NOKUT in their report [1], and visualized with numbers from ”Samordna opptak” [3] in Figure 1.2. With both such a high demand and high enrollment in computer science, the resources that go into universities to teach computer science needs to be well utilized.

This thesis is written at and is focusing on the Norwegian University of Science and Technology (NTNU) in Norway and on education and students at this university. The course that has been the focus of this thesis is TDT4100, Introduction to Object-oriented programming.

Norway is one of the most developed countries in the world in leveraging technology, according to the Global Information Technology Report. This is illustrated in the Networked Readiness Index in Figure 1.3. This index shows how Norway scores compared to other similar countries for usage, skills, and impacts of technology. While Norway scores higher than the average from related countries in all categories, the skills category has the lowest gap. Improving computer science education is a vital step in increasing Norway’s readiness for the new fourth industrial revolution [4] and increasing technological skills.

To improve computer science education, old practices need to be questioned, and research into the best way to teach students computer science should be thoroughly explored. At NTNU in Norway, the most common practice, in any course, is to give out mandatory assignments that have to be completed to be allowed to take the exam at the end of the semester. These exercises follow the weekly lecture series that most courses have, shadowing the progression of the lecture plan. The assignments usually follow a schedule, with weekly or bi-weekly delivery. These deliveries reflect an attempt to control how the students approach the subject, what they do to learn, and at what pace they should approach the subject. The fundamental motivation

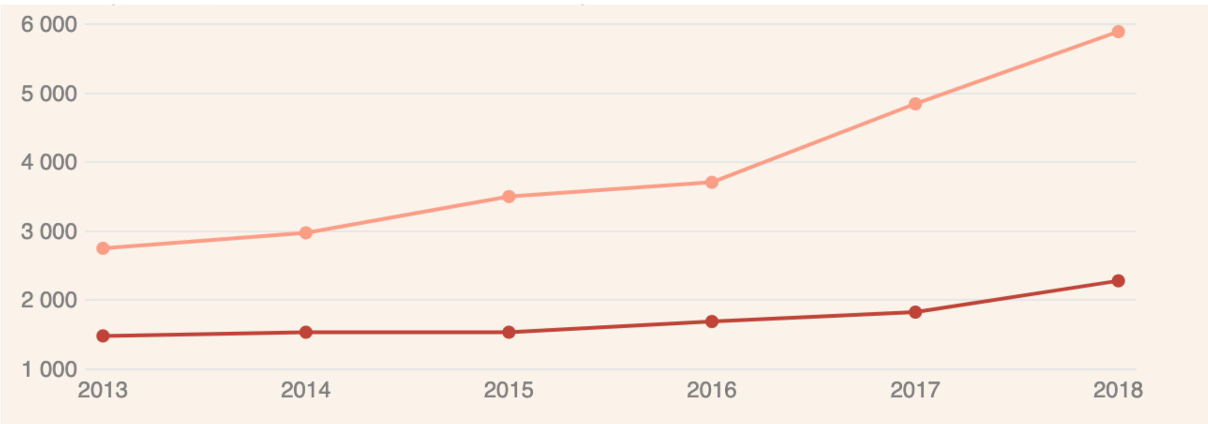


Figure 1.2: The number of ICT study admissions (bottom) and number of students wanting an admission into ICT (top), in Norway [3].

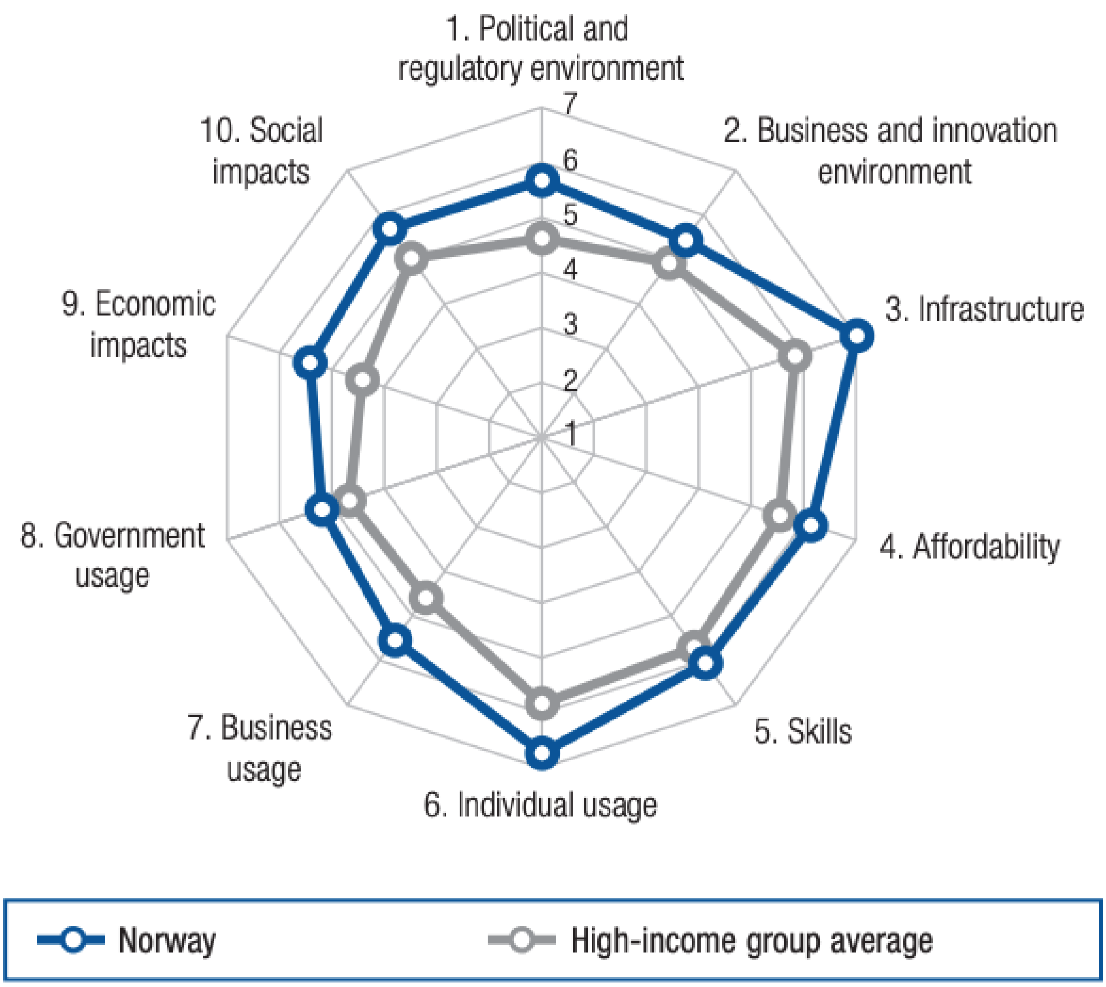


Figure 1.3: Norway's Network Readiness. [4]

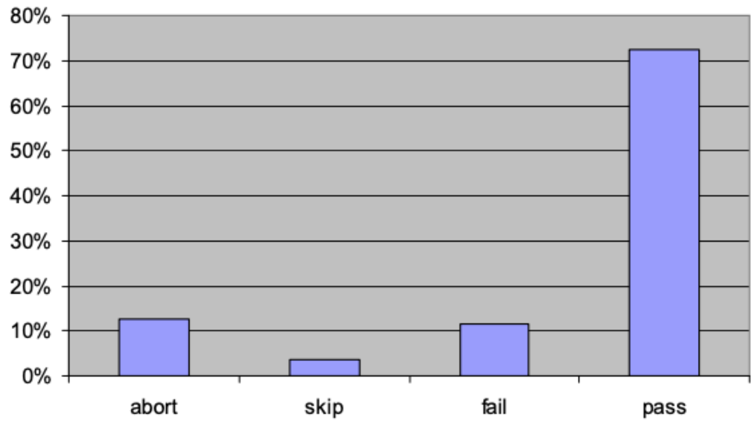


Figure 1.4: Failure rates in introductory programming throughout the world. Aborted students are students who aborted the course altogether, skip is students that attended the course but did not show up for the exam, fail is students who take the exam, but fail it, and pass are students that completed the course and passed the exam. [8]

behind this is to force the student to distribute their work evenly throughout the semester, and not cram all their work to the end before the exam. Besides, assignments help in testing the curriculum that is not tested during the exam.

Most commonly, in introductory courses, these assignments do not affect the final grade but are graded on a passed or not passed basis. As programs are files that are simple to copy and edit, plagiarism is also tempting and misused in programming assignments that are given out [5].

Failure rates in computer science courses is high [6, 7], at 28 % worldwide, as illustrated in Figure 1.4, and some research suggest it is even higher [8]. Looking at the numbers, it is obvious that whatever teaching measures can be taken to improve the courses should be done. It should be noted that recent research has suggested that the pass rates in introductory programming courses have increased the recent years [9]. Figure 3.1 also illustrates the failure rates in a introductory programming course at NTNU. The motivation behind this thesis is to explore whether mandatory assignments are helpful in introductory programming courses by investigating how it affects learning outcome and motivation for students. The idea is that computer scientists students should be motivated to learn the important foundations of computer science and will want to gain an understanding of the subject. Also, instead of relying on a reused exercise system that has been pretty much the same for the last years one can take into advantage the work that others have done before and utilize the best and most important resources that can be found on the web and in textbooks. Each student can choose for themselves how to master the subject, as there is no best way to learn anything that works for everyone [10]. Students have different learning styles and learn best when doing activities that are best suited for their particular style.

1.1.1 Personal Motivation

The personal motivation for writing this thesis has been my experience as a teaching and scientific assistant during my years at NTNU. Since my second year at university, I have closely followed students taking this course, and I have held weekly lectures to prepare students for the assignments.

I have always been unsure whether this is the right approach for teaching, especially for such a hands-on subject as introductory programming. I, therefore, set out when writing this thesis to explore what could be done to improve the quality of education and student life for programming courses at NTNU.

I hope with this thesis to look at the structure of assignments that we have been following for years at NTNU, to see whether they are helpful or damaging for students and whether it is worth looking into other options to improve computer science education.

1.2 Research Goals

This thesis aims to answer the following research question.

> How are students affected by mandatory assignments in an introductory programming course?

This question is further divided up into two main categories of research

1. What is the effect of mandatory assignments on students' learning outcome?
2. What is the effect of mandatory assignments on students' motivation?

In addition, while not the main focus of this thesis, students' time usage and learning approach in combination with mandatory assignments will be explored.

1.2.1 Hypotheses

The hypothesizes of this thesis is as follows. These will be further explored for statistical hypothesis testing in Chapter 4, and are written for use with statistical methods.

1. There is no improvement or reduction in learning outcome for students that do not have mandatory assignments compared to students who have mandatory assignments.
2. There is no improvement or reduction in motivation for students that do not have mandatory assignments compared to students who have mandatory assignments.

The idea behind writing the thesis, and the theory that it is based on, was that the learning outcome hypothesis would be supported, but that the motivation hypothesis would be rejected.

The expected result was that there was a difference in motivation for students having mandatory assignments, and students given autonomy.

The thesis also explores how time usage affects learning outcome in a constrained way.

1.3 Structure of Thesis

This thesis is structured as follows: Chapter 2 includes relevant theory and previous work in the area of teaching computer science and the theory of Self-Determination. Chapter 3 elaborates on the course that this thesis has followed, and how the course links up with the relevant theory. Chapter 4 explains the research strategy and statistical methods that will be used in the conduction of the experiment. Chapter 5 gives appropriate background information and more detailed information about the data collection process. Chapter 6 presents the results of the research and discusses the implication of these results. Chapter 7 concludes the thesis. Lastly, chapter 8 presents further work in the area of mandatory assignments in computer science courses in universities.

Chapter 2

Background and Previous Work

This chapter explains background information and theory that are relevant to this thesis, as well as previous work into relevant areas. First, we differ between formative and summative evaluation. Next, diverse approaches to teaching computer science are elaborated on, and previous studies on mandatory assignments and teaching computer science that have been found are described, and their findings summarized. Lastly, the theory of Self-Determination Theory is explained and put into context with a university setting.

2.1 Formative and Summative Assessment

To talk about the way students are assessed today, we need to differentiate between formative and summative assessments. The distinction between these two roles was first written down by Scriben in 1967. He explained summative evaluation as an assessment used to judge the value of an educational program, what had the student learned. Formative assessment targeted improvement for the student, and how they could improve learning [11]. Bloom extended this definition of the purpose of formative evaluation to ”Provide feedback and correctives at each stage in the teaching-learning process” [12]. Further explanation about this and the difference between evaluation and assessment is visualized in Figure 2.1

Summative evaluation is the most used at universities worldwide for assessment, namely an exam. While many studies provide evidence that formative feedback far outperforms summative feedback [13, 14, 15], a review that went through this research, observed that there was ”no agreed upon lexicon concerning formative assessment” and recommended methodological approaches in the efforts to demonstrate positive effects that could be attributed to formative assessments [16].

Further studies expanded upon previous reviews, inquiring after better definitions of formative evaluation, and conceptualizing well-specified approaches for methodology and process to recognize where formative assessment is helpful [17].

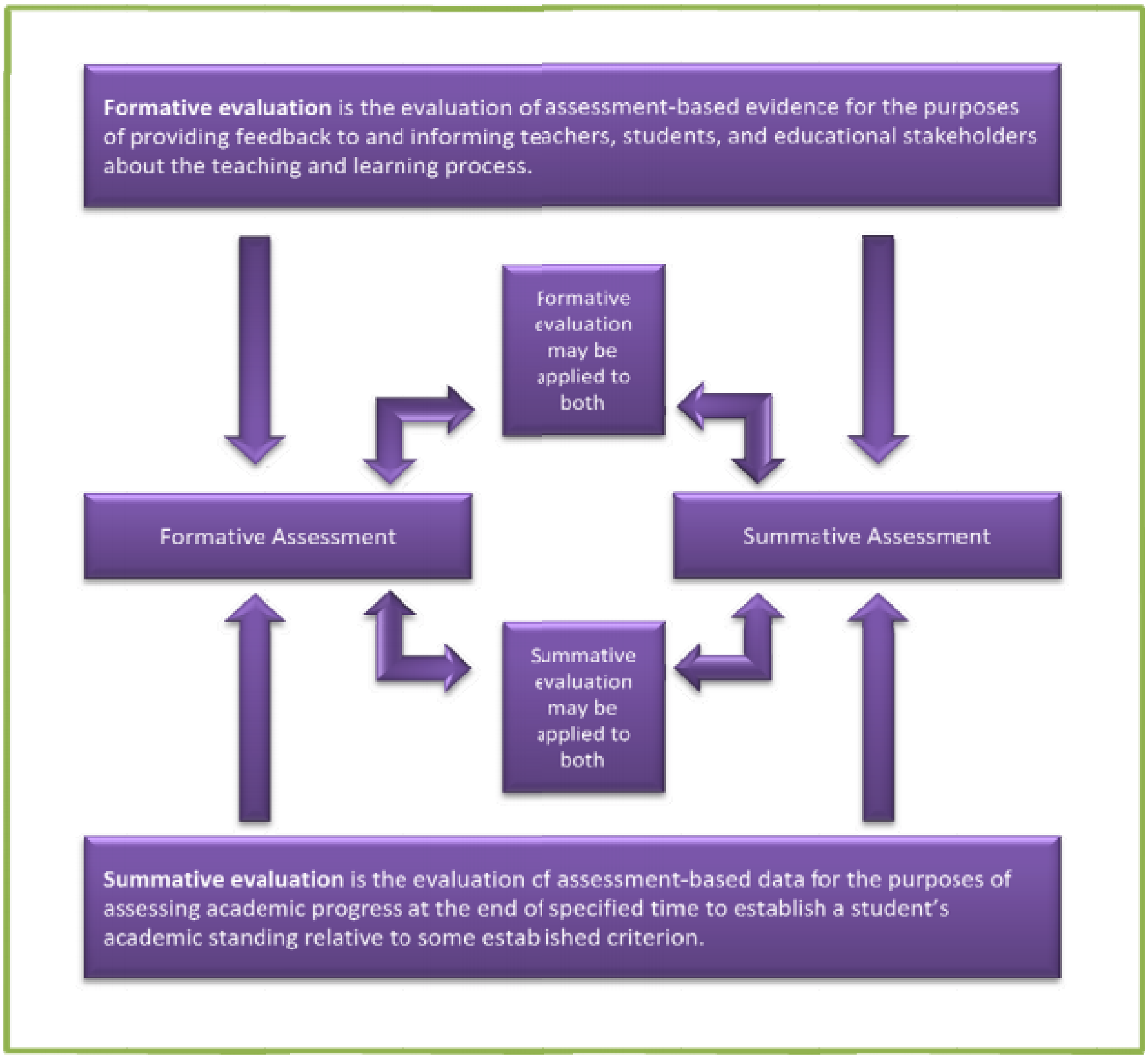


Figure 2.1: A figure detailing the difference between formative and summative evaluation [16].

Table 2.1: Intervening activities to increase CS pass rates. Adapted from [21]

Activity	**Description**
Collaboration	Activities that encourage collaboration between students
Content change	Change of the teaching material
Contextualization	Activities aligned towards a specific context
CS0	A preliminary course intended to give students a boost of motivation and preliminary knowledge
Game-theme	Learning trough the use of games
Grading schema	A change to how the students were graded. For example, assignments could weight more than the final exam
Group-work	Increased group-work commitment, such as group project
Media computation	Declare the usage of other medias, e.g video courses
Peer support	Support by peers as mentors og tutors
Support	Umbrella term for other support activities, e.g increased hours of teaching availability

Regardless of what is most effective, formative evaluation in its pure form is likely in limited use in Norwegian education today [18]. A British study by Jessop et al. concluded that study program descriptions affect the choice of evaluation platforms and that they lay strict guidelines for what tools to use by implicitly asking teachers to use summative evaluation for reliable grading. These descriptions limit the use of alternative evaluation systems, and modulize the study programs, leading to an increased amount of summative assessments [19]. Regardless of definition, multiple studies have shown that when the number of formative evaluation increases, students will learn more, and that it has a positive effect on students' learning [14, 15], also for the most low-performing students [20].

2.2 Teaching Introduction to Programming

Numerous techniques to teach introductory programming have been applied throughout the years. A systematic review of them was done by Vihavainen and Airaksinen in 2014 to compare the impact that different approaches have had on the pass rates of programming courses.

An overview of the different approaches is shown in Table 2.1. The articles reviewed had implemented these intervening activities in between semesters or years, therefore, other factors could also have affected the results [21], which also were incomplete concerning which activity was best suited for learning.

Numerous other studies have also investigated what type of activities are most useful to teach computer science. In a review by Luxton-Reilly et al. in 2018, a systematic review of the literature of introductory programming was done in order to get an overview, going through a total of 2189 papers. One of the four categories explored was teaching. Among other findings, they find that self-paced learning had few examples of usage in universities worldwide. Self-paced learning is a form of mastery learning where students are supposed to demonstrate they have achieved an appropriate level of mastery of a topic, before they can move on the next, more advanced, topic in the course. They also found that problem-based learning could increase motivation and social interactivity. However, little evidence that it increases the learning outcome of the students. Problem-based learning was mainly project-based, answering open-ended questions [22, 23]. Through the review, they found evidence that students preferred structured assignments [24]. A criticism of most papers from the writers of the review was about the context of the experiments performed in the different articles. There were little details about the activities and information the classes where the studies had been completed, making it difficult for a reader to determine if the results were transferable to their own teaching context [25].

An older review from 2003 by Robins and Rountree looked at early psychological and educational studies of programming. They summarized that programming is not easy. There are bountiful difficulties that must be mastered, many of which must be dealt with at the same time. Most of the time spent in teaching introductory programming is related to learning the semantics of a new language and putting this together to write a program. Models of how to build applications and the understanding of the strategy needed to make programs are overlooked as too difficult from the start. They recapped with recommending a fundamental change to teaching programming classes, with a substantial shift in both curriculum and teaching style, with more focus on strategy and models needed instead of focusing on semantics and merely getting a program to run [7].

An interesting contrast to the paper by Robins et al. is Luxton-Reilly with his paper, ”Learning to program is easy” from 2016. In this paper, he argued that maintaining the ”programming is difficult to learn” view could lead to poor student outcomes and teaching practices. The main problem is the expectations established in an introductory programming course, and it is these expectations that should be changed to create a more equitable environment. Luxton-Reilly argues that the assessment that we use to evaluate students’ learning in computer science may be too ambitious, and we expect too much of the students in their first years. These errors may be significant factors for the dropout and failure rates in programming courses, and could also be essential factors for inequity between genders in computer science. Multiple multi-national studies provide evidence that novice student programmers do not perform at the level expected by researchers [26, 27]. Luxton-Reilly challenges the research community to create research-based results of what novice programmers can achieve after their first programming course, to create more realistic expectations [28].

In an evaluation of different teaching approaches to introductory programming from 2015 by Koulouri et al. they studied three distinctive factors for how to improve CS1 programming. The choice of programming language and teaching problem solving before programming were found to yield significant improvements in student performance, however, it had variable effects on acquisitions of basic concepts in programming. The last factor was how to use feedback effectively and formative. Here, they found that formative feedback was not useful unless students actively sought out and responded to feedback. To be effective, feedback should be timed and targeted to specific features that one wants students to improve [29].

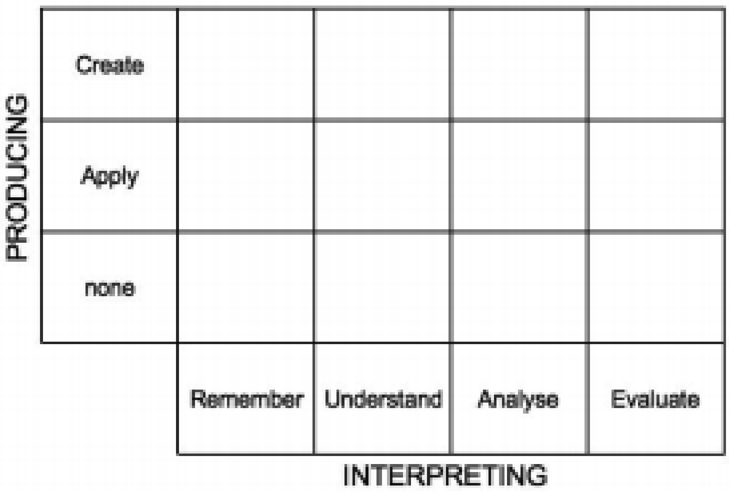


Figure 2.2: A taxonomy for computer science learning by Fuller et al [32].

As computer programs are files that can be run by a computer, there has been significant research into how to automatically grade and correct programming assignments, reducing the strain on teaching resources. These have plenty of issues that need to be looked at, especially for a system that grades the student based on these assignments [30]. This is also used at some courses at NTNU.

2.2.1 Taxonomy

An educational taxonomy can be a useful tool for structuring learning objectives and student attainment of these objectives. Different subjects of learning make taxonomies hard to use, as it is hard to find agreement on the classification of their items [31]. A taxonomy for computer-science was devised by Fuller et al. to differentiate students' abilities to design and build software. The reason to use this is to assess a students capabilities in computer science and engineering [32]. This taxonomy is shown in Figure 2.2

The dimension of Fuller's taxonomy matrix represents two distinct spans of adeptness. This is the ability to understand and interpret existing code (bottom), and the ability to write programs of your own (left), with different levels for each competency. The different levels are adapted from the revised Bloom's taxonomy, a commonly used taxonomy in education [33].

Students taking a course in computer science may choose different learning paths through the matrix [34]. Some may try to apply concepts that they remember, but that they do not fully understand, thus locking themselves in a trial and error approach. Others may purely learn by theory and place themselves in the none/Evaluate category, which means they can read and analyze existing code, but cannot produce code that solves a problem. Others may be more practical, being placed in the create/understand cell of the matrix. They can apply existing code and theory to produce code, but cannot analyze or evaluate the code. Recognizing where a student is in the taxonomy to employ relevant activities and find out where the student is lacking could significantly help a student in reaching a higher level of competence.

2.2.2 Learning Styles

An important concept to understand is that there is no set way for everyone, and students have different learning style preferences. Various approaches to modeling the different types of learning have been tried out [35]. One of the more used has been from Felder who classified learners according to the Felder-Silverman Learning Style Model [36]. Here, the learner is classified into two types of learner in five different categories.

- Active or reflective learners. Active learners learn by trying new things out or working alongside others to learn something. Reflective learners learn best by thinking through things, preferably on their own.
- Sensing or intuitive learners. Sensing learners learn through practical and concrete concepts and oriented through facts. Intuitive learning learn by conceptual things, oriented toward theories and meanings
- Visual or verbal learners. Visual learners prefer pictures, charts, and figures. Verbal learners prefer written or spoken explanations
- Sequential or global learners. Sequential learners learn in orderly steps, one concept at a time. Global learners learn in large leaps and think of the whole system
- Inductive or deductive learners. Inductive learners go from specific to the general. Deductive learners go from the general to the specific.

It is clear with students' learning in different ways that there is no way to please everyone. A freer "choose-your-own" path of education is then intuitively said to be a better approach, suited for different styles of learning.

This set of learning styles from Felder is just one of many theories about different styles of learning. However, it illustrates the need for an education model that is adapted to suit different styles. If students learn in different ways, we can not use the same principles to teach all students the same curriculum [35].

2.3 Mandatory Assignments

As limited research has been done on how assignments affect university students, research into similar categories has been conducted. This is mainly how homework, in all areas of school, can affect students' motivation and learning outcome, and the publications on this topic. It is appropriate to point out that compulsory exercises in a university setting are not the same as homework in primary and secondary school. In lower education, the schedule during the day is packed, and homework assignments must be done at home, in the evening. At university, on the other hand, each course has limited hours of lecture per week, which at NTNU is not mandatory to attend, so there is vacant time also during regular work hours. In this time, work with exercises in the course is intended to take place, and time is set aside for this in a university schedule.

The reason for having a mandatory assignment in a course setting is often twofold. Compulsory assignments could be there to qualify students for the exam, or it could be to qualify student besides the exam. If the purpose is to qualify students for the exam, the assignments are meant to yield the same learning outcomes as those that are tested by the exam. The exercises are there to help students in working evenly during the semester. Numerous studies about procrastination show that students want to delay any work they have to do as long as possible [37, 38, 39, 40]. Assignments are a way of combating students procrastination.

Having assignments besides the exam is for the assignments to address other learning outcomes of the course than those of the exam. The exam could be too short to test all that should be learned in the course, and such the assignments are needed as a supplement in certifying that the students have learned all that they are supposed to. For example, this could be practical knowledge like a chemistry lab, which is unfeasible to test during an exam. Math assignments in a math course are more straight forward learning to prepare students for the exam. Programming courses fall in between these two examples, with assignments often mainly focusing on preparing students, but may test larger collaboration projects and coding challenges for which the exam does not have enough time.

Previous work on homework

One of the early research results into the subject of mandatory assignments was published in 2002 by Trautwein et al., with data collected from 1976 7th-graders in mathematics classes. Here, mainly, three results were significant. Firstly, the frequency of homework assignments had a positive effect on achievements in math tests. Secondly, lengthy homework assignments had a small negative impact on learning outcome. Lastly, and most important in relevance to this thesis, monitoring of homework completion did not contribute to achievement gains. This means that setting the homework as mandatory did not achieve any more learning outcome [41].

Other studies have concluded differently, so the results in the literature are inconsistent. Multiple studies show a positive relation between math achievement and homework [42, 43, 44], while others finding a non-relation, or even a negative impact on achievement, among these a study from 2010 on 28 different schools, where neither frequency nor homework time had any relation to performance in class according to Jong et al. [45]. An interesting find here is that often conflicting studies may be from different countries, with US students having a higher achievement gain from homework than the rest of the world [46].

Similar inconsistent results have been shown in studies linking homework and science achievement. Some were finding a positive relation [47], with some also going further in findings on intrinsic motivation, using Self-Determination Theory. Here, the results showed that intrinsically motivated students did significantly better on a test. However, more choices of different homework types did not lead to an increase in intrinsic motivation, according to Christensen [48]. Again, other studies have shown otherwise, with no correlation between the frequency of homework and science scores [49].

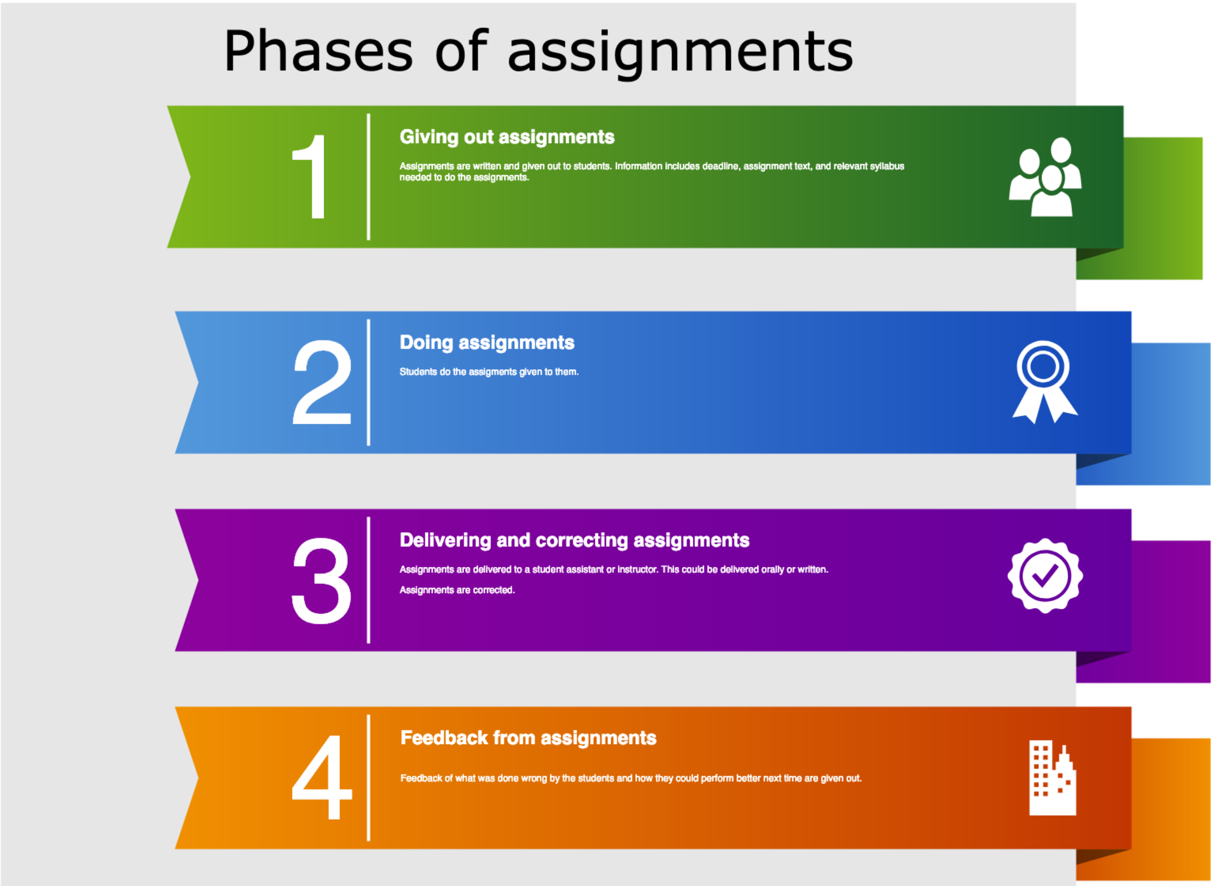


Figure 2.3: Timeline of the assignment phases. While most studies agree that phase one and two is important, there are conflicting results whether feedback and getting assignments corrected are helpful for students

A variety of factors may have contributed to these inconsistent findings in previous literature. Like the type of homework, grade, how achievement is measured, or what kind of homework indicators that have been used. Studies have been convened on different data, including total time spent on homework, the frequency of homework, the percentage that was completed, the effort needed to complete the work, or the grade given to the homework if being evaluated by the teacher.

It is easy to think that homework, as looked on in Norway is mandatory and has to be delivered to your teacher, but of the research reviewed in this thesis, none has shown a clear correlation between homework feedback and student motivation or achievement gain. It should be noted that homework completion rate has been shown to have an effect, but not the actual deliverance of the homework.

It is important to emphasize that these studies have been done on homework, at a lower level of education than a university. Therefore, the results are not necessarily applicable and transferable to a university setting.

Research on assignments college and university level

An interesting study from 2010 by Gutarts et al. regarded college-level calculus, interestingly noted the same observation as I have made, that there are few studies done regarding the effects on homework or mandatory assignments. They divided a class into two groups, one with compulsory and graded assignments, while the other was assigned the same assignments, but instead given weekly quizzes as grading activity. The hypothesis that group one would outperform group two in a later test. The results, however, revealed that there were no statistically significant grade difference between these two groups [50]. This result builds on early equal results that monitoring assignments completion, rather than just giving them out as an aid in learning the curriculum, does not affect students performance [51]. However, if students are not given any exercises to aid in learning the syllabus, some results put them at a disadvantage compared to students getting mandatory assignments according to Cartledge [52].

Similar results were found in college degree economic course, in a study by Miller looking at feedback and grading of assignments. They tried out a concept called selective grading, where only a few select assignments were graded, and it had no effect on students' learning outcome, they produced at the same quality and delivered the same number of assignments [53].

Research on whether assignments are helpful in programming courses are limited. A review from 2016 by Danielsiek et al. about ways to teach computer science found no evidence that results on assignments were any indication on how students would perform at the exam. This was regardless whether the assignments counted towards the final grade, or whether it was just a stepping stone for being allowed to take the exam. [54].

A Norwegian analysis by Haugan and Lysebo from 2018 argues why the number of mandatory assignments in engineering education should be reduced, in a very similar experiment as this thesis has performed, but on a larger scale. They noted that in Norway, the use of mandatory assignment has increased, without any quality improvement in students' learning outcome. In this

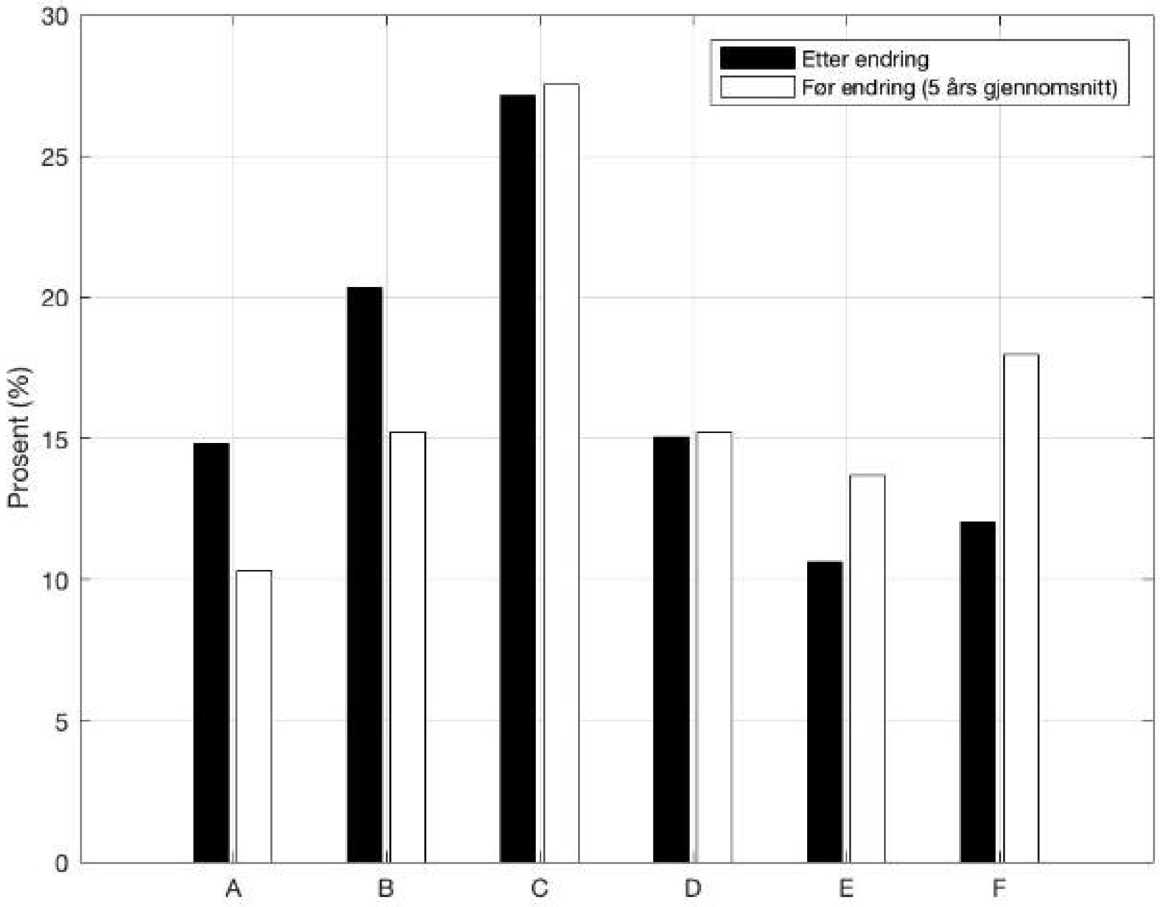


Figure 2.4: The average grade before and after the restructuring of teaching program by Haugan and Lysebo [18] Black columns show results before the intervention, while white columns show the average grade for the next five years after the intervention.

experiment, they replaced the compulsory assignment evaluation with a formative assessment with no mandatory deliverance in multiple courses. They asked themselves the questions:

1. Can pure formative assessment be a good option for the extensive use of assignments and will the students' effort and time usage change?

2. Are the students' benefits of pure formative assessment dependent on their level of prior knowledge?

In this study, they concluded with multiple important findings. Among them that the student, now with less necessary work, spent more time on each course than before, one of the most important reasons for having mandatory work in the first place. They also found that the average grade increased after the restructuring of the teaching program, as shown in Figure 2.4. This also included, to their surprise, the results for the students with the worst results on a preliminary test [18].

2.4 Self-Determination Theory

The hypothesis of this thesis and the experiment is based on the assumption that students are more motivated when they get to choose their way of learning something, instead of sticking to a rigid university system. Behind this hypothesis is the theory of Self-Determination.

Self-Determination Theory (STD) is the macro theory that explains this concept. Self-Determination theory focuses on the motivation behind the choices that we make, without external influence on our decisions. Deci and Ryan (1985) coin the definitions and process of intrinsic and extrinsic motivation and how students can use self-determination theory for building their competence. In an article from 2000, they explain how intrinsic motivation is preferred.

> Comparisons between people whose motivation is authentic and those who are merely externally controlled for an action typically reveal that the former have more interest, excitement, and confidence, which is turn is manifest both as enhanced performance, persistence, and creativity [55].

There are three aspects of SDT, which leads to happiness in our work. This can be called the three nutrients of intrinsic motivation, and are the three basic physiological needs in STD, the accomplishment of which lead to wellness in your life [56]. These are illustrated in Figure 2.5 and include competence, autonomy, and relatedness. You cannot have everything from day one in work, but eventually, the goal is to accomplish all three [55].

Autonomy and competence together lead to intrinsic motivation, which is defined as engaging in a task for the rewards inherent in the task, such as interest and enjoyment. Intrinsic motivation is further explored in Section 2.4.4. On the opposite side, extrinsic motivation is engaging a task for the rewards outside of the task, such as grades or toys. Together, all these three nutrients will lead to passion and intrinsic motivation in the work that you do. These concepts are further described in the next sections.

2.4.1 Autonomy

Autonomy is most controversial, but also most central when it comes to intrinsic motivation. Autonomy is a behavior that is self endorsed. You are self-initiating of the tasks that you do. You may be given tasks by your university or workplace, but you can choose how to do these tasks yourself, and you may have a voice when it comes to which tasks you are given. Humans want to have choices and control of our actions [58].

2.4.2 Competence

To feel mastery of things that are important and the work that you do. People need to build up their competence and master the tasks they are assigned to feel that their work is of importance.

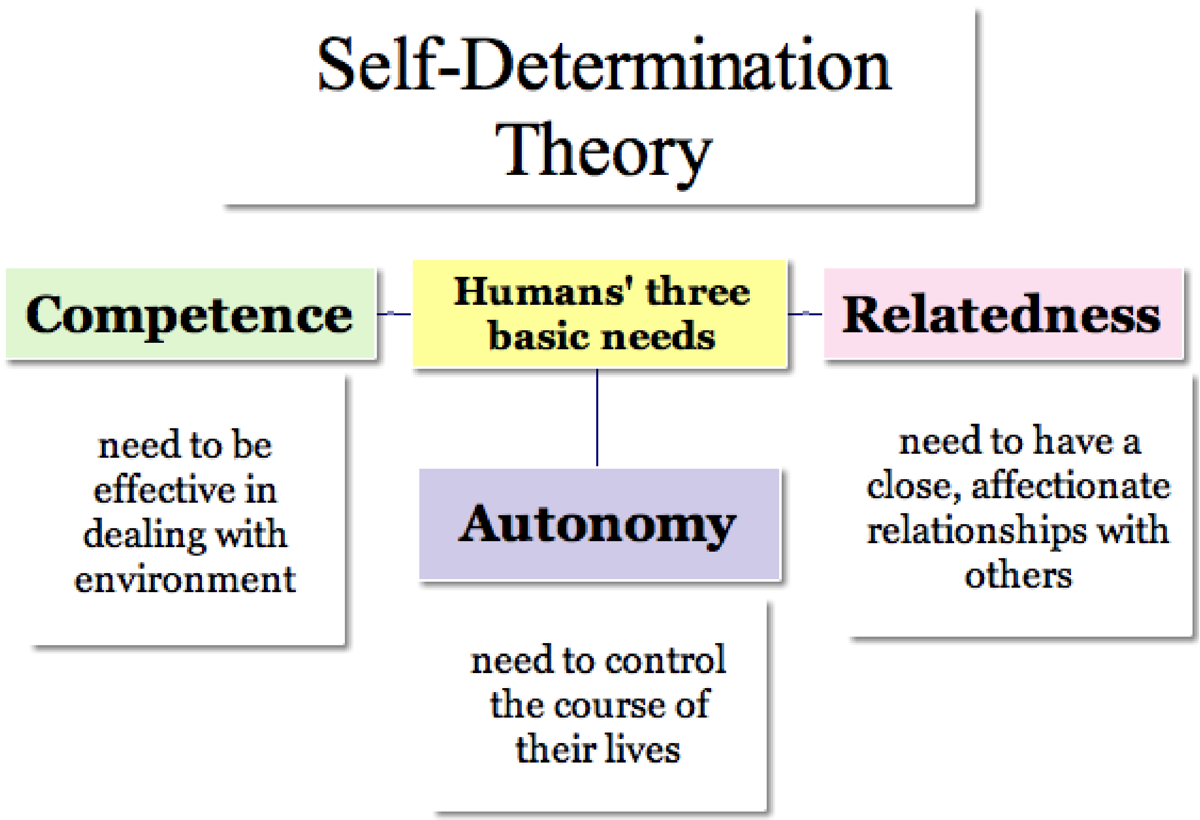


Figure 2.5: An illustration of the concepts of Self-Determination Theory [57].

Activities that are done should result in a feeling of self-development. Positive feedback also helps in satisfying the need for competence [59], while negative feedback can have the complete opposite effect [60].

Research has proven that people are curious and self-motivated. When they are inspired to do their work and are striving to learn, they will acquire mastery of new skills, and apply these skills to the best of their ability [55].

2.4.3 Relatedness

Relatedness is about feeling care for, and in companionship with other people. We require interactions with the people around us, and we want to maintain positive, frequent relationships. Repeated interactions with the same person are more beneficial than an everchanging environment, not being able to build long-lasting relationships. This has been a well-established fact a long time and is a vital part of the basic Maslow motivational hierarchy [61]. Humans are motivated by belonging to something, frequent, positive and long-term interactions and relationships with the people around us [62] Achieving this in university is out of scope for this report, but needs to be taken into effect for having a well designed university degree, not only focusing on the learning outcome and getting students qualified for a job.

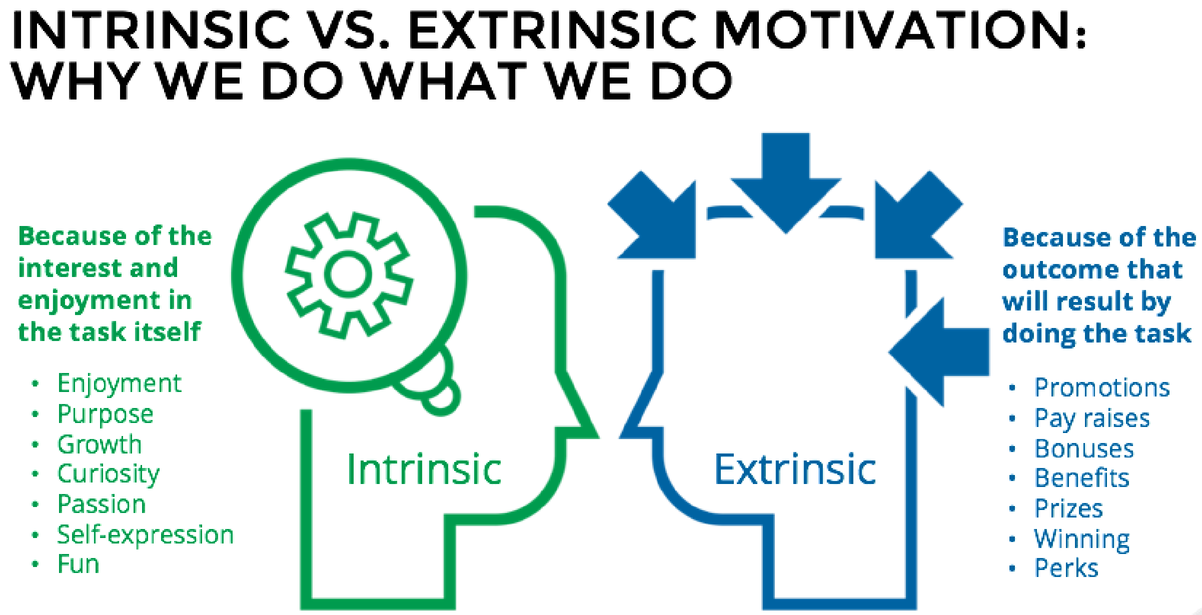


Figure 2.6: Intrinsic vs extrinsic motivation [56]

2.4.4 Intrinsic Motivation

Motivation is the need or desire to do something, and different motivational factors are what drives us.

To explain why Self-Determination Theory is important intrinsic motivation needs to be explained. Intrinsic motivation is a natural motivation, curated towards humans want of exploring, mastering, and curiosity. This is very evident in children, who do things out of their interest, while most of our school and work system are set up to work towards extrinsic motivation [63]. If you are driven by personal interest or enjoyment in work, you are inspired by intrinsic motivation factors. Extrinsic motivation is about wanting to get wealthy, famous or good-looking, and working towards their goals [55].

We know that if your orientation is towards extrinsic motivation goals, you will be less psychologically stable than if you have an intrinsic motivation like having meaningful relationships, personal growth, or contributing to your community. Employees that are intrinsically motivated work at a higher level of productivity and want to develop professionally, and intrinsic rewards to be much stronger than financial rewards in increasing employees motivation [64]. We all have a mixed of extrinsic motivation and intrinsic, so you cannot be all focused on personal growth, and do not care about extrinsic factors at all, but if extrinsic factors are dominant, you will be less happy and content than those more motivated by intrinsic factors [56]. Intrinsic factors and extrinsic factors are summarized in Figure 2.6.

These are not opposites. You can have motivation from both factors. However, multiple studies have shown that intrinsic motivation leads to better performance and increased learning. Extrinsic motivation can have adverse effects on intrinsic motivation [65].

2.4.5 Self-Determination Theory in Universities

In a study from 2004, researches tried to look at the difference between German and American university students, in terms of autonomy and competence. In Germany, students are given more freedom to learn the course material, while traditional American universities have a lot more assignments, and through that - more feedback. The study showed that German students had more autonomous motivation while they felt less competent than their American counterparts. However, the feedback they received was looked at more positively than the Americans, although the feedback was more infrequent.

In a review by Niemic and Ryan Self-Determination theory's approach to educational practice was evaluated. They suggested that a combination of intrinsic motivation and autonomous extrinsic motivations would engage and lead to an optimal learning environment in contexts of education. In traditional university learning, external controls, supervision, and evaluation are used, often accompanied by rewards or strict punishments. That could be better grades, or the ability to take the exam and pass the course. This is made by the belief that external contingencies enforce learning on students behalf.

Under such conditions, the feelings of enthusiasm and interest that could accompany learning are replaced with anxiety or stress, and students are no longer interested in what is taught, but rather what is needed to pass an assignment [66]. If students can identify themselves with the demands that are required of them, and they understand the benefit of the hours they put down in the course, extrinsic motivation can transform into intrinsic motivation. On the other hand, external factors like deadlines, surveillance, and testing will undermine interest and motivation for an activity [67].

Students are autonomous when they, by their own free will, devote their time to their studies. The need for competence is satisfied when students feel their work is challenging, but they can meet the challenge. The satisfaction of both of these is essential to keep intrinsic motivation. Many studies show that these are necessary conditions to uphold the level of intrinsic motivation high [68].

In educational settings, we can provide support for the need for competence, autonomy, and relatedness. Reeve and Jang published a paper on how teachers in education could support autonomous behavior in students. This includes actions like giving students time to do learn things on their own, giving out solutions to exercises, or free seating arrangements [69].

An experiment in 1984 divided college students into two groups, one who was supposed to learn the material to teach it to another student, and another group that was to be tested on the same material. Results showed that students who learned the content to explain it had higher intrinsic motivation and also had a higher learning outcome [70].

The educational process and activities in a university cannot be said to be fun or satisfying in a natural way. It is based on an extrinsic motivational process, work hard for better grades and a good job. For math problems, memorizing Latin names of body parts, and learning the periodic table obtaining intrinsic motivation may be hard. Students may, therefore, need other incentives to learn the material, and in need of extrinsic motivation, the most known is grades. Taking away this external regulation of promise to obtain a good grade, and students lose what motivation they may have to learn something [66]. Students may also motivate themselves through the use of identified regulation, a process in which you convince yourself that mastery of a concept is needed and useful for future competence in your profession. Such extrinsic factors provide a higher autonomy sense, on the borderline between extrinsic and intrinsic motivation. Numerous studies have given results that students that are given more freedom and autonomy perform academically better, and also obtained more interest and enjoyment in the course material, as well as being overall happier [71, 72, 73].

Chapter 3

The Course

In this chapter, the theories, and previous work from Chapter 2 will be put into context with the class that is the focus of this thesis. The course, its learning activities, and the structure will be explained.

3.1 About the Course

This thesis has been written with the focus on, and the aid of, a course in Object-Oriented Programming at NTNU, the largest university in Norway. The course, TDT4100 Introduction to Object-Oriented Programming yields 7.5 ECTS points and is for most students taken during their second semester. Typically, around 700 students are registered for the course.

This can be looked on as a traditional CS2 course, a more advanced introduction to programming, and the first introduction to object-oriented programming at the university. Students have had an introductory course to programming beforehand, which teaches procedural programming with the Python language. The TDT4100 course uses Java as programming language. Full-time students take three other courses in parallel with this course.

The course content and its official description from NTNUs webpage is as follows:

> Basic algorithms and data structures, constructs, and control flow in object-oriented languages. Modularization and re-use. Standard application programmers interface (API). Unit testing, error detection, and tools for this. Object-oriented design. Use of class, object, sequence, and collaboration diagrams in the UML. Use of design patterns. Simple app architecture. Java is used as an implementation language.

The course is known for quite high failure rates compared to other programming courses at the university, as seen in Figure 3.1. It should be noted that several students also use what is known as a tactical failure. This is gambit where students deliberately fail the exam to have

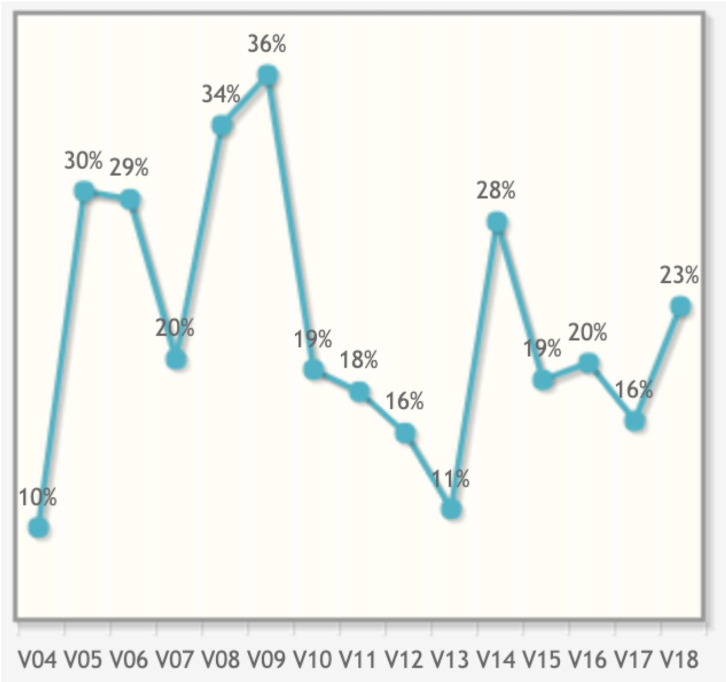


Figure 3.1: Failure rates in the course since it started.

more time to study during the summer and take the continuation exam to earn a better grade, rather than failing because of lack of knowledge. The extent of this activity is unknown. The variety in the failure rates is interestingly high, for a course with such a high number of students, and may have a connection with more students tactically failing when the exam is perceived as challenging.

While the failure rate is not alarmingly high compared to worldwide failure rates in introductory programming courses, one must not see this as a problem with the students rather than the course. As Bennedsen et al. note in their failure rates review:

> False views on failure and pass rates can have serious implications for the quality of introductory programming courses. A lecturer with a high failure rate might accept that ”this is just the way programming courses are since all programming courses have high failure rates” and consequently not take action to improve the course in order to reduce the failure rate [8]

The failure rates for the exercises can be seen in Figure 3.2, where most students qualify to take the exam after having done the mandatory assignments.

The grades in the course for the last exam can be seen in Figure 3.3. As can be seen, the grade is quite skewed to the right of a normal distribution. Low grades given can be an added factor of discouragement for courses that already perceived as difficult [74]. It has also been shown to discourage woman even more significantly than men [75], which can be an added factor for gender inequality in computer science. Experiencing the introductory courses as too difficult may discourage students from finishing their degree [76]. Although we may not want to make the course any easier, we should provide the students with the resources they need to achieve

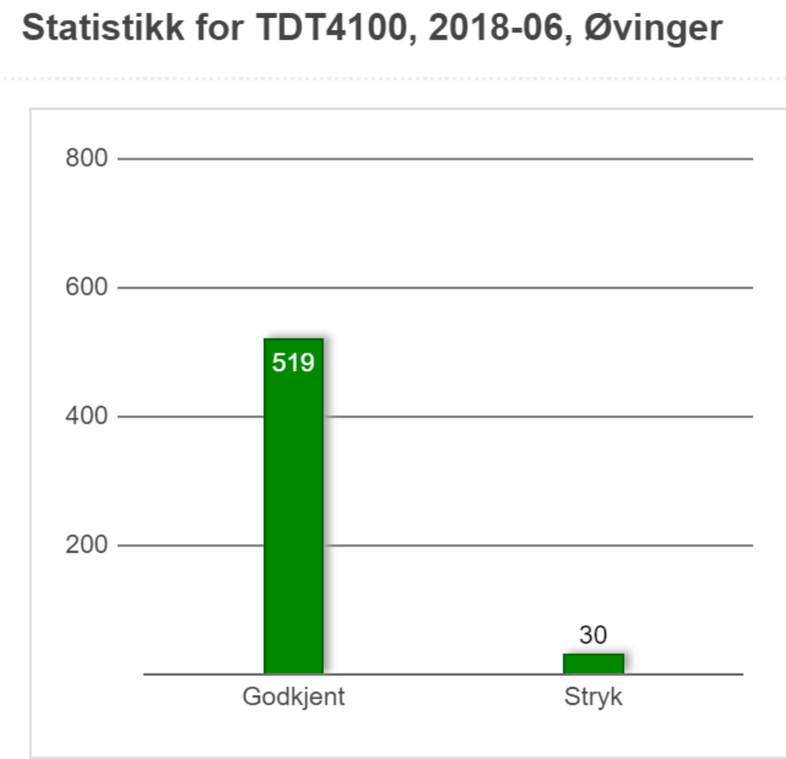


Figure 3.2: Failure rates for the exercises in the course, spring 2018. About 95 % passed the assignments and qualified for the exam

their best. Taking into consideration the paper by Luxton-Reilly, one may also want to consider if the expectations of the students are too high after a year of programming [28].

3.1.1 Structure of Course

The course is structured over 14 weeks of organized teaching activities. Table 3.1 shows what is taught in the weeks that the course is ongoing. The exam is usually held 4-7 weeks after the lectures, and teaching activities have ended, with time set aside for self-study during these weeks.

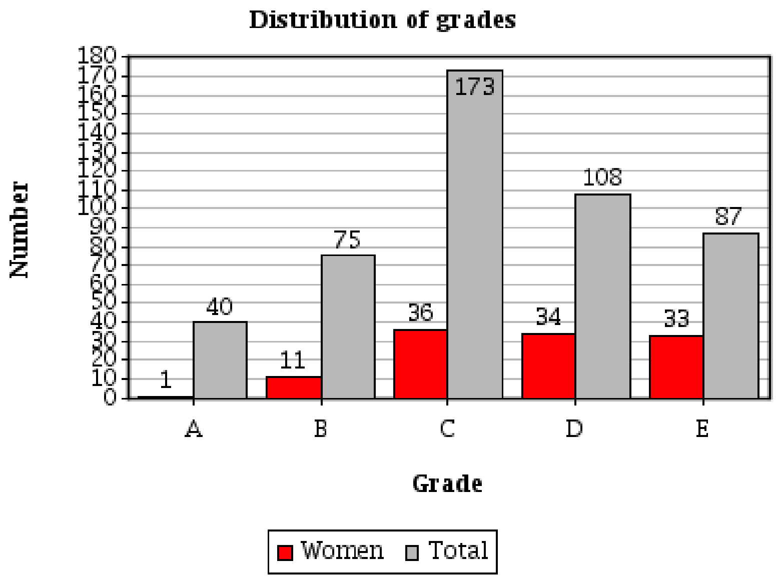


Figure 3.3: Grade distribution from the ordinary exam of TDT4100 in spring 2018

Table 3.1: Structure of course. What is taught during which week of the course.

Week	**Subject**
1	Introduction to the course
2	Classes and objects
3-4	Objects, encapsulation and validation
5-6	Objectstructures, UML diagrams, testing and debugging
7-8	Java essentials, Collection, Comparable, Maps
9	Exception handling, file management
11-12	Delegation and observer-observed pattern
13-14	Inheritance and summary
14-	Self-study and exam

3.1.2 Learning Activities

Table 3.2 shows the regular learning activities of the course. There are six hours of lectures each week, where two of them focus on the weekly assignments and tips and tricks for doing these.

Mandatory assignments are delivered almost weekly. There are ten assignments during the semester, where two of the more extensive exercises have a two-week deadline instead of one. The assignments are based on the learning curriculum for the current week and the week before,

Table 3.3: An estimate of how much time that is spent on assignments in the course

Activity	Estimated Time	Who is involved
Doing the assignment	Three hours	Students
Waiting in line for delivering the assignment	15 minutes per assignment)	Students
Approving the assignment	15 minutes per student	Students and Teaching Assistants
Overview of who has done the assignments Following up students that have missed assignments because of illness, etc.	80 hours (per semester)	Teaching staff
Registering which students can take the exam	6 hours	Administrative staff

thus shadowing the structure as shown in Table 3.1. Along with the exercises, tests are given out to test whether the student's code is correct, using the JUnit framework [77]. To pass the tests, students have to code correctly for all edge cases, as well as name their methods correctly after the task description. The assignments are delivered online but have to be demonstrated to a teaching assistant within a week after the delivery deadline. A teaching assistant is a student that has completed the course in an earlier semester and has been hired by the university to aid in teaching the course the current semester. They sit a designated number of hours available for approval and guidance of assignments. This is planned to be a formative evaluation, where the teaching assistant goes through the students' code, and see how they could improve, and if they have learned the concepts. However, talks with students and assistants have shown that this is mostly a summative evaluation where the students show the teaching assistant that the tests have passed, and then the assignment is approved.

Table 3.2: Learning activities in the course

Activity	Time usage
Lectures	4 hours each week
Assignments lectures	2 hours each week
Assignments	2-8 hours each week, dependent on Student
Self-study	Up to each student.

The assignments and the lectures make up most of the time spent on the course by the students. As a guess on how much time is spent on the assignments by the teaching staff and the students, see Table 3.3. Time spent creating the exercises is not mentioned here. As can be seen, many hours are spent on maintaining the assignment system. Some of them are completely unproductive, like time spent waiting in line for approval of the assignment. Those are hours that could be spent on guiding and teaching instead of administrative work of approving that the students have done what they are supposed to do.

Of the ten assignments, eight have to be delivered to qualify for the exam. They do not count towards the grade itself. Each assignment is graded on a point basis between 50-100 and to qualify for the exam one has to reach 750 points.

Table 3.4: Learning resources available in the course

Learning Resources	How it is used
Code examples	All codes written in lectures, and otherwise, is pushed to a gitlab server where the students can pull the code and look at examples
Blackboard page	Blackboard is a net based learning platform, used for delivery of assignments and as information channel
Wiki page	Wiki page with code examples, explanation of theory concepts and common mistakes
Book	The book Big Java, Early Objects has been used in the course for the past years.
Teaching assistants	Teaching assistant are available ten hours a day throughout the weekdays. They can help with understanding the course, and are used for approving assignments

3.1.3 Learning Resources

Other learning resources than those already specified in Section 3.1.2 are described in Table 3.4. The book in the course is seldom used by the students, except as a reference on the exam. What seems to be most used is the teaching assistants and the code examples that are given out from the lectures. Many students also use the internet and resources available online extensively to learn the course material.

3.2 The Experiment

In this experiment, we looked at the effect of mandatory assignment on student's motivation and learning outcome. Looking at the concepts from Self-Determination Theory in Section 2.4 we see that compulsory assignments, with deadlines, requirements of solving, approval demand, and where failing has the consequence of the student not being able to attend the exam are all outer impacts on the students' learning that may reduce motivation for learning the course curriculum.

Dropout rates of 30-40% are not uncommon in computer science courses. Studies have also identified excessive workload as a significant factor in the decision of a student to drop out of class [78]. If the course turns out to take too much time, and be more difficult than other courses, students will drop out [79].

To test out the effect of mandatory assignments in the course, an experimental group will try out a system where the assignments are no longer are mandatory, and replace the exercises with formative assessment. In this formative assessment, the course's learning goals are tested instead of a specific activity, to see if the student has learned anything, and if there is anything they should study more.

One of the things that have been looked upon is feedback, and that assignments are a way of giving feedback to the student. However, fast feedback is essential when solving exercises [80]. When feedback is given a couple of days after having completed an exercise, the student may not get any value out of the session. Another critical issue is the timing of feedback, and when it is given. According to multiple studies, feedback that is given together with a grade or approval can have less effect than only delivering the grade [81, 14, 13]. The experiment will thus not be an approval session, but a session with the teaching assistant where you talk about what you have learned. This makes the session approach a form of only formative assessment, which will give the student a more in-depth approach to learning [82].

In a pure formative feedback session, there is also no point in cheating, or copying off the exercise, a culture that is widely recognized and known at NTNU. Plagiarism, however, could still happen, by a student being embarrassed to meet their teaching assistant without any prepared material, even if admitting the lack of effort would have had no negative consequences.

The goal of the experiment was to see whether assignments increase the learning outcome of students and whether motivation is higher for students that are given autonomy. Based on the theory from Chapter 2, we expected that the learning outcome would be the same between the groups and that motivation would be higher for the experimental group. To see how the results were gathered, and how the students were tested, see Chapter 4

Chapter 4

Methodology

This chapter will describe this thesis scientific approach to answer the research questions based on research theory. The chapter discusses the research method, research design, data collection issues and considerations, and ethical dilemmas involved in the writing of this thesis.

4.1 Research Strategy

The two research questions in this thesis are quite different when it comes to approaches that are required to answer them. Research question one bases itself on measurements of learning outcome. Research question two is about students' motivation, how engaged and fun do they think it is to learn the course and as such is based on personal opinions. Due to these differences, a mixed methods research strategy has been used to answer these questions, utilizing multiple methods to answer the research questions.

For research strategies, we differ between qualitative and quantitative methods. A quantitative design is more suited for large-scale information gathering, measuring numbers in quantity, while a qualitative approach is more suited to look at peoples attitudes and opinions. Quantitative methods have been mostly used in this experiment for answering the research questions, while qualitative methods have been used for further insight.

For research question one, a quantitative experiment was chosen as the most suitable approach. A quantitative experiment outputs numerical data, which is ideal for statistical analysis and gives the options to vary on other variables. As data generation method, an observation of the tests that students performed was used, and the difference between these, as can be further read about in Section 4.5.1

For research question two, a survey was chosen as the most suitable approach, asking students whether they were more motivated now than when the course started. A survey is also regarded as quantitative data. In addition to the survey, interviews were conducted with the experimental group, giving further qualitative insight into their motivation. The testing of motivation can be read more about in Section 4.5.2

4.2 Methodology

A research method is a tool used to get insight into the research questions, and one should choose the method that is best suited to answer these questions. When scientific research is conducted, several choices have to be made that is related to the scientific method. These choices are closely related and can be looked at a number of levels when it comes to data collection, analysis of data, research design, and scientific theory. One of the first choices to be made is the choice of research method. In this thesis, an experiment has been selected as the research method. A literature review or a selected analysis on existing courses could have been conducted, but an experiment was conducted to get the most relevant data that was needed. Also, in order to improve computer science education at NTNU, experimenting with local conditions was paramount.

To answer the research questions in this thesis, a quasi-experimental research method has been utilized. Quasi-Experimental research is research that investigates the effect of an intervention on a research population but without random selection [83]. All participators in this study volunteered to be part of either the control group or the group testing no mandatory deliverance of assignments. Voluntary participation leads to questions about the validity of the results, as the average type of students volunteering for such groups may differ from the overall average student mass. As such, we may have a selection bias. This is partly controlled for with a pretest and a posttest to look at the change of rate in learning, instead of just the overall learning outcome at the end of the semester. The results are as well looked at in connection with previous results in similar courses and compared to these. In a quasi-experimental research method, you also have the limitation that there may be loss of participants during the study. A more random selection process could have been achieved by asking students to volunteer for the research project, and then randomly dividing them into the experimental and the control group. While this would have achieved a more random selection, it still would not have been entirely generalizable to the larger population as they are still students that are willing to volunteer for something. In addition, there was the added chance that fewer students would have volunteered to partake in the experiment, that made this strategy unpractical.

This experiment has been conducted with a pragmatic research mind-view. This incorporates multiple research methods and combines them to achieve a goal [84]. This has been done to use as many research methods as possible to obtain an answer to the research questions. This gives the opportunity to be flexible, and combine quantitative data, like test results and questionnaires with qualitative data. Examples of qualitative data include interviews with participants.

4.3 Data Collection Strategy

The strategy that has been used in this thesis is to get two groups from the pool of students taking TDT4100 the spring 2019 semester. How the participants were recruited can be read about in

Table 4.1: Data sources used in this thesis.

Data Source	**How it is used**
Preliminary Test	Assess early knowledge in the course
Weekly reports	For time usage and self-reflection of how to learn the course. Also used to see how motivation changed during the weeks.
Posttest Test	Assess difference in results between first and second test, and compare the students following the assignments, and the students not delivering assignments.
Interview with teaching assistants	Qualitative data for whether any difference was perceived in the two student groups.
Interview with students	Interview with students to assess motivation, and how they felt about assignments.
Final survey completed and other data	Survey asking about motivation, whether assignments had been
Initial survey	Initial survey about motivation, previous grade in introduction to IT and what they were studying.

Section 4.7. One group was given exception from the ordinary assignment program and were to deliver no assignments. As the course description from NTNU still demanded a delivery as part of the course acceptance, they would have to meet up biweekly to demonstrate the knowledge that the assignments were supposed to teach them. These sessions focused on the learning goals of each assignment, but the students were given autonomy to decide for themselves how to meet these goals. This was done in any way they wanted, they could either talk through the theory and explain the process or idea behind, or they could show some code that they had programmed to explain their points.

The other group was recruited as a control group, to compare the results of the students following the assignments, and the students not having to take the exercises.

The pretest and posttest control group design was chosen because of its suitability for educational experimentation, as presented by Cohen [85], to negate issues with internal validity. Quasi-experimental research was conducted for ethical reasons, which can be read about in Section 4.6.1

For results, both groups took a preliminary test, to assess their knowledge in the course after two weeks. They were then given weekly reports to report how much time they had used, what they had done to learn the material, and what they were planning to do next week. This data was gathered, even though it was not the main focus of this thesis, to be used in further research into how students work. This data and its opportunities can be read more about in Chapter 8

In the last weeks of the course, they were given a posttest, to assess their change of knowledge from the first test. This test was more complicated than the first test, given that they should have learned a lot during the semester. The results of these tests were used to answer research question one. At the same time, as given the posttest, they were also given a final survey, answering questions about motivation. Table 4.1 provides a summary of the data sources that have been used in this project.

For research question two, the primary sources of data is a combination of qualitative and quantitative methods, with the main focus on qualitative. The quantitative approach was a set of interviews conducted at the end of the experiment, to find out how the experimental group felt during the experiment, ask questions on motivation, and how they approached learning them-

Table 4.2: Research variables. Independent variables explain what has changed for the groups, dependent variables what is observed in the groups, and controlled variables what is kept the same.

Independent Variable	Dependent Variables	Controlled Variables
The assignments given to the students in order to qualify for the exam	The result on the pretest and posttest. Weekly questionnaires given to the groups. Question sheet at the end of the experiment	Both groups are asked the same questions, and given the same tests Participants in both groups share the same background knowledge, and takes similar degrees.

selves the material.

The quantitative data collection was the combination of an online survey conducted at the sign up of the two test groups, as well as a similar survey conducted at the end. In this survey, the question, ”On a scale from 1-6 how motivated are you for the course”, was asked. On the last poll, the question ”Are you more motivated now, than when you started the course” was asked as well, as seen in Figure 4.1. This was done to distinguish those who felt they were already on top of the motivation scale at the start of the experiment period, to see if they considered if any changes had happened. Questionnaires were chosen for use in this thesis since it is a very efficient way to collect large sources of quantitative data from different respondents. Compared to only doing interviews, it is easier to ask multiple students the same questions. The disadvantage is that one can not obtain additional information or have a dialog with the respondents. This creates a challenge in asking the correct questions for the data gathering [86].

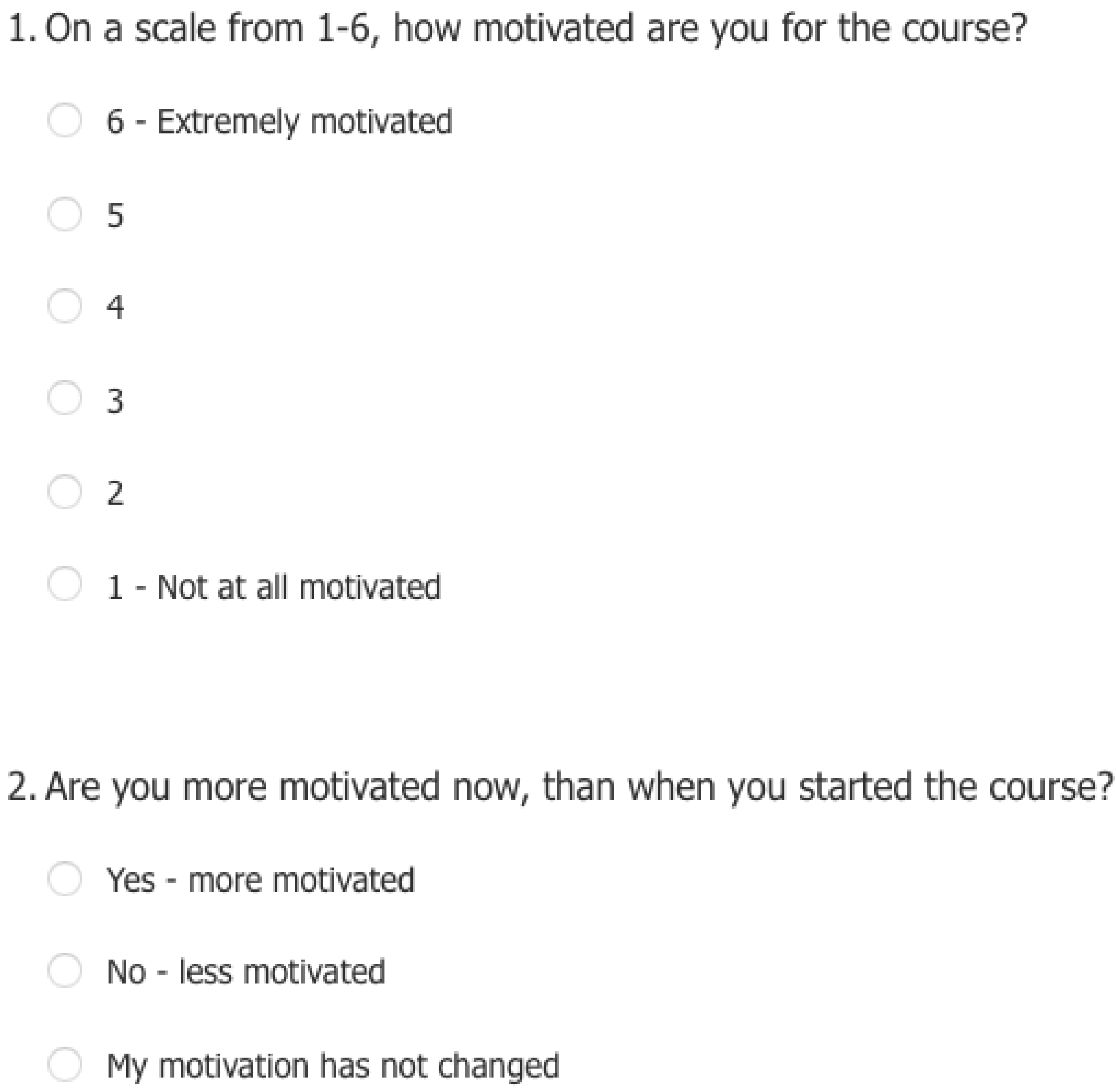
The survey was completed by the user at their own choice of location and time during the course of a week. That meant we were not able to control the environments, but simultaneously not suffering from the user giving the answers that the observer wants to see.

4.4 Statistical Analysis

4.4.1 Variables and Hypothesis

Any scientific experiment is conducted to disprove a hypothesis. Based on the results, one can argue for refutation or that the result seems to support the hypothesis. For the statistical analysis of hypothesis testing, variables are used. In that regard, it is essential to differentiate between independent, dependent, and controlled variables. This can be explained in order of what is changed, what is observed, and what is stationary during an experiment [86]. An explanation over these variables for this study is seen in Table 4.2

To measure the effect of the independent variable, three main dependent variables have been used. The difference between the pretest and the posttest was used to analyze learning outcome, as well as looking at the test results. The questions about motivation the end of the experiment was used to look at motivational boost if given another assignment system. Lastly, weekly questionnaires about time usage and how they spend their time was used to evaluate time usage


1. On a scale from 1-6, how motivated are you for the course?

- 6 - Extremely motivated
- 5
- 4
- 3
- 2
- 1 - Not at all motivated

2. Are you more motivated now, than when you started the course?

- Yes - more motivated
- No - less motivated
- My motivation has not changed

Figure 4.1: Online survey about motivation at the end of the experiment.

of assignments. Based on these variables, the null hypotheses for the experiment was defined as:

- **Learning outcome - H0:** There is no improvement or reduction in learning outcome for students that do not have mandatory assignments compared to students who have mandatory assignments.
- **Motivation - H0:** There is no improvement or reduction in motivation for students that do not have mandatory assignments compared to students who have mandatory assignments.

The following alternate hypotheses have also been formulated:

- **Motivation - H1:** Students with no mandatory assignments will have increased motivation compared to students who have mandatory assignments after the assignment period is over.
- **Learning outcome - H1:** Students with no mandatory assignments have a higher learning outcome compared to students following the assignment system.

These hypotheses will be used with a statistical method to see if the result indicates a denial of the hypotheses. If the H0 hypotheses are rejected, that will give support for the alternate hypothesis. If H0 is not rejected, we can not outright reject H1. However, we will be unable to reject H0, and that any difference in results is from pure chance or other variables that we have not taken into consideration.

4.4.2 Statistical Method

This experiment has been done as a quasi-experiment with a pretest-posttest design to compare results between the participating groups. What was measured is the effect of the inference, more detailed what happens to students that are not given mandatory assignments. With different statistical methods, one can either use the difference between the posttest and pretest, known as the gain score as one dependent variable, or the pretest and posttest combined as statistical variables. The pretest-posttest method is used to get a level of internal validity, with the rate of change instead of just one test measuring the learning outcome, as there could be various other factors contributing to the level of competence. If the pretest and posttest are used some changes to the scores need to be conducted to correct for the nonequivalent group design (NEGD), this will be elaborated on in Section 4.4.4. The overall result we were searching for was

- Is one of the groups displaying a statistically significant difference in the improvement of results between the pretest and the posttest?
- Is one of the groups displaying a statistically significant difference in change of motivation between the start of the semester and the end of the semester?

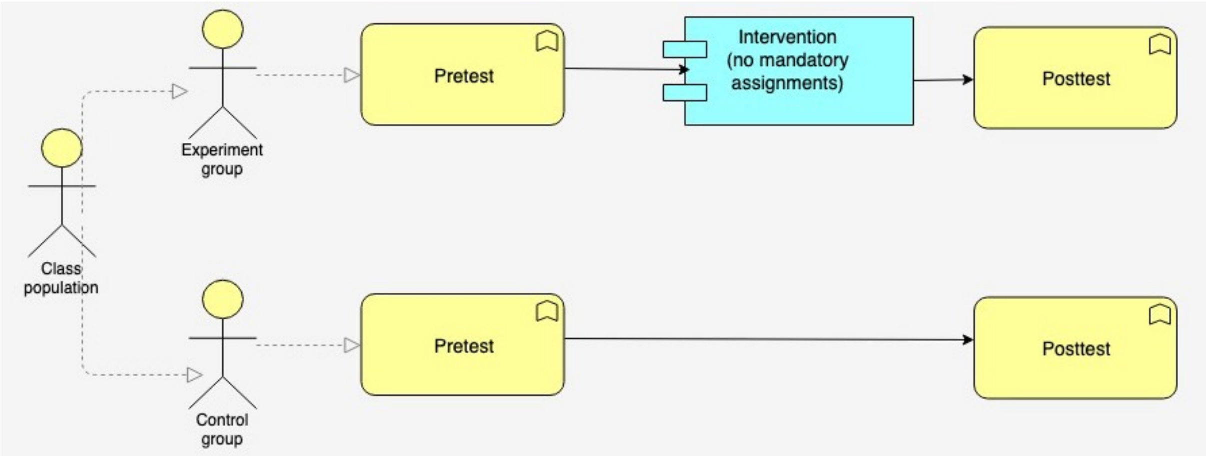


Figure 4.2: A two-group pretest-posttest experiment. The first group has an intervention and one wants to measure the effect of this intervention.

- Is there a difference in time spent between the different groups, compared to the results on the tests?

Note that contrary to the other two elements, time spent on the course will not be evaluated with a statistically significant requirement, as it is not the main focus of this thesis and its research questions. It will, however, be looked at, to discuss better the result of motivation and learning outcome, and also results collected during this experiment can be further researched to find how time affects the other variables.

The two-group pretest-posttest design is often visualized as in Figure 4.2. Note that the selection from the class population and to the control and experimental group has been voluntarily, and not the result of random selection.

There might be pretest differences between the groups, due to the non-randomized design, so any statistical method needs to take this into consideration [83]. Two methods will be focused on, which will be explained in the next sections. The independent t-test, and the Analysis of Covariance (ANCOVA), which are methods that are often used for the analysis of two group pretest-posttest experiments.

4.4.3 The T-test

To look for statistically significant results between these two groups, a statistical method is used. We will first compute the Gain score. The gain score is a simple equation that is calculated for each person that has taken the pretest and the posttest.

$$Gain = posttestscore - pretestscore \tag{4.1}$$

This simplifies the statistical calculations, in that gain score becomes the dependent variable that is influenced by an independent variable, the intervention. We have two separate groups, meaning that any member of a group belongs to one group and one group only. Either they have followed the conventional assignment system, or they have been part of the experimental group with no mandatory assignments. To analyze the differences between the two groups, the independent **t** test was used to look for significant differences in the mean of this gain score for each of the groups.

We have two hypotheses that are used, as explained earlier. The null hypothesis that there is no difference in improvement or drop of competence or motivation between the two groups. The alternative hypothesis that there exists a difference between the two groups.

The t-test is used to compare the mean of the change between the posttest and the pretest to look for a statistically significant difference. If we call our experimental group A, and the control group B - this can be done by using the following equation

$$t = \frac{m_A - m_B}{\sqrt{\frac{S^2}{n_A} + \frac{S^2}{n_B}}} \tag{4.2}$$

where m_x represents the mean value of the group X, and n_x represents the total number of participators in the group X. S^2 is defined as the common variance of the two samples and is calculated using the formula

$$S^2 = \frac{\sum (x - m_a)^2 + \sum (x - m_b)^2}{n_a + n_b - 2} \tag{4.3}$$

The sum will sum over all the gain score for all the samples in the two groups.

The level of significance can then be read out from a t-test table. If the t-test statistic value is higher than a critical value, the difference between the samples is significant. The lower part of the fraction in the common variances equation is called degrees of freedom and is commonly calculated using the total number of respondents subtracted by the total number of groups in comparison. Hence it becomes

$$d_f = n_A + n_b - 2 \tag{4.4}$$

as was also seen above in Equation 4.3

The computations of the statistical method will be done using the Stata software for statistical analysis [87]. The statistical process of performing the t-test is summarized in Figure 4.3

4.4.4 ANCOVA

Another approach to statistical analysis of two groups in the Analysis of Covariance called ANCOVA. The purpose of ANCOVA is estimating the difference between the groups on the

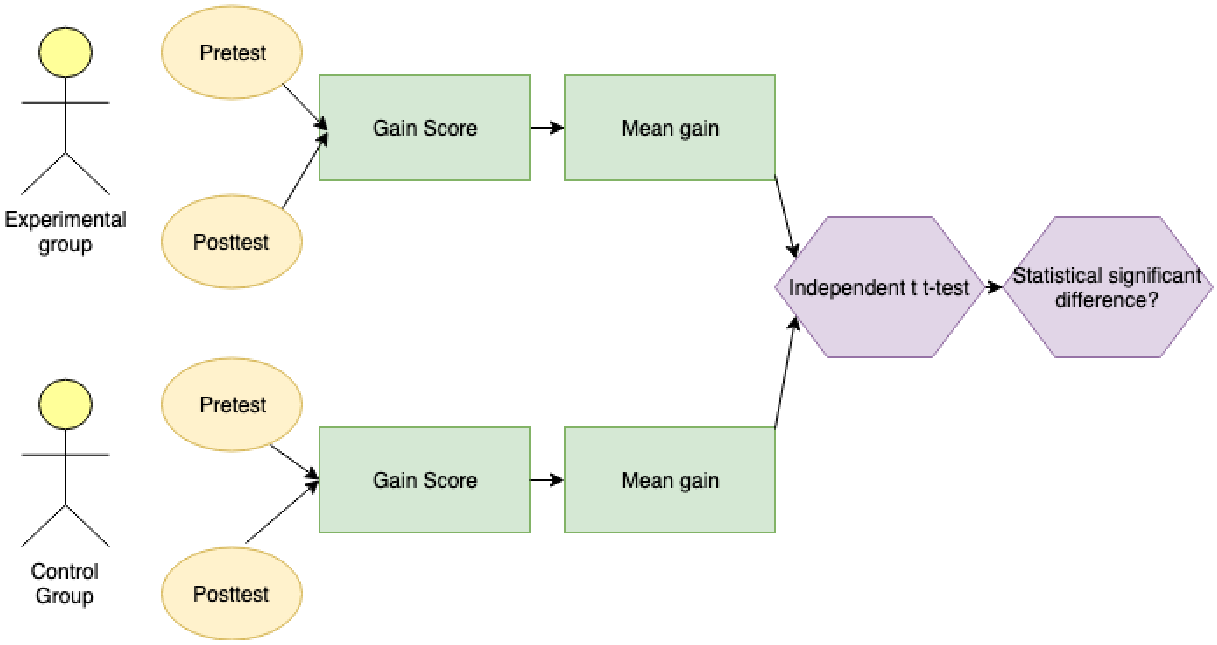


Figure 4.3: The process of statistical analysis when used with the independent t-test.

posttest, after having adjusted for initial differences in the pretest using a regression model. Since there is a Nonequivalent Groups design, the pretest scores have to be adjusted for measurement errors, in a Reliability Corrected Analysis of Covariance model, which uses a variation of ANCOVA [88].

The reason for the adjustment is that the ANCOVA model has a bias when working with non-randomly assigned groups. This is due to measurement error in the pretest, as well as initial non-equivalence between the two groups on the pretest. The less similar the groups, the bigger the initial difference. To fix this problem, new pretest scores are created for each person, adjusted for pretest unreliability, and use this adjusted score in the ANCOVA model. The formulas for this are given below in Equation 4.5 and 4.6

$$x_{adj} = \bar{X} + r(x - \bar{x}) \tag{4.5}$$

where x_{adj} is the adjusted pretest value, $\bar{x}$ is the group mean, x is the original pretest value, and r is the reliability of the group. The reliability score can be calculated in several ways. In this thesis, we have used Cronbach's Alpha [88].

$$y_i = \beta_0 + \beta_1 x_{adj} + \beta_2 z_i + e_i \tag{4.6}$$

where y_i is the outcome score for result i, β_0 is the coefficient for the intercept, β_1 is the pretest coefficient, β_2 is the mean difference for treatment, x_{adj} is the adjusted pretest score, z_i is the dummy variable for which group the person belonged to, and e_i is the residual for the result [88].

The ANCOVA model is a more advanced statistical model and was further used in the experiment, for verifying the initial results and the initial analysis, to find statistically significant differences between the two groups. The result of the ANCOVA model is a P-value, for signifying how large a probability that the results of this were gained by change and an adjusted R-squared value, explaining how much of the difference in the scores can be explained by the given variables.

The statistical process of using the ANCOVA model is visualized in Figure 4.4

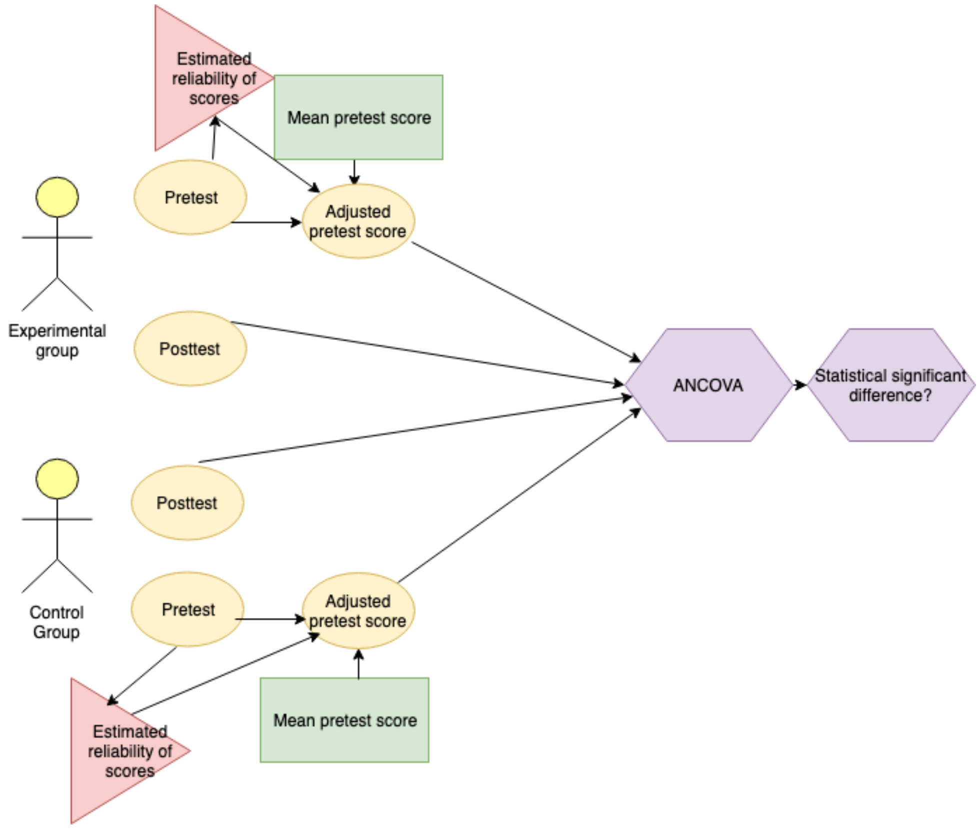


Figure 4.4: The process of statistical analysis when used with the modified verson of ANCOVA for NEGD.

4.5 Evaluation Approach

When analyzing quantitative data, the analysis starts when all data is collected. Only then can one start to go through the data and see what they could indicate [88]. In quantitative research, a statistical program, like Stata [87], is usually used to analyze the data that is collected by using the methods in Section 4.4.2. In addition, Microsoft Excel was used to calculate small data changes, like the adjusted pretest score, and sort the relevant data together [89].

4.5.1 Learning Outcome

Most of the experiments conducted from previous research, as discussed in Chapter 2, used results from mid-terms or exams to measure students' learning outcome after a semester. Some involved a test of the lecturers or researchers design in measuring the effect. In this experiment, there was not enough time to wait for the exam results, so learning outcome has been measured with a pretest-posttest design, with a posttest measuring learning outcome after the semester, and the pretest looking at fundamental knowledge. To deal with the quasi-experimental designs where everybody has volunteered, the learning outcome was measured through either the change of learning or a modified pretest. This is both due to the non-randomized population, but also due to the low number of participants, giving answers less statistical significance of an overall class distribution of grades. As naturally, they have learned much more during the semester, the second test was more difficult and involved more object-oriented programming principles than the first test. Hence the majority of students scored lower on the second test than the first test. Both these tests were corrected by me, using anonymized IDs that did not indicate to which group the writer of the answers belonged. The result of these tests was then done trough through statistical analyses to see if the data was statistically significant.

4.5.2 Motivation

Motivation is a highly personal opinion and is such a harder concept to measure that learning outcome. As described in Section 4.1 motivation was measured through online questionnaires, where the beginning motivation was compared to the end motivation rated by themselves. This is similar to how learning outcome was measured, only no validation or correction of the tests was done; the answers are already there as answered by the participants. The evaluation of motivation was also done through statistical tests to see if the data was statistically significant.

4.6 Considerations

4.6.1 Ethics

While writing this thesis, I have also been a lecturer in the course, with weekly assignment lectures explaining the assignments, and going through relevant theory and practice. This is a position I have had the last two years, while I have been involved in the course for the previous four years. While I, therefore, have strong ties to the course and might be able to see short-comings and improvements better than many others, I am also responsible for the assignments in the class, and may not be unbiased when it comes to predictions or conclusions. To avoid any prejudice against any persons or group towards the outcome, all material from the control and experimental group have been strictly looked at with only the ID of the participant. This made me unbeknownst which group belonged to before the end of the data collection and test correcting. This was to avoid any preliminary bias existing when discussing and analyzing the results. My supervisor has also followed along and read this thesis carefully, to see if any shortcuts were taken to reach a wrongful conclusion. My goal is to find out whether the assignments are helpful in the course, and I hope to improve the course and students education through this thesis and has such as only the hope of enhancing student process of learning programming at NTNU.

It is also essential to state that this is a voluntary study, where all participants volunteer and are free to exit the program at any time. The project has been reviewed to the Norwegian Centre For Research Data [90] and was approved by them. All participants in the study have been given a unique ID to identify their test results and weekly questionnaires. The link between their names and their ID are stored separately and will be deleted at the end of this research project. This was done because the ethics of separating a group of students to test a type of education scheme, without knowing the result of this, may cause them to to get either an advantage or disadvantage compared to the rest of the students. Because of this, any random selection in an educational research project should be very carefully considered, and voluntary participation was decided to be more ethical for this experiment.

4.6.2 Data Distortion

Data distortion is about the potential distortion of data. In a mix of quantitative and qualitative data, there are aspects to be aware of to be able to realistically and objectively analyze the data. According to Rogers and Preece, there is five concepts one needs to be aware of [91], as visualized in Figure 4.5.

Reliability Reliability describes how well the data can be reproduced if given the same circumstances. This highly varies depending on the circumstances in which they are produced. Controlled lab experiments have a higher order of reliability than observing students in their natural setting. This experiment is conducted in a natural environment, with students being able to perform their test and answer all questionnaires online and with no control. Due to this, it cannot be ruled out that some experiment participants may have communicated with each other during the semester, and affected each other study habits and results, both within and across experiment groups. This means that repeating the experiment might lead to vastly different results, mainly due to the low number of participants.

Scope The scope defines how we can generalize the findings of a study. Given that the experiment only focuses on introductory programming courses and students, it has been quite specialized already. Only students that are working on a computer science related degree have been selected for the experiment, which means that generalization to other degrees may not be feasible, as they may have lower motivation for learning programming.

The experiment has only been performed with one participating school, which means generalizing to other types of assignment systems, or other universities where students might have different mindset is not acceptable until a similar experiment has been tried out in other universities, as mentioned in Section 4.6.4.

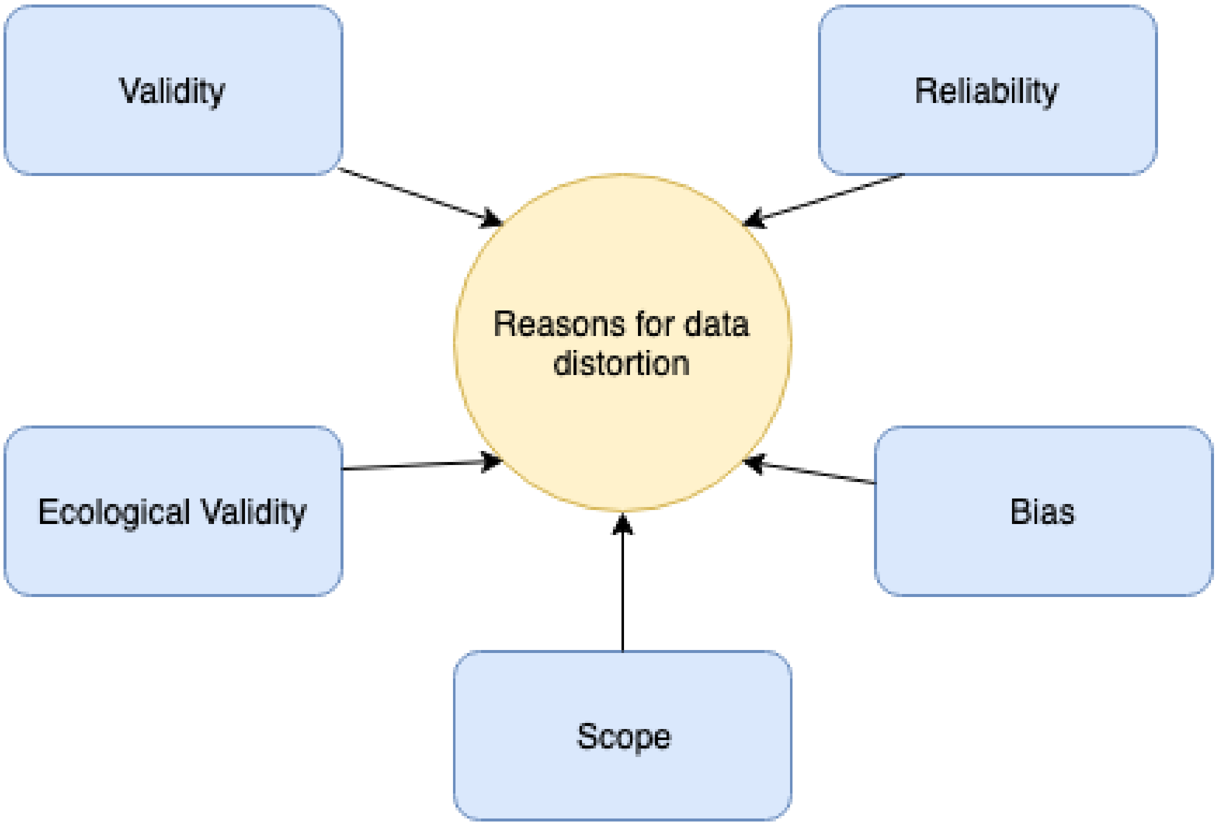


Figure 4.5: Different reasons for data distortion.

Validity The validity of a study describes whether the method used measures the intended purpose. This disqualifies quite many lab experiments, that is intended to test users behavior at home, for example. As there have been no carefully controlled environments in this experiment, and everything has been done in students own time, when they want to, this study has high validity. It should be noted, however, that due to students having many courses, the effect on this study from the other courses that the students are taking may have affected the result. This includes that students having mandatory assignments in other courses may be tempted to delay studying for the programming course until they are done with other subjects.

Bias The bias phenomenon may occur when the results may be distorted due to factors that were not known, or due to design flaws in the experiment. The researches themselves could also be biased, wanting a specific result. In all evaluation methods used, there is a certain bias in this experiment. For the pretest and posttest results, there could be bias in how the results are scored, and how much points are given to each test result. This is partially covered for by rating the students only with the unique ID, thus not knowing which group the student belongs to when grading the tests. There could also be other factors that are different to weigh in, like the mood, or mistakes done when weighing.

Other bias factors include how students respond to different questions or what they put into them. When answering questions about motivation, your mood, and motivation for this day may be a huge factor, especially if you are having a bad day. There could also be individual bias against answering the most extreme option on any questionnaires, as not to seem abnormal.

Table 4.3: Different triangulation methods, and their relevance to this study.

Type	**Description**	**Relevance to this study**
Method triangulation	The study uses two or more data generation methods	Yes, but only one for each RQ
Strategy triangulation	The study uses multiple research strategies	Yes
Time triangulation	The study has been performed at two different points of time	No, but encouraged
Space triangulation	The study has been performed in multiple countries	No, but encouraged
Investigator triangulation	The study is done by multiple researchers who then compares results	No
Theoretical triangulation	The study draws on two or more theories.	Focus is on Self-Determination Theory

Bias is tough to account for, and repeatably of experiments to validate results is the best way to account for bias.

Ecological validity Ecological validity is how the environment could affect the research performed and may influence or distort the results. Since all data collected is done by the user, there is a high natural ecological validity factor. This removes the factor that behavior may change when participants are aware that they are being studied.

As all results except interviews are collected in a natural setting for the participants, it must be said to be a real-world approximation, and the study has high ecological validity, as the stimuli of real experience are highly present.

4.6.3 Triangulation

Triangulation in research is a way of ensuring validity. This is done by using multiple methods, strategy, or otherwise to corroborate the findings presented in the research. There are numerous types of triangulation, as presented by Oates and summarized in Table 4.3. Their relevance to this thesis is described as well, as well as encouragement for further triangulation in further work relevant to this thesis. As can be seen, there are multiple ways to achieve triangulation, and using all of them is unfeasible for most research projects.

4.6.4 Repeatability

Being able to repeat experiments is vital for any research. Any research could be corrupted by unrecognized factors, and conclusions can not be drawn until an experiment has been repeated enough times. Repeating experiments with the same result ensures that other factors have not influenced the outcome, and the result was due to a fault in the equipment or measurements [86].

Due to this being one of the first of its kind in looking at mandatory assignments in computer science courses, as well as still demanding some meetings between teaching assistants and students. This means that the experiment has not been a repetition of something previous. To support the conclusions of this thesis further, this experiment should be repeated, both at the same university and course, and other universities with a similar structure of classes.

This is especially true when the experiments include a low number of participants and a mix of quantitative and qualitative data [86]. For readers of this thesis that is interested in repeating the research, it is highly encouraged. All questionnaires and tests used in this experiment can be found in the Appendix. All results are also available in the Appendix for repeated statistical analysis.

4.6.5 Practical Issues

The most significant practical issue for this thesis and experiment was to find willing participants for the experimental group and a control group. For the experimental group, the bonus of not having to do assignments should have been sufficient. The reason behind this could be various, some examples.

- Laziness, students want to put in as little work as needed to take the exam.
- Flexibility, planning to do the exercises but wanting to do so in own pace, instead of being focused on deadlines.
- Cleverness, thinking themselves particularly talented, and think the standard exercises will be too simple and boring. They want to spend more time on larger, more ambitious projects.
- Students that are wanting to feel special, being part of a small group that receives different treatment.

For the control group, however, some incentive was used for being part of such a research experiment, and gift cards were purchased to add as an economic incentive for participants. My role as a teacher in the course also helped in being able to inform well about the project, and recruiting willing participants during the lecture. Some students would not see it as a bonus to be freed of assignments. They could know from their self-insight knowing that they quickly end up procrastinating and do too little coursework unless they have some fixed deadlines.

4.7 Participant Recruitment

To be able to answer the research questions, and get the data needed for this thesis, voluntary participants had to be recruited from the course. About 700 students are taking the course each semester, and the hope was to get around 20 students in both a control group and the experimental group.

As explained in Section 4.6.1 I have been a lecturer in this course as well, so two of the beginning lectures in the course, about three minutes were set aside for informing about the project, and describing what participation would mean for the students involved. This was also informed about in one of the main lectures in the course. There were about 300 students present in each of the lectures. This was done to put a face on the person that they would be helping, as well as being able to answer any questions students might have, as well as objections they might have to take part.

To reach the students not attending the lectures, the course's annunciation web page was also used. This included a link to sign up for the research project, as well as a thorough description of the project, and what it would mean to participate. Stating that being part of the project is voluntary and that they could leave at any time was necessary, as described by Oates [86].

Chapter 5

The Experiment

This chapter explains the practical arrangements of the experiment, how the students were tested, and background information about the respondents.

5.1 Timeline

A timeline of how this experiment has evolved, and the different stages are shown in Figure 5.1

5.2 Respondents

The respondents were recruited at the start of the semester, as described in Section 4.7. The total number of students who delivered all material and did not opt out from the experiment was 40. Initially, 25 students delivered the pretest for the experimental group. However, only 22 students

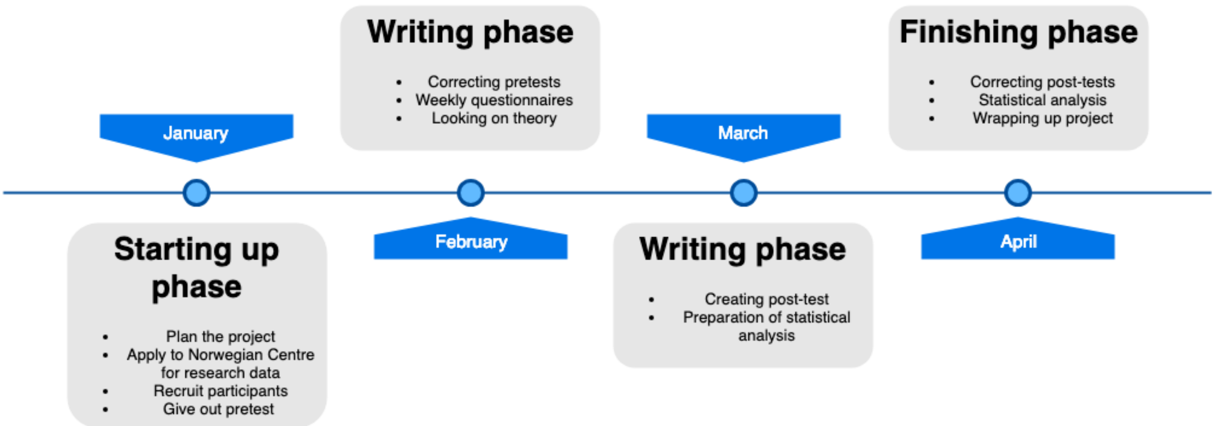


Figure 5.1: Timeline of different stages of the thesis.

Table 5.1: The distribution of the respondents in the different groups.

Group	Number of students
Experimental group	22
Control group	18

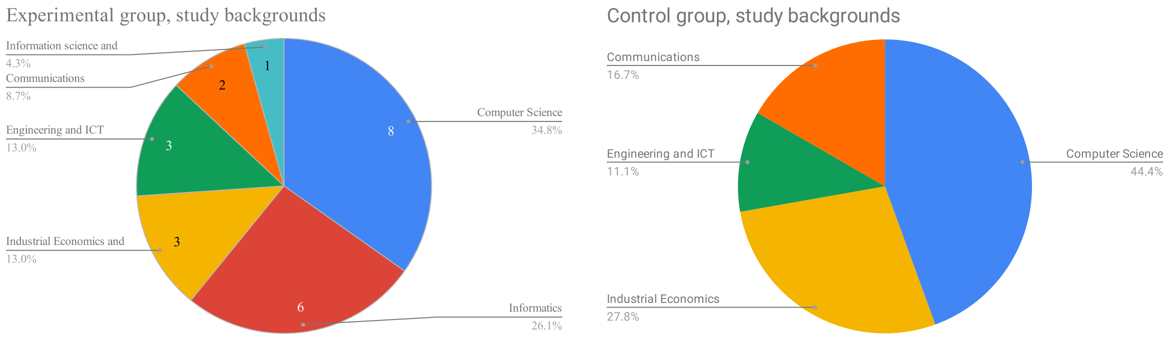


Figure 5.2: Charts displaying study backgrounds of student respondents.

delivered the posttest. The students' distribution over the control group and experimental group is seen in Table 5.1

The different study backgrounds of the students who have followed this program, both the control group and experimental group are displayed in Figure 5.2. Their distribution of grades in introduction to IT the semester before is also shown in Figure 5.3.

As we can see, the distribution is quite similar for both the experimental group, and the control group, both when it comes to studying backgrounds and grades. The studies represented are:

- Computer Science
- Informatics
- Industrial Economics and Technology Management
- Engineering and ICT

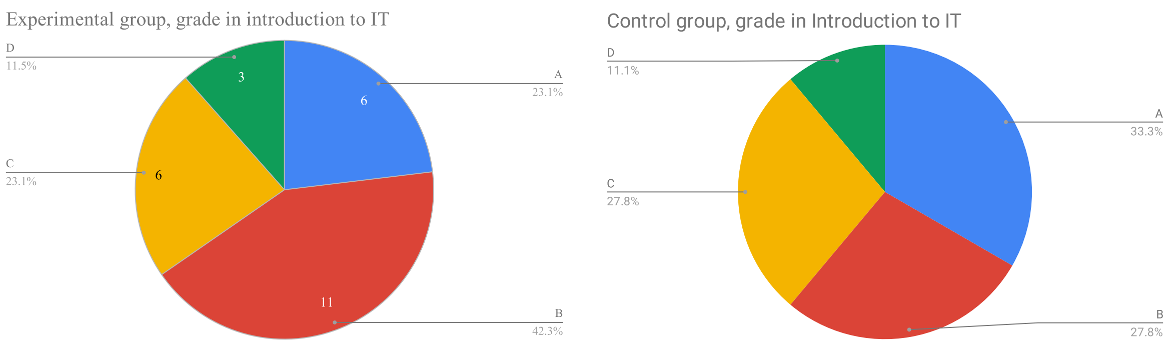


Figure 5.3: Charts displaying previous grade in Introduction to IT course for student respondents

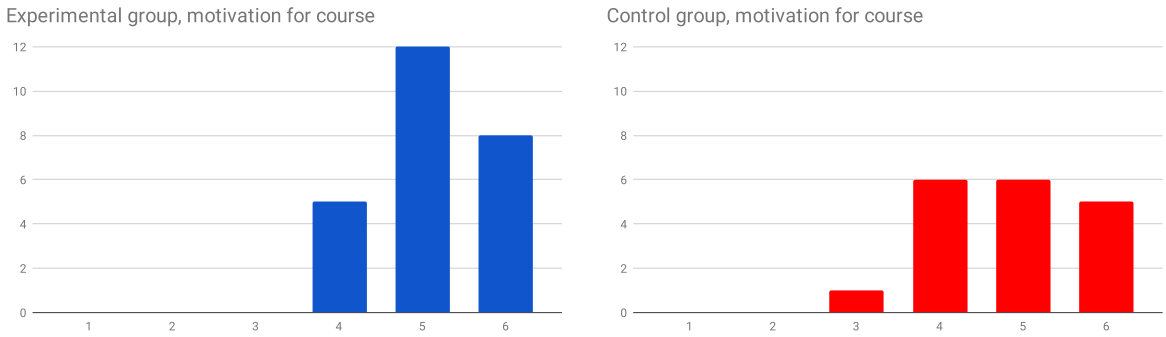


Figure 5.4: Charts displaying respondents motivation for the course at the beginning of the semester

- Communications technology
- Information science and ICT

All study backgrounds have strong relevance to IT and were selected to participate because of this. Two applicants to the experimental group were not admitted into the experiment, due to not having taken an introductory IT course before, as well as little relevance to IT for the study. The control group lacks participants from informatics, a group that is primarily represented in the experimental group. On the grade side, the control group slightly outperformed the experimental group on the last IT course, which can be expected for a group volunteering for additional tests. This was not looked at as a significant problem, as most grades were represented for looking at differences between weak and strong students. Also, the overall gain score or a modified pretest score will be looked at, instead of only a learning outcome which would have been used if only the posttest results were used.

For motivation, the results are quite similar as can be seen in Figure 5.4, with most students being at the upper end of the motivation spectrum.

5.3 Setup

This section explains the setup in how the different hypothesis from Chapter 4 were tested on the respondents.

5.3.1 Testing Learning Outcome

Learning outcome for the groups was tested with a pretest and a posttest, and the gain score was calculated. The pretest was distributed to the students in week 5, at the beginning of the semester. The students then had one week to complete the test. The test had the following requirements and solicitations.

- The test could be done at any time, at any location according to the students choosing.

Task 2 - The Country class.

2. Writing the **Country**-klassen. (30%)
   a. **Country** should have the following fields. **String** name, **int** inhabitants, **int** squareMeters.
   b. Create getters and setters for the fields in **Country.** Make necessary assumptions about validation. (5)
      i. **name-**field should only have characters, no spaces or numbers or otherwise.
   c. Write the function **immigration(int number**) that should update inhabitants with new immigrants. (5)
   d. Write the function **emigration(int number)** that should update inhabitants with people that have emigrated (5).
   e. Write the function **immigration(int number, Country otherCountry).** This function should get emigrants from **otherCountry,** reduce inhabitants there, and increase inhabitants in this country. (5)
   f. Write the function **union(Country country)**. This should return a new **Country-object,** with the sum of inhabitants and squareMeters of this country and the country-object. The name of the country should be the country that executes the method. (5)

Figure 5.5: An example of a task in the pretest.

- The test could be done in any order, with seven different tasks.
- There was a mix of theory and practical coding exercises. For the coding tasks using the internet to aid in solving the task was allowed.
- The students were told to sit for two full hours, without disturbance.

As the pretest was done at the start of the semester, there was a limited amount of Java and object-oriented Programming knowledge that could be tested. The test focused on object-structures, validation and encapsulation, and procedural Java programming tasks familiar to the students from the introduction to IT course, but with a new programming language. An example from the assignment can be seen in Figure 5.5. The complete pretest is available in Norwegian in the Appendix 8.1

The posttest was distributed in week 14, at the end of the semester, where students again had one week to complete the test at their own time, and with the same requirements as the pretest. The exception was that multiple tasks built on the same foundation, so one of the exercises had to be completed before the others. This time, the students were supposed to have learned

Task 7 - The observable Technique

1. Create an interface - TransactionListener - with the method: void listenToTransaction. Decide which arguments are natural for this method to accept based on the task.
2. Extend the **Bank-class** from task 3, to be able to register and remove listeners, and tell listeners about each a time a transaction has happened between two **Accounts.**
3. Create an implementation of **TransactionListener, InternalTransactionListener**, which should print to the console if the transaction has happened between two accounts with the same owner.

Figure 5.6: An example of a task in the posttest.

more during the semester. Therefore, more object-oriented programming and java techniques were tested. This included interfaces, like Comparable and Comparable, use of Java objects like Collection and Map, writing to and reading from file, advanced object structures, delegation, the observable-technique, and inheritance. An example from this test can be seen in Figure 5.6, while the full test in Norwegian is available in Appendix 8.2

5.3.2 Testing Motivation

The motivation of the students was tested by the student respondents own personal opinion on how motivated they felt for the course at the beginning of the semester and the end of the semester, along with the posttest. The students were given a survey where they could answer how motivated they were on a scale from 1-6 at both surveys. In addition, they were asked the question if they felt more motivated than at the beginning of the course. The surveys can be seen in Appendix 8.5

5.3.3 Interviews

At the end of the semester, after finishing the posttest, an interview was conducted with those who had been part of the experimental group and had been given no mandatory assignments.

This was conducted to clarify how they felt after having followed such a project, and especially how they approached learning the course with no guidelines and no compulsory assignments. Many of them had completed some of the exercises, and were asked why they had done so, and if they felt different than those who had to follow and do all the assignments. They were also asked what other types of resources they had used, as well as how they used the resources available in the course, like teaching assistants, wiki-page, and lectures.

The interviewees were at the start of the interview told that the meeting would be recorded and transcribed, but that otherwise no information about them, who they were or other, would be linked to the interview data. The data from the interview would also not be connected to any of their other results. The full interview guide can be found in Appendix 8.3

For the explanation behind the choices of data gathering methods, as well as problems that can arise because of these choices, see Chapter 4.

Chapter 6

Results and Discussion

This chapter presents the results of the experiment that has been conducted. First, the results from the tests, surveys, and interviews will be presented in their sections, followed by a discussion on the result and their impact. A final combined discussion on what the results can be interpreted as follows at the end in Section 6.5

6.1 Learning Outcome Results

Firstly, a difference in the pretest score between the two groups was noted, as can be seen in Table 6.1. This shows that the control group outperformed the experimental group on the pretest, indicating a slight edge to the control group individuals when it comes to previous knowledge of programming.

In the table we also see the average score on the posttest, noting a slight decline in both groups on the posttest, probably due to the higher difficulty or extended length of this test.

This resulted in average gain scores for the groups as shown in Table 6.1, calculated using Equation 4.1. As can be seen, the experimental group had on an average a higher gain score, meaning that they performed better on the posttest compared to how they did on the pretest compared to the control group.

The gain score of the two groups looked to be approximately normally distributed, visualized in Figure 6.1. A skewness score was also calculated [88], which indicated that the data was roughly symmetric. Therefore, a t-test could be performed to see if the difference between the

Table 6.1: Average scores and gain score for the control and experimental group.

Group	Average pretest score	Average posttest score	Average gain score
Control Group	61.17	53.36	-7.81
Experimental group	53.55	48.73	-4.82

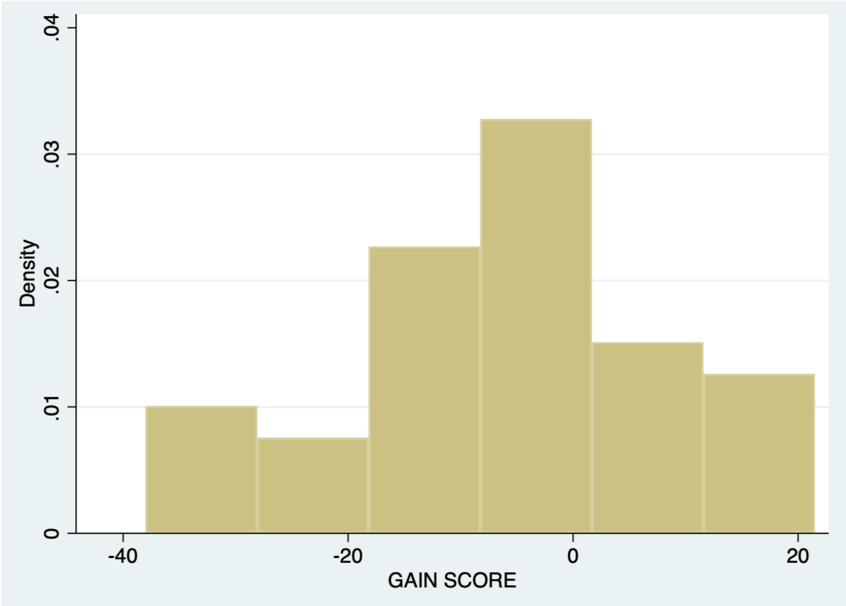


Figure 6.1: Histogram over the gain scores.

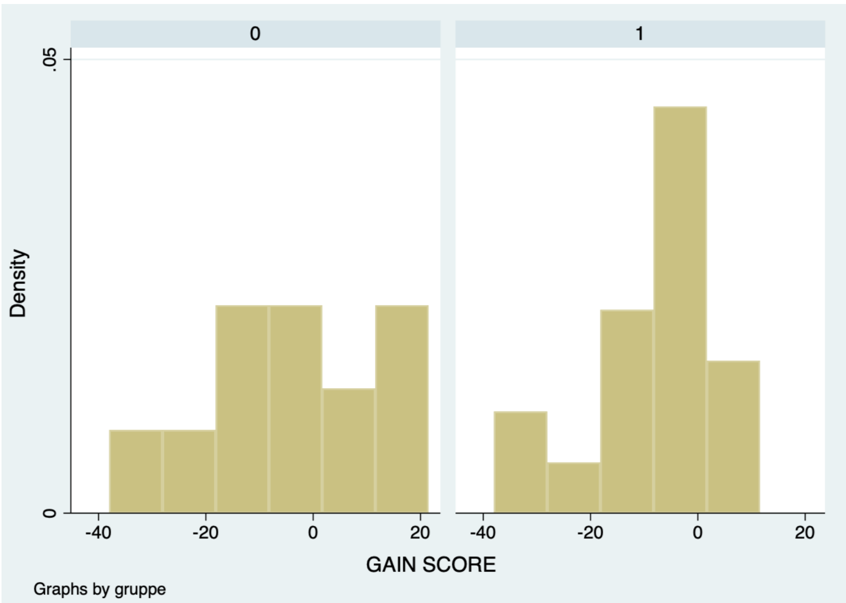


Figure 6.2: Histogram over the gain scores by group. 0 is the experimental group, and 1 is the control group.

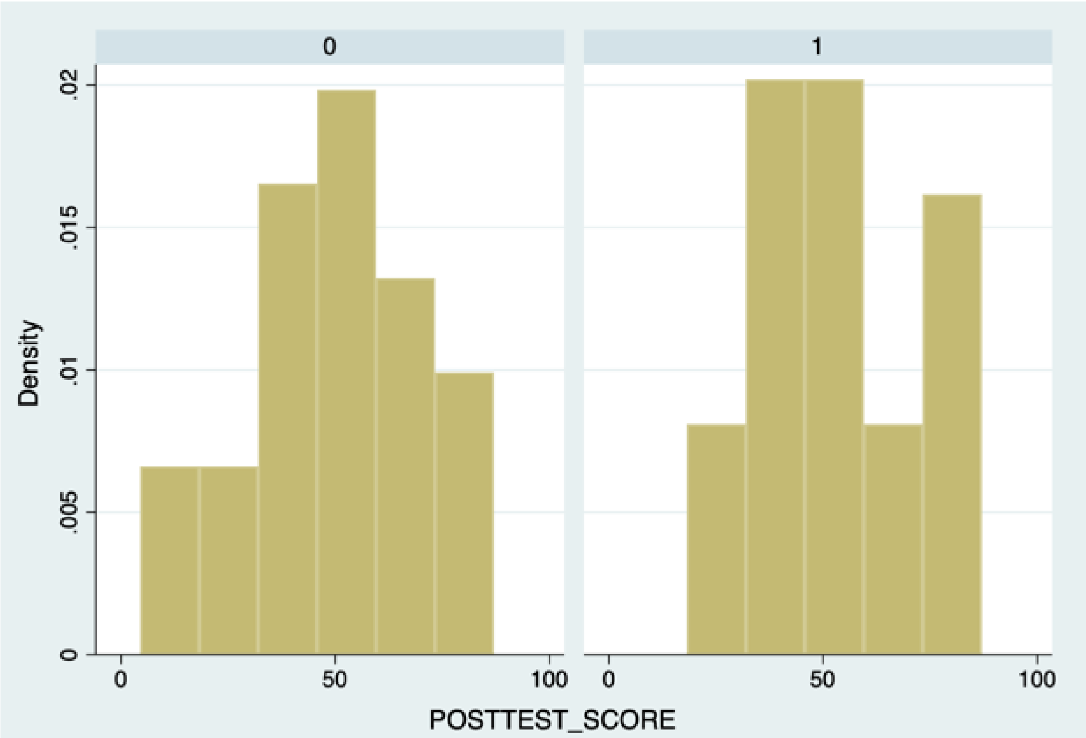


Figure 6.3: Histogram over the posttest scores, 0 is the experimental group, and 1 is the control group.

mean gain score of the experimental group and the control group had any statistical significance. Looking at the gain scores by which group they belonged to, we see in Figure 6.2 that the gain scores become less normalized. This could be due to the lower participant numbers and could have affected the t-test results. A histogram over the posttest scores for the different groups is also displayed in Figure 6.3. This is less normalized, noticing a skew towards the right for both groups. This made the posttest scores unpractical for statistical analysis with t-test and is why only the gain score is used by this test. ANCOVA does not use the assumptions that the data has to be normalized and is a better model for use with this data.

6.1.1 Running Statistical Models

The t-test was performed using the H0 hypothesis, that there is no improvement or reduction in learning outcome for students that do not have mandatory assignments. Running the T-test on the results of the gain score in Stata [87], yielded no significant difference for these groups, (P=0.51), with data visualized in Table 6.2. The means of both the control group and the experimental group were well inside the 95 % confidence interval of these two variables, mainly due to a high standard deviation of the dataset. The t-test tries to explain whether there is a substantial statistically probability that the dataset differs because of the independent variable, the different treatment in assignments that the groups had. Running this test gave the result of it not being statistically probable that the group variable could explain the difference. This implies there are not enough reasons to reject the null hypothesis and prefer the alternative hypothesis.

To verify the result of the t-test, a reliability corrected analysis of covariance model was run, as

Table 6.2: The results of running a t-test on the means of the gain scores.

Group	Observations	Mean	Standard Deviation	95 % confidence interval	t	P
Control Group	18	-7.8	11.6	-12.2 —- 2.5	0.67	0.51
Experimental group	22	-4.8	16.6	-13.6 —- -2.0		

Table 6.3: The results of ANCOVA on a model with group and adjusted pretest score

Model	P	Adjusted R-value
Adjusted pretest score	0.00	
Group	0.67	0.45
Group & Adjusted pretest score	0.76	

explained in Chapter 4. First, the reliability was calculated using Cronbach's Alpha, giving a reliability score of 0.817.

This reliability was used to calculate adjusted pretest scores, per Equation 4.5, for feeding into the ANCOVA model. This yielded, like the t-test, no statistically significant differences for explaining the posttest scores based on the pretest score and the group (P=0.77, adjusted R = 0.45). The R-value comes mainly from the adjusted pretest score, being responsible for explaining 45 % of the differences in the posttest score. The results from the statistical analysis can be seen in Table 6.3. To summarize, the ANCOVA model also indicated that there was not enough evidence to reject the null hypothesis.

The full table of results on the posttest and pretest can also be seen in the Appendix 8.4

6.1.2 Other Models

To see if any other factors could have played out on the results, several statistical tests were conducted with new models. Among others, an analysis of whether the mean of the groups, when looking at the previous grade in Introduction to IT. The mean results of these groups shown in Table 6.4

The result here is exciting, although a statistical t-test showed no significant statistical find, giving P-values as seen in Table 6.5

Table 6.4: The mean of the gain scores when also compared to previous grade in introduction to IT.

Grade in Introduction to IT	Experimental Group	Control Group
A	-13.1	-8.58
B	-6.35	-2.9
C	-0.75	-7.2
D	8.67	-19.25

Table 6.5: Results of running the t-test on the gain score when divided by group and grade in Introduction to IT.

Group	Observations	Mean	Standard Deviation	95 % confidence interval	t	P
Students with grade A in introduction to IT						
Control group	6	-8.6	14.7	-24.1 —- 6.9	-0.52	0.61
Experimental group	5	-13.1	13.6	-29.9 —- 3.7		
Students with grade B in introduction to IT						
Control group	5	-2.9	6.1	-10.5 —- 4.7	-0.38	0.71
Experimental group	10	-6.4	19.7	-20.5 —- 7.8		
Students with grade C in introduction to IT						
Control Group	5	-7.2	6.6	-15.3 —- 0.9	0.99	0.36
Experimental group	4	-0.75	12.8	-21.2 —- 19.7		
Students with grade D in introduction to IT						
Control group	2	-19.3	22.3	-219.4 —- 180.9	2.23	0.11
Experimental group	3	8.67	5.8	-5.7 —- 23.0		

New models were also run with the modified ANCOVA model to see if any other variables better could explain the difference in the posttest score. The previous grade in Introduction to IT was encoded into two groups of high-performing (A and B) and lower-performing students (C and D), to see whether this variable better explained the differences. This yielded approximately the same results as before, with the adjusted pretest score still being mainly responsible for explaining the difference, although now with an adjusted R-value of 0.47 (P=0.79). The same result, with an even lower adjusted R-value (0.36, P=0.84), was the result of running the model with motivation change instead of the previous grade.

To summarize the results, there was no indication that any variables, outside of the pretest score, could explain the differences in the posttest score in any significant way. No statistical models gave evidence to reject the null hypothesis.

For further work, the exam results of the two groups can also be compared, which is elaborated upon in Chapter 8

6.1.3 Discussion

The results that yielded no significant change in learning outcome between students that do mandatory assignments and students that are given autonomy and freedom are interesting. Exercises are there to force students into learning and working evenly throughout the semester. However, if there are no apparent benefits to them, why do we spend so much time correcting and making sure the students do the assignments?

A formative assessment from assistants giving feedback to students who want to learn is well known for providing excellent results for students that are open for feedback, as described back in Chapter 2. However, students that are not open for feedback are spent many resources on checking whether they have done the assignments. This could be resources that are better spent on more receptive students, focused on teaching the students what they need to learn when they are open for learning it, instead of a fixed schedule for every student, that does not provide any autonomy.

Assignments are an excellent way to see whether someone has learned something, and to following alongside similar tasks as will be given on the exam, as that is how the final grade will be given. The results here, however, that measuring whether students have done the assignment is not more helpful than merely helping them alongside with the assignments or their projects, letting them learn however they like.

As for the entire experiment, conducted with such a low number of students, there are significant reasons why the result could be as it is. The students following alongside know full well that they work best given autonomy, and therefore signed up for the experimental group. The control group, while given the option of freedom, chose to follow alongside a strict schedule. There are of course outliers here, with the probability that several of the participants signing up for the experimental group because they did not want to do assignments, and wanted to have more free time and do less work throughout the semester. There are individual differences between students, and some need to be pushed, while others work best with autonomy and freedom, which will be more explored in Chapter 8. This bias of a low number of participants was elaborated upon in Section 6.5.1.

The results looking at the difference between the grades is particularly interesting, even though there were not a statistically significant enough difference. The idea beforehand, was that more autonomy and more freedom would be better for the high performing students, which manages to learn on their own, and are not in a significant need for guidance as the others. However, the result indicates the exact opposite, with A and B students in the experimental group getting outperformed by A and B students in the control group, and the opposite for C and D students. Contrary to popular belief, that may mean students that do not perform as well, may not require appropriate guidance, but rather need autonomy to work at their own pace, instead of being forced through a specific set of assignments. It could also mean that weaker students cheat more on the assignments, as they are unable to do them, and when given more autonomy feel the need to complete them without having the pressure of a deadline. This result is also somewhat more consistent with the findings from Haugan and Lysebos study, where the weaker students in the pretest, did even better on the exam [18]. It should be noted that the number of observations within each grade is very low, and a higher number of participants is needed to get a more meaningful result. It may also be that stronger students attribute more of their learning to the exam period, and learn more in a shorter period, and therefore have delayed more of the work until the end of the semester.

When discussing weaker and stronger students, it can also be discussed to what extent resources should be spent on students. In the current assignment system, all students have to meet their teaching assistant to demonstrate their code and understanding of this to the assistant. They may meet up as often as they want to get help in understanding the assignment and complete it. With resources that could focus less on approval of exercises, but more on teaching and guiding students, resources could be further utilized by the students that need them. Some mechanism might also be in place to get the weaker students to use the available resources.

There will always be students that do not work, when not giving strict guidelines for what to learn, and when to deliver. The discussion must be whether it is more important to provide more autonomy to the students who want it than to force everyone through the same mandatory arrangement.

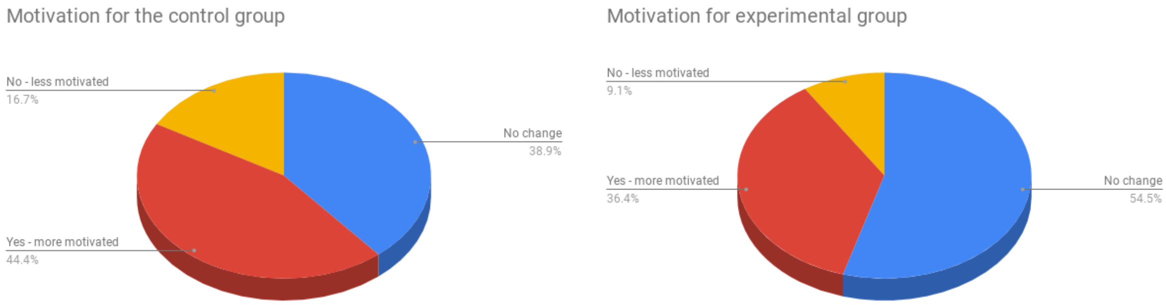


Figure 6.4: Charts displaying results of motivation for the two groups.

The results found in this experiment is consistent with some of the results from the literature in Chapter 2, that monitoring assignment completion do not increase learning outcome [41, 50, 51, 53]. These studies also showed no statistically significant difference for similar experiments, with some students having compulsory assignments, and others given more autonomy.

Given that assignments, or at least mandatory assignments, does not seem to be any help in students' learning, the focus onward should be on how students learn, and what is the best way to aid in their learning process. Previous work in teaching computer science curriculum from Chapter 2 can be used. Most important is how to use formative feedback effectively as a tool for guiding students. Students learn in different ways, as illustrated in the taxonomy from Section 2.2.1, and has a variety of different learning styles. A more autonomous approach to teaching computer science could help students to find a way that best suits their particular learning style and their approach to acquiring new knowledge. The assignment system that is used today may be well suited for active, sensing, and sequential learners. It does not cater to the needs of global, intuitive, and reflective learners, who needs time to think about concepts, and orient more toward theories. The students approach to learning could be helped along by various exercises or assignments, be them mandatory or not, to guide in this process. The choice of method could be exercises, group projects, pair programming, or other practical tools for teaching computer science. However, if only given compulsory assignments, that will not leave room for self-study and for learning styles that are not aligned towards exercises as a learning activity.

6.2 Motivation Results

As described in earlier chapters, results about motivation were received from a survey at the end of the testing period, where the students could choose whether their motivation was higher, lower, or the same as when they started the course. The results for the different groups can be seen in Figure 6.4

For statistical analysis, we wanted to test the null hypothesis, that there is no improvement or reduction in motivation for students that do not have mandatory assignments. Converting the results from Figure 6.4 into numeric values, 1 signifying a decrease in motivation, 2 meaning

Table 6.6: The results of a t-test for comparing the means of motivation results for the control and experimental group

Group	Observations	Mean	Standard Deviation	95 % confidence interval	t	P
Control Group Experimental group	18 22	2.27 2.28	0.75 0.6	1.9 —- 2.6 2.0 —- -2.6	0.023	0.98

no change, and 3 symbolizing an increase in motivation, a statistical test could be applied to the encoded value. This was done and used the Stata [87] software program for analyzing mean motivation change based on the two groups. The results can be seen in Table 6.6. The null hypothesis was that these two groups were equal, and the results yielded absolutely no statistical difference between these two groups (P=0.98).

As mentioned in previous chapters, the students also rated their motivation in a number between 1-6 at the start of the semester, and at the end of the semester. Interestingly, some oddities in the results here appeared, with numerous students rating their motivation higher than before, yet with a lower number. More inconsistencies with students less motivated rating their motivation with a higher number was also present in the dataset, making it apparent that the students did not remember their previous motivation rating. Regardless, a t-test was run on these numbers as well, trying to see if any statistically significant difference between the means of these number was present, but there was not enough evidence to rejected the null hypothesis using this data as well (P=0.58).

As a final model, an ANCOVA model was run on the data as well, trying to see if a mix of previous motivation and group was statistically significant for the current motivation. The result indicated that prior motivation was responsible for 43 % of the change (R=0.43, P=0.0001), while the combination was not statistically significant (P=0.77).

6.2.1 Discussion

The results for motivation were surprising, considering Self-Determination Theory as described back in Chapter 2. What was expected before the experiment was started was that motivation result for the experimental group, that were given autonomy to approach the course however they liked, would have a higher increase in motivation, however that was not the case. A range of different explanations for this is possible:

The control group had quite high previous motivation status, and with no random selection, this group could be among the highest motivated students in the course, able to keep their motivation high throughout the course as they liked the course and the assignments they provided.

Many in the experimental group had also done the assignments throughout the course, as they did not want to fall behind and miss learning the same things as the others in the class, as can be read more about in Section 6.3 and 6.4. This fear may have caused a lack of autonomy, where the students felt they had to do the exercises anyway.

There is also the added factor that not full freedom was provided to the students. They still had to meet up and show that they were following along and was learning, and this added mandatory session could have reduced motivation for this group as well.

With such a low number of participants, while the results indicate no change in motivation between the groups, as discussed several different options could be an explanation for this. Of course, it could just as well be that the freedom in choosing assignment gives the autonomy to provide a higher degree of motivation, thus yielding about the same results for the different groups, contrary to the results of Christensen [48].

It could also be that the other factors of Self-Determination Theory outweighed the added autonomy feeling that the experimental group was given. They could have missed out on relatedness, feeling outside from the rest of the class who did the assignments and had a common goal to work towards. They could also have felt a bit lost, missing out on increasing their competence. The control group could be quite sure that the assignments, created by the teaching staff, taught them the curriculum that was needed for the exam. Approaching the course through other online object-oriented programming courses did not provide the same assurance. As such, the other two components of intrinsic motivation could have balanced out the added autonomy, to give similar results in motivation. The extrinsic motivation of wanting to get a good grade on the exam could also have affected these results.

A consideration that could be contemplated is that the way we measure motivation is skewed, and it was not scientifically based motivation survey, but purely based on the students own experience in whether they were more motivated now than when the course started. The results that the experiment indicates is hard to argue for further actions, due to that that either approach seems to work for the students' motivation. Further work is necessary to study students' motivation in context with compulsory coursework.

6.3 Interview Results

Some interviews were conducted with the experimental group to get some qualitative data insights into how the group experienced having no mandatory assignments, and how they felt about the exam and how prepared they were. Ten interviews were held at the end of the experiment, one of these were conducted with just text responses, while nine were held in-person. These interviews were transcribed, and their results are presented here. Interviews were only done with the experimental group, as it was their experience of having no assignments that were the focus. To present the interview results, a summary of relevant findings will be presented, as well as some relevant quotes for each finding. As mentioned in Chapter 5 the full interview guide can be seen in the Appendix, 8.3.

- A majority of the experimental group followed wholly, or partially along with the assignments that the rest of the class did.

> *I was unsure whether what I did covered everything that we had been trough on the curriculum, so I also looked at earlier assignments to see whether there were items that I had not learned well enough.*

- Many felt that it was easier to deprioritize the course when they had other classes with deadlines coming up.

There have been times where I have not worked with Java in a week because I have done other things. Then I work more next week. This has caused me to not work as evenly as I could have done if it was mandatory.

I noticed that it was much easier to relax on this type of arrangement compared with those who followed the ordinary assignments program.

- The lack of mandatory assignments made it more fun to work with this course than to work with other classes.

It will be easier to deprioritize it, but this course is also what I bring out when I have time to spare. Because I'm a little inspired by the fact that I don't have to do it. Then I get some boost by doing it when I have time.

It has been inspiring to do another type of assignment scheme. I have had to work differently, more independently, and have taken responsibility myself, and I react positively to that. I get to decide for myself how I want to learn and what to learn.

- Many said something alongside, ”it works for me, but not necessarily for everyone”, meaning someone always has to be pushed to do something, with deadlines.
- Many felt that the biweekly meetings with teaching assistants were a good thing to force themselves to work and have something to prepare for these meetings.

Mandatory attendance with the teaching assistant has been very practical. Otherwise, I probably would have delayed things more, have something firm to work towards.

- A number followed web-based courses to learn themselves the curriculum. Most of these courses were based on small videos explaining a subject and many practical assignments. Many of these felt that they were unsure whether the courses fulfilled the curriculum, and such ended up doing more work by looking at the exercises as well.

I ended paying for Udemy's Java course and started early on working with this. The first parts of the course were a lot like introduction to IT with procedural programming. I spent a lot of time on basic things like variables, operator, and expressions and felt that when attending lectures that I quickly was falling behind doing this.

- Having to focus on the learning goals, and not assignments, meant they focused more on what they were supposed to learn, and not just passing tests.

I have completed the assignments to learn something, not because I have to. I think I have learned more by that, and it has been more motivating and fun to work with the course

I've looked more at the learning goals of this course

- The interviewees mostly agreed that the motivation was high. However, it had gone up and down during the semester. Especially when other courses were deadline heavy, motivation to work with Java was lower.

My motivation has been perfect all the way. It's freedom under responsibility, but I feel that for my part, it has been very good because I feel I can structure my weeks freely.

- Most were happy with not having to deliver any assignments

I thought it was a good thing that the assignments were not something that you had to do. You could choose more freely what to work with.

It feels like the system of most courses is rigged for everybody to come through at a mediocre level. There is no room for an adapted arrangement, regardless of whether you are good or bad.

6.3.1 Discussion

The results from the interview summarize most that can be discussed regarding this. That the majority of them were happy to be free from mandatory assignments, and that they felt this fitted better to their learning style comes as no surprise, as the selection to the experimental group was not random. They chose for themselves to follow an alternative path to prepare for the exam.

It is interesting that many chose to follow the assignments, even when not having to do them. It is noted that those who decided to do so, did it because they wanted to be sure they learned everything that was related to the exam, and not miss out of anything. This shows a considerable focus on the exam and the grade that is given there, while not the most important for a university to teach. The university wants students to have learned the learning goals of the course, and the exam is a summative way to measure that. Many things in a course are not asked about on the exam, due to time or practical constraints, and students choose not to focus their time on such knowledge. This is also consistent with the findings from Millers' study, where students still delivered the assignment when only a select few were graded [53].

Many noted that it was more fun to do the assignments when they did not have to do it, and did not have to complete everything, but rather focus on the learning goals. This is what professors also want to achieve with assignments, to focus on learning goals, and that the students have learned something, not just performed a task successfully. Their biweekly conversations with teaching assistants also achieved a more formative feedback session, where they focused on whether something was learned, and how the student could improve. This session should be further explored in further work to see how students could benefit most from a session with an experienced student.

The students saying about their motivation goes along with the theory of Self-Determination. That autonomy in how to learn and approach your task is vital for intrinsic motivation. Also, competence, another of the essential factors in SDT, is an important factor, where the students get to both use their competence in programming however they like, and increases their competence, as learning is the ultimate goal of any course. Relatedness, as mentioned back in Chapter 2 has not been the focus of this thesis, but is important not to miss. A critical find here is that some of the interviewees mentioned that they did the assignments because that is what their friends did. This is a weakness with such an experiment, where students are divided from the main bulk of the student population and are naturally drawn back to what the other students do to learn the course. Humans flock together and want to do the same activities, and to get a more viable result, one might consider experimenting on a larger scale, to get a larger group of students to work together and maintain the vital relatedness of Self-Determination Theory. While the motivation results from Section 6.2 argues that motivation is the same for the control group as well, the result from the interviews does not give any disadvantage to students' motivation when not giving assignments.

Table 6.7: Average hours spent per week on the course.

Group	Average hours per week
Control Group	7.0
Experimental group	5.5

6.4 Survey Results

This section discusses other results that resulted from answers to weekly surveys, and the last survey given to both the experimental and the control group. As verified in the interviews, a majority of the experimental group chose to do the ordinary assignments as well to learn object-oriented programming. This is visualized in Figure 6.5.

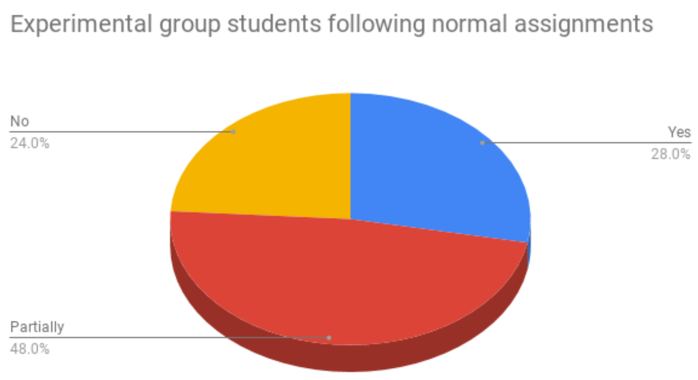


Figure 6.5: The answers to the question: Have you followed the ordinary assignments in the course even though you did not have to.

The students also reported weekly how they approached the course, what they had for plans to learn it next week, as well as their time usage this week for the course. This yields interesting results, with the average control group student spending more time on the course than the average student, even though the analyzed learning outcome was similar. This difference in average time spent can be seen in Table 6.7 and in Figure 6.6. This is inconsistent with the result from Haugan and Lysebo, where students used more hours on average when given no compulsory assignments [18]. In their study, they replaced obligatory assignments in almost all courses, and that could have affected the results, as total hours of school work has not been monitored. It could be that the students in the experimental group spent more hours on their other courses compared to the control group.

As a final remark, the students of both groups were asked the question, in your own personal opinion do you believe assignments could be given out, but not collected. The answers to this question were very different between the two groups, as can be seen in Figure 6.7

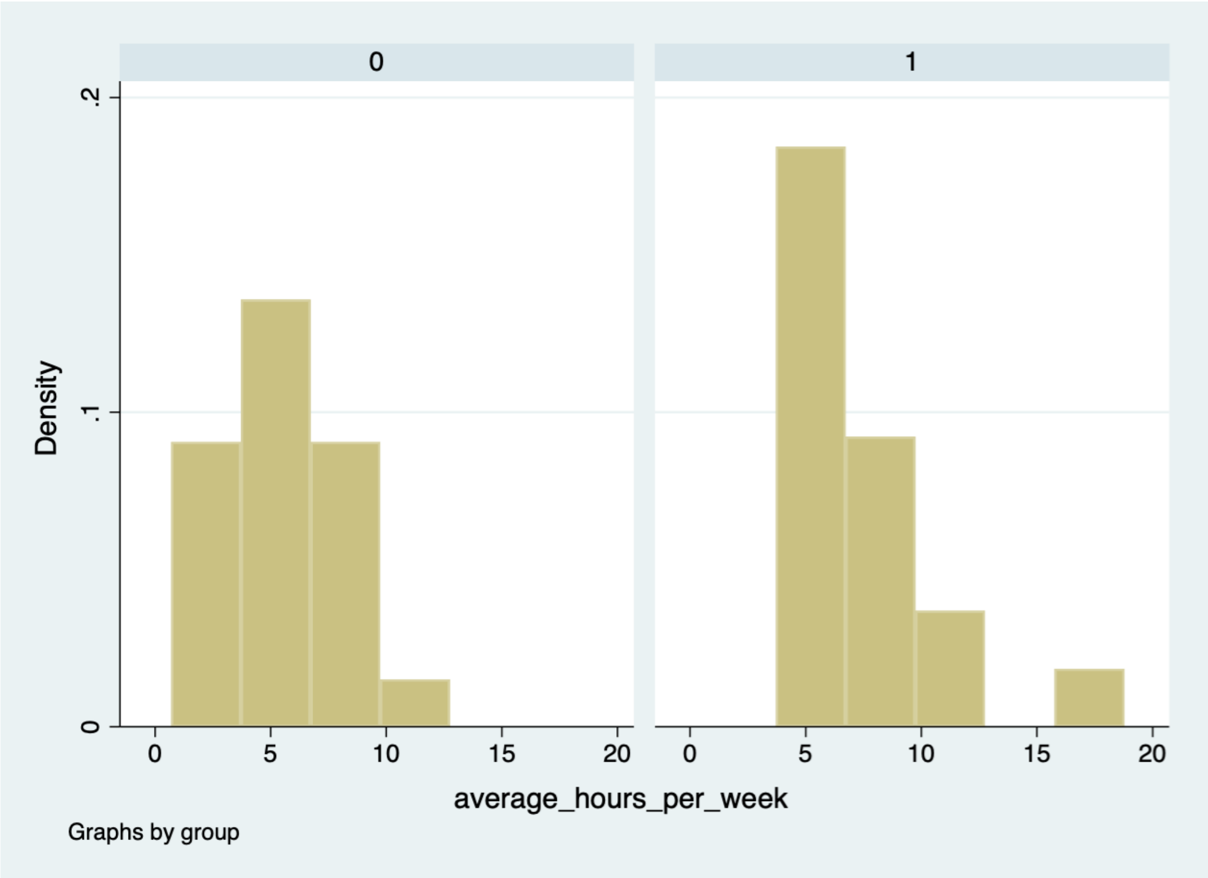


Figure 6.6: Average hours spent per week on the course in a histogram by group. 0 is the experimental group, 1 is the control group.

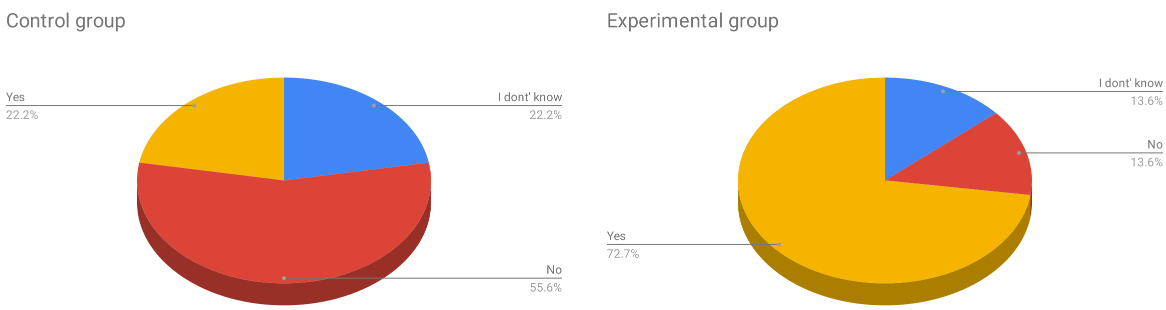


Figure 6.7: Charts displaying the answer to the question, Do you believe assignments could be given out, but not collected.

6.4.1 Discussion

The data from other surveys and weekly surveys have been interesting to look at and lays a groundwork for further insights into these, as will be mentioned in Chapter 8. It is interesting to see that many students in the experimental group chose to follow alongside with the assignments. As elaborated upon in Section 6.3 this is most likely due to them not wanting to fall behind, and miss any critical knowledge that the other students gained from the assignments. It is also a safe way to learn the course, while not being forced to pass them in a detailed schedule, but focusing on learning from them in your own time, when you are free to work on it in your own pace and not stressed by a deadline.

Time spent per week gives another insight into the data, where the students in the experimental group on average worked less than the control group, but had on an average higher learning outcome, even though the higher learning outcome was not statistically significant. This may be because the control group had to do enough assignments that were the demand for each assignment, while the experimental group only had to do enough to feel that they had learned the subject. It could be that the exercises demand too much from the student, without the learning benefit being higher after a while. Mapping the average hour to how they approached the subject, doing assignments or following other courses has not been done, but may be done in further work to see whether that may have had any effect on the results. It is noted that both the control group and the experimental group reported fewer hours per week on the course than what is expected by NTNU. This could be due to a misunderstanding of the survey, where not everyone reported the total number of hours spent in lectures. Those that reported they were in lectures, but not how many, were added three hours to their count for that week, equally distributed between the control and the experimental group. The fewer hours indicate that the burden of the assignments may not be a significant problem, as there is enough time set aside for self-study even when doing the exercises. Without data for hours spent on the other courses, it is hard to see whether that may affect the hours spent on this course.

The answer to the question of whether assignments could be collected, it is interesting to see the difference in responses, although not necessarily surprising. An interesting observation is that most of those who answered no in the experimental group also had responded to no to whether they had partially followed the ordinary assignments in the course. This could mean that they felt they missed something by not doing the assignments that the rest of the students did, or that they had not worked hard enough throughout the semester and wanted exercises to push them into working.

6.5 Combined Discussion

The results of this experiment have indicated there is no statistical difference in motivation or learning outcome for students having mandatory assignments, and for students having no mandatory assignments, but formative feedback sessions with their teaching assistants. Interviews and surveys have also indicated that students being released from deliverance of compulsory assignments will do the assignment nevertheless. They do so because they want to learn the subject and prepare for the final exam. It is unsure whether they would have done that if they knew that their classmates not necessarily had done these assignments. It could be that when they knew everyone else had to do these, they were afraid of falling behind. Whether they did exercises or not, the result indicated that the average time spent on the course per week was less for the students not having to do assignments, even though they achieved the same learning outcome.

Going back to why we have mandatory assignments, there were mainly two reasons. One is forcing students to work evenly throughout the semester, and guiding them in what part of the curriculum they should have gained an understanding of at any given time. Secondly, tests are used to test specific parts of the curriculum that are unpractical, due to time or resources, to test at the exam. It is hard to let go of mandatory assignments, as still these parts would have to be tested somehow. When it comes to the first reason, this is just one of many possible options to teach students the material and to help them work. While assignments can be beneficial for many students, there is no appropriate documentation that they are helpful for everyone, and lots of resources are spent on testing whether the students have done them. This also adds extra stress for the students [92], who must go from deadline to deadline to complete an assignment. Freeing students from thinking about what to deliver to a deadline, may make them more subject to thinking about what they should learn in any given week. Focus on what to learn instead of what to complete shifts the focus to what is essential for both professors and students alike, and if the admittance of mandatory assignments as a failure can help in that regards, it should be seriously considered.

As multiple studies pointed out, the summative feedback of delivering homework or exercises does not give benefits for the students, and the results of this experiment support these statements. Assignments are a helpful tool for preparing students for the exam, guiding them into learning more about the curriculum of the week, and measuring their progress, but the assessment of the exercises does not necessarily benefit the students. It is interesting that a majority of the students in the control group believed otherwise, and that should also be taken into consideration before launching an all-out experiment testing such an arrangement.

We can not skip over the fact that surprisingly, motivation was not higher for the students not having the assignments. As pointed out in Section 6.4, this may have been to various factors, especially the fact that the control group could have been particularly motivated. The fact that the exam was quickly approaching when the survey was sent out could have also been an added factor to affect these results. The result of this is hard to conclude something by, as we can not indicate that intrinsic motivation will be higher when students are given more autonomy to work in their way. Further research into this is suggested with a larger number of student participants, as well as modifying the experimental group to give them even more autonomy. As they had a particular set of learning goals for every other week in this experiment, they could instead have been given the complete set of goals, a guide for how to approach it, and a set a formative feedback session with the teaching assistant to asses their learning. This session could guide them into further action and learning approaches. This is further explored in Chapter 8

This experiment has been conducted on students from different study programs. All study programs have a high focus on computer science but are built up in different ways. Informatics has a high degree of freedom when it comes to choosing the courses that one wants to focus on, while computer science has a more fixed structure with courses that you have to take. Different study programs may learn and be motivated by different things, and this is important to keep in mind when designing a class. Students from different study programs may have a different learning style, while the same can also be said of students from the same study program. Designing a university course for different learning styles means having to give up inflexible systems for adaptable ones. Lectures could benefit intuitive learners, getting concepts explained, while sensing learners need to get their hands dirty and do something practical. Being aware of these different learning styles means we can design a schedule where a student can pick what activities are best suited for their particular learning style, while still providing them with the resources they need to achieve their best.

The most important part about this is to give computer science students the best tool and guidance for education, to educate the computer scientists of tomorrow the chance to excel in their field when they are finished with their studies. The world needs technologists in the future and will lack as such, and educational institutes should take their part when it comes to finding the best possible way of teaching computer science, and make sure the least amount of students do not drop out off their education.

6.5.1 Bias and Threats to Validity

Due to the quasi-experimental nature of the experiment, the small number of participants, and a variety of other factors, many biases could have affected the results of this experiment, some of which will be mentioned here, and some which are mentioned more thoroughly in previous chapters.

Students may be colored by their experiences with other courses, and their extensive use of mandatory assignments in other classes parallel to the trial in this course. They may thus be tired of deadline sprints and give a more positive review to different types of learning approaches than what they usually would do.

Students that have signed up for the control group have chosen to not sign up for the experimental group, and have as such chosen to do the assignments themselves. They would naturally be motivated for doing assignments and are typically among the most motivated students. The same goes for the experimental group, especially when it comes to learning outcome, that they are the type of students that learn best when given autonomy and freedom, and as such does not represent the entirety of the student population sufficiently. The fact that many of them chose to do assignments anyway leads to thinking that they want assignments to learn anyway, and as such discredits that bias.

The experimental group also have certain threats to validity. They have volunteered and chosen to be part of a small test group. This could lead to them being more positively inclined than what they otherwise would have been and felt more pushed to work harder in the course than they would have done if they knew they were not being measured.

As another threat to validity, much of the reduction in gain score between the pretest and the posttest seemed to be because people were unable to complete the test, thus giving an extra advantage to fast typers, and students solely focusing more on the quality of the first assignments, then not having enough time for the last part. This could have skewed the results, highlighting more individual traits than the learning outcome that could have come out of distinct groups.

Lastly, all corrections of tests have been done by the writer of this thesis. While the respondents were anonymized, so I did not know which group the test belonged to, there is still the possibility of human error in the corrections, making mistakes while correcting the tests that could have changed the result. The probability of this happening to the extent that it would change the statistical result of the test is, however, low.

Chapter 7

Conclusion

This chapter concludes this thesis and the research questions that it set out to answer about the effect of mandatory assignments on students' learning outcome and motivation in introductory programming courses. The essential part about and the ultimate goal of this thesis is to work towards giving computer science students the best tool and guidance for education. To provide the computer scientists of tomorrow with the chance to excel in their field when they finish with their studies. The world needs technologists in the future, and educational institutes should take their part when it comes to finding the best possible way of teaching computer science, and make sure the least amount of students drop out.

7.1 What is the Effect of Mandatory Assignments on Students' Learning Outcome in Introductory Programming Courses

The results of the experiment indicate that there are no statistically significant differences between learning outcome of students following a mandatory assignment program and students that are given more autonomy to obtain the necessary course knowledge through their own means. The result indicates that mandatory assignments are not necessarily helpful for learning the course. Hence, there should be a consideration of whether resources going into approving assessments are better spent otherwise. Assignments are also given out to test curriculum that can not be tested on the exam, but the emphasis on how much of the course is assignments, and how much are self-study should be reconsidered. Assignments along the semester help to push students into working, and give them goals to work towards that is not far into the future, as the exam. However, there should be more focus on formative evaluation and self-study throughout the semester. It is essential not to forget about relatedness and the social arena that the university is as well. Removing all reasons for the students to meet up at the university location may cause loneliness among students, so there should be mechanisms in place to make sure students get to know one another. Group projects and code reviews also give important collaboration skills that the students should learn.

Interestingly, although not statistically significant enough, the weaker students seem to be even more receptive for this kind of autonomy, performing better than the control group on learning outcome. This could be due to the extensive cheating on assignments culture present at NTNU, with no point in cheating if it is only for self-learning.

Interviews with the students in the experimental group show that many students do the assignments fully or partially even if they are not required to do so — those who do not have found other online courses or projects by which to learn the curriculum. Spending resources on checking whether students have done the assignments are then probably not as helpful, as checking whether they have learned something. Interviews with the students also indicated that student who was not required to do the assignments did them with more focus on the assignment learning goals, and less focused on completing the tasks and making tests pass.

The formative evaluation of students' learning outcome during a session with the teaching assistants is quick to disappear when coupled with summative assessment, the approval of an exercise to qualify for the exam. If students' learning outcome is the same regardless of whether this summative evaluation is given, the focus of assignments throughout the semester should be focused on the formative evaluation. This would increase students' learning, focusing them on what they need to learn better and free up resources for students that are receptive for this kind of feedback.

This thesis does not aim to get rid of assignments all together, as exercises are beneficial for gaining knowledge, and knowing what you have learned and what you have missed. However, collecting and grading the assignments may not be as helpful as we once have thought. The assignments are meant to force students to learn evenly throughout the semester. However, the results indicate they will do so anyway, and we should focus on helping them learn by doing these assignments, instead of only measuring their progress in a summative way. Reducing the number of compulsory assignments in a course may then bring together the best of both worlds, avoiding students procrastination while at the same time giving them time to focus on learning the curriculum through self-study.

Further research into other types of approaches to learning activities is suggested, to see whether other activities are more suited than assignments for guiding students along the semester into learning.

7.2 What is the Effect of Mandatory Assignments on Students' Motivation in Introductory Programming Courses

The results of the experiment indicate that there are no statistically significant differences between motivation for the course after having compulsory assignments or not. This result is surprising, considering the theory of Self-Determination, students that were given more autonomy would have been thought to have a higher degree of motivation, as also was the result of several studies. It could be that the exterior motivation of getting a better grade on the exam, and this through doing the assignments well, had a higher effect on students response, than their intrinsic motivation when it came to learning about object-oriented programming. It could also have been that the missing focus on competence and relatedness outweighed the result of autonomy.

Many biases and other factors could have affected this result, most importantly, that the students in the control group chose to participate in the experiment, but wanted to have assignments. Also, the students volunteering for such a control group would often be the most motivated students, volunteering for extra exercises. Further work on a larger group of students is needed to clarify the results and get a more definite answer. Further work should also involve more research on how to best test students' motivation by using other types of surveys that have proven to be proper tools for measuring motivations.

It is hard to give any specific recommendation for future learning interventions from the results of this thesis, as neither treatment had any significant advantage for motivation or learning outcome. However, the fact that there was no significant difference at least indicates that it is not particularly dangerous to experiment with voluntary exercises instead of compulsory ones. After all, the compulsory aspect demands quite a lot of extra resources related to approvals, plagiarism, administration of permissions to take the exam, handling of exceptions such that students who missed some exercises due to disease, etc. With voluntary exercises, these resources could instead be used for better teaching and extra guidance of struggling students.

Chapter 8

Further Work

This chapter elaborates on further work that could be undertaken for further research into how mandatory assignments affect students, and how teaching computer science should be approached.

This thesis has been written based on an experiment with a low number of student participators, thus making it not thoroughly statistically significant. Significant further work should be done to verify the results of this experiment, and further expanding upon the results.

Most important further work, which is also available in a short horizon, is to collect the exam results of the experimental and control group, for those who agreed that this could be used. These results were not available before the delivery of this thesis but could be used to get a new perspective on the actual achievement of learning outcome for the students. Students learn much during the semester, but some would suggest they learn even more in the weeks before the exam. A posttest at the end of the semester is then maybe not a good enough tool for actually measuring learning outcome, as either the assignments or a free autonomous project could have laid a more solid foundation for acquiring more knowledge and be better prepared for the exam. Analyzing these results afterward to see if they match with the result from the posttest will be very interesting.

Conducting a new, similar, experiment with a larger number of students could also yield more reliable results. As the result of this experiment have indicated there is no difference in neither motivation nor learning outcome for students having compulsory assignments or not, the ethical problem of randomly dividing students into two groups could be said to be diminished.

This could also be done as a similar, larger experiment, with not as much measuring, where students sign up for either of the options, either following the assignments or learn on their own. Whether they should be measured biweekly, as they have in this experiment, is left to the conductors of the next test. However, a form of summative evaluation to guide students' learning has been seen to be productive and beneficial. This would reduce any ethical dilemmas, as participation is still voluntary, while the students who want to be pushed and feel the need for that still have that option.

While the results had a small indication that weaker students benefited from autonomy, there were no students among the weakest part of the population, those that barely passed the introduction to IT course. With a more significant number of students doing a similar experiment, these could be more focused on, to see whether they required more guidance, or that they were good left on their own. This could reduce the chance of them cheating throughout the semester, but rather focused on learning.

In addition to experiments comparing compulsory and voluntary exercises, there could also be experiments comparing different types of compulsory exercises. For instance, one could experiment with process versus product-focused assignments, where the product focused would be like the typical style today. You solve a problem with code, and then this is auto-checked or shown to the teaching assistant that approves. The process-focused could be more like this experiment has done, a weekly or biweekly session with a teaching assistant, where you talk about what you have been able to learn and what you are struggling with. In a programming course, this discussion with the teaching assistant might typically also be about attempted solutions to programming problems, and get help in how to solve these problems. This could be done in different paces for the students, a student could struggle longer with theory, while the rest of the class has moved on to new concepts.

For the students, this has been one of four courses they have had this semester, the other having mandatory assignments, often weekly. As the students mentioned in interviews, it is easier to deprioritize this course, when they did not have a fixed schedule or a date that assignments had to be delivered on, and other exercises were gathered around the same time. Managing to conduct a similar experiment, with all the courses together, to see the effect on students behavior, motivation, and learning could yield interesting results with less disturbance from other factors.

An experiment trying out a change in emphasis on how many assignments are given out during the semester could also be undertaken to see if that could change anything. Most of the students' time at NTNU is now going from deadline to deadline, delivering assignment after assignment. Reducing the number of exercises, but while still keeping some to push students into working, and not delay everything until the weeks before the exam, could help in increasing self-study. A challenge with this is if the assignments had to cover the curriculum of the missing assignments, thus only increasing the size of the assignments that were left. While talking about increasing self-study, one should ask the question if we are lecturing too much. Nominally, students are supposed to spend twelve hours per week with the course, but in practice, many students spend less time, as also was seen in Section 6.4. The average time spent per course, according to Studiebarometer for computer science students is nine hours. [93]. A student who goes to all the lectures will then only have three hours left for self-study and doing assignments. Hence, although reducing assignments may increase self-study, reducing the number of lectures might have even bigger potential, as long as one were able to motivate students to do something useful with the time that was freed up. Self-study shifts the focus to learning goals and learning the curriculum without the press of an upcoming deadline. This can also be combined with added research into how students approach their course. What do they do to learn the course when not working with exercises given by the staff.

There is also several data and statistical analysis that can be done with the data collected in this thesis. Mapping hours spent on the course, alongside descriptive information of how each student have approached the course, what they have planned to do, what they the result ended up on the exam. Combining this data and test results may give valuable insight into how students learn, and the individual differences between them. Using all data collected to look for different relationships was out of scope for this thesis, but may give insight into the future for those looking for it.

Looking at the teaching assistants role in guiding students into learning is also something that can be mapped in the future. How can it be ensured that their formative evaluation ends up being a formative evaluation? The resources that are spent on hiring teaching assistants and their training need to be targeted to reach this goal. Looking at the session that teaching assistants have with their students and how this session can be of maximum benefit to the students can help take into advantage all resources that a course has available.

Both university employees and students are curious and eager to improve learning quality at universities. Focusing research into how to give students the best possible foundation for further learning, and preparing them for the technological future where they will be sorely needed will be vital for Norway and all other countries preparing for the fourth industrial revolution. Finding a systematic and evidence-based approach for doing this should, therefore, be of top priority for universities worldwide.

Bibliography

[1] DAMVAD. *Dimensjonering av avansert IKT-kompetanse*. 2014.

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

[3] Samordna opptak. https://www.samordnaopptak.no/info/, 2019. Accessed: 2019-04-03.

[4] Silja Baller, Soumitra Dutta, and Bruno Lanvin. *Global information technology report 2016*. Ouranos Geneva, 2016.

[5] Mike Joy and Michael Luck. Plagiarism in programming assignments. *IEEE Transactions on education*, 42(2):129–133, 1999.

[6] Susan Bergin and Ronan Reilly. The influence of motivation and comfort-level on learning to program. 2005.

[7] Anthony Robins, Janet Rountree, and Nathan Rountree. Learning and teaching programming: A review and discussion. *Computer science education*, 13(2):137–172, 2003.

[8] Jens Bennedsen and Michael E Caspersen. Failure rates in introductory programming. *AcM SIGcSE Bulletin*, 39(2):32–36, 2007.

[9] Jens Bennedsen and Michael E Caspersen. Failure rates in introductory programming: 12 years later. *ACM Inroads*, 10(2):30–36, 2019.

[10] Richard M Felder. How students learn: Adapting teaching styles to learning styles. In *Frontiers in Education Conference, 1988., Proceedings*, pages 489–493. IEEE, 1988.

[11] Ralph Winfred Tyler, Robert Mills Gagné, and Michael Scriven. *Perspectives of curriculum evaluation*, volume 1. Rand McNally, 1967.

[12] Benjamin S Bloom. Some theoretical issues relating to educational evaluation. *Educational evaluation: new roles, new means: the 63rd yearbook of the National Society for the Study of Education*, (part II):26–50, 1969.

[13] Terence J Crooks. The impact of classroom evaluation practices on students. *Review of educational research*, 58(4):438–481, 1988.

[14] Paul Black and Dylan Wiliam. Assessment and classroom learning. *Assessment in Education: principles, policy & practice*, 5(1):7–74, 1998.

[15] John Hattie and Helen Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007.

[16] Karee E Dunn and Sean W Mulvenon. A critical review of research on formative assessment: The limited scientific evidence of the impact of formative assessment in education. *Practical assessment, research & evaluation*, 14(7):1–11, 2009.

[17] Randy Elliot Bennett. Formative assessment: A critical review. *Assessment in Education: Principles, Policy & Practice*, 18(1):5–25, 2011.

[18] John Haugan and Marius Lysebo. Hvorfor antall arbeidskrav bør reduseres. *Uniped*, 41(03):347–360.

[19] Tansy Jessop, Laura Gubby, and Angela Smith. Space frontiers for new pedagogies: a tale of constraints and possibilities. *Studies in Higher Education*, 37(2):189–202, 2012.

[20] Paul Black and Dylan Wiliam. Inside the black box: Raising standards through classroom assessment. *Phi Delta Kappan*, 92(1):81–90, 2010.

[21] Arto Vihavainen, Jonne Airaksinen, and Christopher Watson. A systematic review of approaches for teaching introductory programming and their influence on success. In *Proceedings of the tenth annual conference on International computing education research*, pages 19–26. ACM, 2014.

[22] Marianne Lykke, Mayela Coto, Sonia Mora, Niels Vandel, and Christian Jantzen. Motivating programming students by problem based learning and lego robots. In *2014 IEEE Global Engineering Education Conference (EDUCON)*, pages 544–555. IEEE, 2014.

[23] Nikolaos Avouris, Stefanos Kaxiras, Odysseas Koufopavlou, Kyriakos Sgarbas, and Polyxeni Stathopoulou. Teaching introduction to computing through a project-based collaborative learning approach. In *2010 14th Panhellenic Conference on Informatics*, pages 237–241. IEEE, 2010.

[24] Daniel C Cliburn, Susan M Miller, and Emma Bowring. Student preferences between open-ended and structured game assignments in cs1. In *2010 IEEE Frontiers in Education Conference (FIE)*, pages F2H–1. IEEE, 2010.

[25] Andrew Luxton-Reilly, Ibrahim Albluwi, Brett A Becker, Michail Giannakos, Amruth N Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, Claudia Szabo, et al. Introductory programming: a systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pages 55–106. ACM, 2018.

[26] Raymond Lister, Elizabeth S Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik Moström, Kate Sanders, Otto Seppälä, et al. A multi-national study of reading and tracing skills in novice programmers. In *ACM SIGCSE Bulletin*, volume 36, pages 119–150. ACM, 2004.

[27] Michael McCracken, Vicki Almstrum, Danny Diaz, Mark Guzdial, Dianne Hagan, Yifat Ben-David Kolikant, Cary Laxer, Lynda Thomas, Ian Utting, and Tadeusz Wilusz. A multi-national, multi-institutional study of assessment of programming skills of first-year cs students. In *Working group reports from ITiCSE on Innovation and technology in computer science education*, pages 125–180. ACM, 2001.

[28] Andrew Luxton-Reilly. Learning to program is easy. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 284–289. ACM, 2016.

[29] Theodora Koulouri, Stanislao Lauria, and Robert D Macredie. Teaching introductory programming: a quantitative evaluation of different approaches. *ACM Transactions on Computing Education (TOCE)*, 14(4):26, 2015.

[30] Brenda Cheang, Andy Kurnia, Andrew Lim, and Wee-Chong Oon. On automated grading of programming assignments in an academic institution. *Computers & Education*, 41(2):121–131, 2003.

[31] Colin G Johnson and Ursula Fuller. Is bloom's taxonomy appropriate for computer science? In *Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006*, pages 120–123. ACM, 2006.

[32] Ursula Fuller, Colin G Johnson, Tuukka Ahoniemi, Diana Cukierman, Isidoro Hernán-Losada, Jana Jackova, Essi Lahtinen, Tracy L Lewis, Donna McGee Thompson, Charles Riedesel, et al. Developing a computer science-specific learning taxonomy. *ACM SIGCSE Bulletin*, 39(4):152–170, 2007.

[33] Lorin W Anderson, David R Krathwohl, Peter W Airasian, Kathleen A Cruikshank, Richard E Mayer, Paul R Pintrich, James Raths, and Merlin C Wittrock. A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives, abridged edition. *White Plains, NY: Longman*, 2001.

[34] Essi Lahtinen. A categorization of novice programmers: A cluster analysis study. In *PPIG*, volume 16, pages 32–41, 2007.

[35] Richard M Felder. Matters of style. *ASEE prism*, 6(4):18–23, 1996.

[36] Richard M Felder and Joni Spurlin. Applications, reliability and validity of the index of learning styles. *International journal of engineering education*, 21(1):103–112, 2005.

[37] Laura J Solomon and Esther D Rothblum. Academic procrastination: Frequency and cognitive-behavioral correlates. *Journal of counseling psychology*, 31(4):503, 1984.

[38] Stephen H Edwards, Jason Snyder, Manuel A Pérez-Quiñones, Anthony Allevato, Dongkwan Kim, and Betsy Tretola. Comparing effective and ineffective behaviors of student programmers. In *Proceedings of the fifth international workshop on Computing education research workshop*, pages 3–14. ACM, 2009.

[39] Piers Steel. The nature of procrastination: A meta-analytic and theoretical review of quintessential self-regulatory failure. *Psychological bulletin*, 133(1):65, 2007.

[40] Caroline Senecal, Richard Koestner, and Robert J Vallerand. Self-regulation and academic procrastination. *The journal of social psychology*, 135(5):607–619, 1995.

[41] Ulrich Trautwein, Olaf Köller, Bernhard Schmitz, and Jürgen Baumert. Do homework assignments enhance achievement? a multilevel analysis in 7th-grade mathematics. *Contemporary Educational Psychology*, 27(1):26–50, 2002.

[42] Shaljan Areepattamannil and Berinderjeet Kaur. Factors predicting science achievement of immigrant and non-immigrant students: A multilevel analysis. *International Journal of Science and Mathematics Education*, 11(5):1183–1207, 2013.

[43] Jehanzeb R Cheema, Kimberly Sheridan, et al. Time spent on homework, mathematics anxiety and mathematics achievement: Evidence from a us sample. *Issues in Educational Research*, 25(3):246, 2015.

[44] Wei-Cheng Mau and Richard Lynn. Gender differences in homework and test scores in mathematics, reading and science at tenth and twelfth grade. *Psychology, Evolution & Gender*, 2(2):119–125, 2000.

[45] R De Jong, Klaas Jan Westerhof, and Bert PM Creemers. Homework and student math achievement in junior high schools. *Educational Research and Evaluation*, 6(2):130–157, 2000.

[46] Huiyong Fan, Jianzhong Xu, Zhihui Cai, Jinbo He, and Xitao Fan. Homework and students' achievement in math and science: A 30-year meta-analysis, 1986–2015. *Educational Research Review*, 20:35–54, 2017.

[47] Jan Wilson and John Rhodes. Student perspectives on homework. *Education*, 131(2):351–359, 2010.

[48] Emily Fast Christensen. *The effect of homework choices on achievement and intrinsic motivation*. 2002.

[49] J Daniel House. The effect of student involvement on the development of academic self-concept. *The Journal of Social Psychology*, 140(2):261–263, 2000.

[50] Borislava Gutarts and Franque Bains. Does mandatory homework have a positive effect on student achievement for college students studying calculus? *Mathematics and Computer Education*, 44(3):232, 2010.

[51] Gail Weems. The impact of homework collection on performance in intermediate algebra. *Research and Teaching in Developmental Education*, pages 21–25, 1998.

[52] Carolyn M Cartledge and John E Sasser. The effect of homework assignments on the mathematics achievement of college students in freshman algebra. 1981.

[53] Ellen Miller and Geraldine Westmoreland. Student response to selective grading in college economics courses. *The Journal of Economic Education*, 29(3):195–201, 1998.

[54] Holger Danielsiek and Jan Vahrenhold. Stay on these roads: Potential factors indicating students' performance in a cs2 course. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 12–17. ACM, 2016.

[55] Richard M Ryan and Edward L Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1):68, 2000.

[56] Edward L Deci and Richard Flaste. *Why we do what we do: Understanding self-motivation*. Penguin books London, 1996.

[57] What is the self-determination theory of motivation. https://positivepsychologyprogram.com/self-determination-theory/, 2019. Accessed: 2019-02-12.

[58] Susan Harter. Effectance motivation reconsidered. toward a developmental model. *Human development*, 21(1):34–64, 1978.

[59] Edward L Deci. Effects of externally mediated rewards on intrinsic motivation. *Journal of personality and Social Psychology*, 18(1):105, 1971.

[60] Robert J Vallerand and Greg Reid. On the causal effects of perceived competence on intrinsic motivation: A test of cognitive evaluation theory. *Journal of Sport Psychology*, 6(1):94–102, 1984.

[61] Abraham H Maslow. A theory of human motivation. *Psychological review*, 50(4):370, 1943.

[62] Roy F Baumeister and Mark R Leary. The need to belong: desire for interpersonal attachments as a fundamental human motivation. *Psychological bulletin*, 117(3):497, 1995.

[63] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

[64] Frederick Herzberg. *Motivation to work*. Routledge, 2017.

[65] Christopher P Niemiec, Richard M Ryan, and Edward L Deci. The path taken: Consequences of attaining intrinsic and extrinsic aspirations in post-college life. *Journal of research in personality*, 43(3):291–306, 2009.

[66] Christopher P Niemiec and Richard M Ryan. Autonomy, competence, and relatedness in the classroom: Applying self-determination theory to educational practice. *School Field*, 7(2):133–144, 2009.

[67] Maarten Vansteenkiste, Willy Lens, and Edward L Deci. Intrinsic versus extrinsic goal contents in self-determination theory: Another look at the quality of academic motivation. *Educational psychologist*, 41(1):19–31, 2006.

[68] Edward L Deci, Richard Koestner, and Richard M Ryan. A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological bulletin*, 125(6):627, 1999.

[69] Johnmarshall Reeve and Hyungshim Jang. What teachers say and do to support students' autonomy during a learning activity. *Journal of educational psychology*, 98(1):209, 2006.

[70] Carl A Benware and Edward L Deci. Quality of learning with an active versus passive motivational set. *American Educational Research Journal*, 21(4):755–765, 1984.

[71] Christopher P Niemiec, Martin F Lynch, Maarten Vansteenkiste, Jessey Bernstein, Edward L Deci, and Richard M Ryan. The antecedents and consequences of autonomous self-regulation for college: A self-determination theory perspective on socialization. *Journal of adolescence*, 29(5):761–775, 2006.

[72] Aaron E Black and Edward L Deci. The effects of instructors' autonomy support and students' autonomous motivation on learning organic chemistry: A self-determination theory perspective. *Science education*, 84(6):740–756, 2000.

[73] Wendy S Grolnick, Richard M Ryan, and Edward L Deci. Inner resources for school achievement: Motivational mediators of children's perceptions of their parents. *Journal of educational psychology*, 83(4):508, 1991.

[74] Maureen Biggers, Anne Brauer, and Tuba Yilmaz. Student perceptions of computer science: a retention study comparing graduating seniors with cs leavers. In *ACM SIGCSE Bulletin*, volume 40, pages 402–406. ACM, 2008.

[75] Joanna Wolfe and Elizabeth Powell. Not all curves are the same: Left-of-center grading and student motivation. *age*, 26:1.

[76] Lecia J Barker, Charlie McDowell, and Kimberly Kalahar. Exploring factors that influence computer science introductory course students to persist in the major. In *ACM SIGCSE Bulletin*, volume 41, pages 153–157. ACM, 2009.

[77] Junit 5. https://junit.org/junit5/. Accessed: 2019-4-5.

[78] Theresa Beaubouef and John Mason. Why the high attrition rate for computer science students: some thoughts and observations. *ACM SIGCSE Bulletin*, 37(2):103–106, 2005.

[79] Päivi Kinnunen and Lauri Malmi. Why students drop out cs1 course? In *Proceedings of the second international workshop on Computing education research*, pages 97–108. ACM, 2006.

[80] James A Kulik and Chen-Lin C Kulik. Timing of feedback and verbal learning. *Review of educational research*, 58(1):79–97, 1988.

[81] Ruth Butler. Task-involving and ego-involving properties of evaluation: Effects of different feedback conditions on motivational perceptions, interest, and performance. *Journal of educational psychology*, 79(4):474, 1987.

[82] Graham Gibbs and Harriet Dunbar-Goddet. The effects of programme assessment environments on student learning. *The Higher Education Academy*, 2007.

[83] Donald T Campbell and Julian C Stanley. *Experimental and quasi-experimental designs for research*. Ravenio Books, 2015.

[84] Cleo H Cherryholmes. Notes on pragmatism and scientific realism. *Educational researcher*, 21(6):13–17, 1992.

[85] Louis Cohen. *Research methods in education*. Routledge, London New York, 2018.

[86] Briony Oates. *Researching information systems and computing*. SAGE Publications, London Thousand Oaks, Calif, 2006.

[87] Stata, statistical software for data science. https://www.stata.com/. Accessed: 2019-05-07.

[88] William Trochim. *Research methods : the essential knowledge base*. Cengage Learning, Boston, MA, 2016.

[89] Microsoft excel. https://products.office.com/nb-no/excel, 2019. Accessed: 2019-05-04.

[90] Norwegian center for research data. https://nsd.no/nsd/english/index.html. Accessed: 2019-1-14.

[91] Jenny Preece. *Interaction design : beyond human-computer interaction*. John Wiley & Sons Ltd, Chichester, West Sussex, 2015.

[92] Shannon E Ross, Bradley C Niebling, and Teresa M Heckert. Sources of stress among college students. *Social psychology*, 61(5):841–846, 1999.

[93] Studiebarometeret. http://www.studiebarometeret.no/no/, 2018. Accessed: 2019-05-22.

Appendix

8.1 Pretest

Pre-test masterprosjekt om effekten av obligatorisk øvingsopplegg.

Dette er en test for å kartlegge tidlige kunnskaper i faget TDT4100 Objektorientert Programmering. Dette er **ikke** en test av deg som person, og vil aldri bli brukt til å vurdere deg.

Hensikten med testen er at den skal være lang, og ikke mulig å løse på de to timene som er gitt. Fortvil dermed ikke dersom du ikke får til mange oppgaver, eller noe i det hele tatt. Oppgavene kan løses i den rekkefølgen du vil, men noen av oppgavene bygger på hverandre.

Når to timer er gått, send kodefilene du har skrevet og svarene på teorioppgaver du har til meg på mail. **vegard.hellem@ntnu.no**

Sørg for at det er rolig rundt deg når du skal gjøre oppgavene, og at du ikke blir forstyrret. Du kan bruke internett til å google på kodeoppgaver, men ikke på teori. Det er ikke lov å spørre andre om hjelp.

Hvis noe er uklart i oppgaven, skriv nødvendige antagelser du tar.

1. Teori-oppgaver

a. Nevn to gode grunner til at vi bruker innkapsling (%)
   1. Intern logikk kan endres.
   2. Sikre gyldighet av tilstand

b. Hvilke **synlighetsmodifikatorer** har vi i Java og når bruker vi de. (3.33)
   Private, public, (ingen), (protected)
   Private - metoder som andre skal ha tilgang til (og konstanter (FINAL))
   Private - hjelpemetoder og felt

c. Hva er forskjellen på tilstand og oppførsel? (3.33%)
   Tilstand: Felter
   Oppførsel: Metoder som kan brukes

Oppgave 2

2. Skriv **Country**-klassen. (30%)
   a. **Country-klassen** skal ha følgende felt. **String** name, **int** inhabitants, **int** squareMeters.
   b. Lag gettere og settere til feltene til **Country-klassen.** Gjør nødvendige antagelser om innkapsling. (5)
      i. **name-**feltet skal kun inneholde tegn, ingen mellomrom, tall, bindestrekker eller annet. (5)pretest
   c. Skriv funksjonen **immigration(int** number**)** som skal oppdatere innbyggertallet med nye innvandrere. (5)
   d. Skriv funksjonen **emigration(int** number**)** som skal oppdatere innbyggertallet med at folk har utvandret. 5
   e. Skriv funksjonen **immigration(int** number, **Country** otherCountry). Disse skal få innvandrere fra otherCountry, redusere folketallet der, og øke folketallet i dette landet. (5)
   f. Skriv funksjonen **union(**Country country) som skal returnere et nytt Country-objekt, med summen av innbyggere og størrelsen på de to landene som

slår seg sammen. Navnet skal være navnet til det landet som utfører metoden. (5)

Oppgave 3

3. Skriv **Employee-klassen.**
    a. Employee-klassen, skal ha følgende felt. Id, firstName, lastName,monthlySalary, og startDate, og endDate.
    b. Skriv gettere og settere til **Employee**. Gjør nødvendige antagelser om innkapsling. (2)
    c. Skriv funksjonen **getAnnualSalary()** - denne funksjonen skal returnere årslønnen til den ansatte. (3)
    d. Skriv funksjonen **raiseSalary**(**int** percent) som skal øke den månedlige lønnen til den ansatte med en viss prosent. (5)
    e. Skriv funksjonen **terminateEmployement().** Det blir opp til deg å velge hva du synes det er naturlig at denne funksjonen skal gjøre. (5)

Oppgave 4.

4. **Book og Author-klassene.**
    a. Skriv klassen **Book.** Inkluder følgende felter. Name, price, quantity og author (Som skal være et Author objekt).
        i. Lag gettere og settere for feltene. (2)
        ii. Book skal ha to konstruktører. (5)
            1. public Book (String name, Author author, double price) { ...... }
            2. public Book (String name, Author author, double price, int qty) { ...... }
            3. Lag konstruktørene slik at de oppdaterer feltene riktig.
        iii.

b. Skriv klassen **Author.** Author-klassen skal ha følgende felt: **name, email, gender,** og **List<Book> books.**
    i. Lag gettere og settere for feltene. (3)
c. Skriv **toString-metoden** til Author. Den bør returnere (2)
    i. Author[name=?,email=?,gender=?]
d. Skriv **toString-metoden** til Book. Den skal returnere følgende. (3)
    i. Book[name=?,Author[name=?,email=?,gender=?],price=?,qty=?"
e. Lag metoden **addBook(Book book) i Author-**klassen. Metodene skal gjøre følgende (5)
    i. Legge til boken i listen over bøkene author-har skrevet.
    ii. Passe på at boken sitt **Author-objekt** nå peker på **Author.**
f. Oppdater **setAuthor(Author author)** i Book til å gjøre det samme (5)
    i. Setter author objektet til å være den nye author.
    ii. Legger til denne boka i listen over bøkene **author** har skrevet.
g. Skriv metoden **int potentialIncome()** i **Author-klassen**. Metoden skal beregne mulig inntekt for alle bøkene **Author** har skrevet, gitt bøkenes *price* og *quantity.* **(5)**

Oppgave 5 - Multiple Choice

Dersom flere er riktige, skriv ned alle svar-alternativ du mener er riktig. (1% per riktig)

1. **Hva er meningen med retur-typen *void* i Java?**
   (C) Returnerer null
   (B) **Returnerer ingen data type**
   (C) void brukes ikke i Java
   (D) Returnerer en fri plass i minnet til utvikleren.

2. Hvem kan aksessere felter av typen private?
   (A) Bare static metoder i samme klasse.
   (B) **Bare instanser av samme klasse.**
   (C) Bare metoder definert i samme klasse
   (D) Bare klasser i samme pakke

3. Hva er riktig måte å aksessere metoden *public static calculateCost(int price, int quantity)* i klassen MyClass i en main-metode uten annen kode?

(A) **MyClass.calculateCost(2, 2);**
(B) MyClass myClass = new MyClass();
myClass.calculateCost(2, 2);
(C) calculateCost(2, 2);
(D) myClass.calculateCost(2, 2);

4. Hvilket nøkkelord bør stå foran de fleste metoder i en klasse?

(A) final
(B) **public**
(C) private
(D) void

5. Hvor mange ganger vil følgende for-løkke kjøre?

```
for(int i= 0; i<5; i++) {
    if (i == 3) {
        i++;
    }
    System.out.println(i);
}
```

(A) 3
(B) 4
(C) 5
(D) 6

8.2 Posttest

Post-test masterprosjekt om effekten av obligatorisk øvingsopplegg.

Dette er en test for å kartlegge kunnskaper i faget TDT4100 Objektorientert Programmering etter å ha fulgt et øvingsopplegg. Dette er **ikke** en test av deg som person, og vil aldri bli brukt til å vurdere deg.

Hensikten med testen er at den skal være lang, og ikke beregnet på å bli ferdig på de **to timene** som er gitt. Fortvil dermed ikke dersom du ikke får til mange oppgaver, eller noe i det hele tatt. Oppgavene kan løses i den rekkefølgen du vil, men mange av oppgavene bygger på at du i hvertfall har sett på oppgave 3.

Når to timer er gått, send kodefilene du har skrevet og svarene på teorioppgaver du har til meg på mail. vegard.hellem@ntnu.no**. NB: send det som en zipfil med navnet posttest{id} hvor id er ditt unike nummer.**

Sørg for at det er rolig rundt deg når du skal gjøre oppgavene, og at du ikke blir forstyrret. Du kan bruke internett til å google på kodeoppgaver, men ikke på teori. Det er ikke lov å spørre andre om hjelp. Husk å sette på en klokke med to timer slik at du ikke gjør mer etter at de to timene har passert.

Hvis noe er uklart i oppgaven, skriv nødvendige antagelser du tar.

Oppgaven har vedlagt to filer som godt kan legges inn.

Oppgave 1 - Teori

1. Forklar forskjellen på checked og unchecked exceptions

2. Hvorfor er Interface et nyttig verktøy i observatør-observert-teknikken?

3. Hva er forskjellen på å bruke try/catch og throws?

4. Når du bruker grensesnitt har du mulighet å deklarere som **Grensesnitt objekt = new KlasseSomImplementererGrenseSnitt();** - hva kan være fordeler med at dette er mulig?

Oppgave 2 - Multiple Choice

Droppes.

Oppgave 3 - Bank

Gå ut ifra at du sitter på følgende klasse. (Du kan gjerne kopiere filen inn i en editor)

```
public class Account {

	private double balance;
	private double interestRate;
	private String ownerName;

	public Account(double balance, double interestRate, String ownerName) {
		checkNotNegative(balance, "Balance");
		this.balance = balance;
		this.ownerName = ownerName;
```

```
		setInterestRate(interestRate);
	}

	public Account(){

	}

	protected void checkNotNegative(double value, String valueName) {
		if (value < 0) {
			throw new IllegalArgumentException(valueName + " cannot be negative: "
+ value);
		}
	}

	public String toString() {
		return String.format("[Account balance=%f interestRate=%f", balance,
interestRate);
	}

	public double getInterestRate() {
		return interestRate;
	}

	public void setInterestRate(double interestRate) {
		checkNotNegative(interestRate, "Interest rate");
		this.interestRate = interestRate;
	}

	public double getBalance() {
		return balance;
	}

	public void deposit(double amount) {
		checkNotNegative(amount, "Amount");
		balance = balance + amount;
	}

	public void withdraw(double amount) {
		checkNotNegative(amount, "Amount");
		double newBalance = balance - amount;
		if (newBalance < 0) {
			throw new IllegalArgumentException("The balance cannot become
negative: " + newBalance);
```

```
        }
        balance = newBalance;
    }

    public void addInterest() {
        deposit(balance * interestRate / 100);
    }
}
```

- a) Lag en **Bank-**klasse som tar vare på en liste med kontoer. Banken bør ha metoder for å legge til, og fjerne kontoer.
- b) Lag en metode bankTransfer(Account account, Bank newBank) som skal overføre kontoen til den nye banken. Pass på at alle assosiasjoner er oppdatert. (Gamle banken skal ikke inneholde kontoen)
    - b2) Lag og metoden bankTransfer(List<Account> accounts, Bank newBank) som skal overføre flere kontoer til den nye banken.
- c) lag en metode accountTransfer(Account from, Account to, int amount) som skal overføre penger fra en konto til en annen.
- d) Lag en metode **getTotalAmount** som summerer opp hvor mye penger som er i banken
- e) Lag en metode **Map<String, double> getPersonBalance()** som skal returnere et Map med hvor mye penger hver unike person har på de forskjellige kontoene sine.
- d) Lag metoden

Oppgave 4 - Grensesnitt og annet

- a) Det skal være mulig å sammenlikne Account-objekter. Lag funksjonalitet som gjør det mulig å sortere med kontoer i **Bank-klassen** både på størrelse (balance), og på navnet til eieren. Den naturlige sorteringen er på balance, men det og være mulig å sortere på navnet. Vis også et eksempel på dette i en passende **main-metode.**
- b) Det skal være mulig å skrive koden. for(Account account: bank) hvor bank er et account-objekt. Implementer nødvendige grensesnitt og metoder for å gjøre dette mulig.
- Gå ut ifra at følgende grensesnitt eksister: (Kan godt kopieres inn)

```
public interface FinancialInstitution {

    public int getTotalAmount();
```

```
public boolean isSafe();

}
```

La Bank-klassen implementere grensesnittet, og implementer nødvendige metoder. isSafe() - metoden skal returnere true dersom banken oppfyller **alle** disse kriteriene.

- Har minimum 20 kontoer, med minimum 10 forskjellige personer som eiere.
- Har minimum 10 millioner kroner i total penger.
- Har **ikke** mer enn 5 kontoer som har en rente over 3 %.

Lag gjerne egne hjelpemetoder for dette.

Oppgave 5 - Filhåndtering

Gå ut ifra at du har Account-klassen fra forrige oppgave. Gå ut ifra at det finnes med en fil: kontoer.txt med formatet “belop,rente,eier”, eksempelvis

```
2000,2.5,Ola
5000,1.5,Kari
20000000,3.0,Petter
40000,2.5,Erna
```

Følgende klasse kan kopieres.

```
Public class bankIO {
    Public List<Account> getBankAccounts(String filename) {

            }
    }
```

a) Fyll ut metoden, slik at den leser inn kontoene og returnerer en liste med Account-objekter.

b) Lag også metoden **load(String filename) {}** i **Bank-**klassen, som skal bruke BankIO-klassen til å hente inn kontoer.

Oppgave 6 - Arv

- a) Lag en ny klasse **CreditAccount** som arver fra **Account-**klassen fra oppgave 3, men gjør det mulig å ha negativ balanse på kontoen. Det skal ikke være mulig å ha mer negativ balance enn en **limit -** som blir satt når kontoen blir opprettet.

Oppgave 7 - observatør og observert

- a) Lag et Interface - **TransactionListener** med en metode: **void listenToTransaction**. Bestem selv hvilke argumenter som er naturlig å ta inn. (Det kan lønne seg å lese hele oppgaven først)
- b) Utvid **Bank-klassen** til å kunne registrere, fjerne og si ifra til lyttere hver gang det skjer en transaksjon mellom to kontoer.
- c) Lag en implementasjon av **TransactionListener, InternalTransactionListener,** som skal printe ut til konsollen dersom transaksjonen som har skjedd er mellom to kontoer med samme eier.

Task 7 - The observable Technique

1. Create an interface - TransactionListener - with the method: void listenToTransaction. Decide which arguments are natural for this method to accept based on the task.
2. Extend the **Bank-class** from task 3, to be able to register and remove listeners, and tell listeners about each a time a transaction has happened between two **Accounts.**
3. Create an implementation of **TransactionListener, InternalTransactionListener,** which should print to the console if the transaction has happened between two accounts with the same owner.

8.3 Interview guide

Intervjuguide

Tema: Hvordan studenten opplevde semesteret

INTRO:

- Helt anonymt, og utenfor de andre delene. Derfor vil vi kanskje spørre om ting du har svart på før.
- Du kan trekke deg når som helst hvis du vil det, og hvis det er noe du ikke vil skal brukes.

1. Kan du beskrive hvordan du har gått frem for å lære deg faget?
   a. Har det endret seg noe over tid?

2. Har du savnet obligatoriske aktiviteter?

3. Hvordan har det gått å ha flere fag samtidig?
   a. Er det lettere å nedprioritere Java når de andre fagene har obligatoriske aktiviteter?

4. Hvordan har du benyttet deg av det “vanlige opplegget” (øvinger, forelesning etc.?
   a. Hvis du har gjort øvingene, hvor mange har de fått til?
   b. Hvorfor har du gjort det?

5. Hvordan har du benyttet deg av studentassistenter?

6. Hvilke andre ressurser i faget har du benyttet deg av?
   a. Wiki
   b. Forelesninger/Øvings-forelesninger
   c. Bok

7. Har du fulgt andre kurs på nett?
   a. Hva har du fått ut av dette?

8. Hvordan har motivasjonen din vært?

9. Har du lært noe? Føler du deg forberedt på eksamen? Forberedt på neste semester/arbeidslivet?

10. Avsluttende kommentarer?

8.4 Results

PRETEST_SCOI	POSTTEST_SC(	GAIN SCORE	GROUP (0 = Exp	Previous Grade	Study	Adjusted pretest
69	87	18	0	C	Computer Scienc	66.17
41.5	34.5	-7	0	C	Informatics	43.70
62	34	-28	0	B	Engineering and	60.45
59	45.5	-13.5	1	C	Communications	59.40
28.5	30.5	2	1	C	Computer Scienc	34.48
36	57.5	21.5	0	B	Communications	39.21
52	61.5	9.5	0	B	Engineering and	52.28
77	63	-14	0	A	Informatics	72.71
79	77	-2	0	A	Industrial Econor	74.34
64	53.5	-10.5	0	C	Computer Scienc	62.09
57	58	1	1	A	Industrial Econor	57.76
37.5	55.5	18	0	B	Communications	40.44
42	31.5	-10.5	0	B	Informatics	44.11
56.5	51	-5.5	0	B	Computer Scienc	55.96
83	76.5	-6.5	1	A	Industrial Econor	79.00
58.5	50	-8.5	1	B	Industrial Econor	58.99
73	38	-35	1	D	Engineering and	70.83
74	61	-13	1	C	Computer Scienc	71.65
66	36	-30	1	A	Computer Scienc	65.12
46.5	58.5	12	0	D	Computer Scienc	47.79
31.5	18	-13.5	0	B	Industrial Econor	35.53
54	46	-8	1	C	Engineering and	55.31
80.5	82	1.5	1	B	Computer Scienc	76.96
72	49	-23	1	A	Computer Scienc	70.02
78	85.5	7.5	1	A	Industrial Econor	74.92
60	38.5	-21.5	0	B	Engineering and	58.82
48	60	12	0	D	Computer Scienc	49.01
69	33	-36	0	A	Computer Scienc	66.17
13	4.5	-8.5	0	A	Informatics	20.42

75.5	79.5	4	1	B	Computer Scienc	72.88
49	48.5	-0.5	1	A	Industrial Econor	51.23
43	39.5	-3.5	1	D	Computer Scienc	46.32
40	35	-5	0	A	Computer Scienc	42.48
27	25.5	-1.5	1	B	Communications	33.25
52	42	-10	1	B	Computer Scienc	53.68
44.5	46.5	2	0	D	Information scien	46.16
71	67.5	-3.5	0	C	Informatics	67.81
71	67.5	-3.5	1	C	Communications	69.20
75.5	80	4.5	0	B	Computer Scienc	71.48
62.5	24.5	-38	0	B	Industrial Econor	60.86

8.5 Surveys

The following surveys have been used to recruit participants and gather data in this thesis.

- Participant recruitment surveys
  - Control group: https://forms.office.com/Pages/ResponsePage.aspx?id=cgahCS-CZ0SluluzdZZ8BSYnLepi-05Kh_DuQi6k-LZURE9MRThaR1ZIMldQMFVGVUVSUUJWTDYzSS4u
  - Experimental group: https://forms.office.com/Pages/ResponsePage.aspx?id=cgahCS-CZ0SluluzdZZ8BSYnLepi-05Kh_DuQi6k-LZUMjdLM0M4WVZYUENWUktUTDdVMFVLSkFSSy4u
- Weekly report: https://forms.office.com/Pages/ResponsePage.aspx?id=cgahCS-CZ0SluluzdZZ8BSYnLepi-05Kh_DuQi6k-LZURjFWOVoxV0FPVVpDRjlHNlA4VzFLOTRK
- Final survey:
  - Control group: https://forms.office.com/Pages/ResponsePage.aspx?id=cgahCS-CZ0SluluzdZZ8BSYnLepi-05Kh_DuQi6k-LZURE9MRThaR1ZIMldQMFVGVUVSUUJWTDYzSS4u
  - Experimental group: https://forms.office.com/Pages/ResponsePage.aspx?id=cgahCS-CZ0SluluzdZZ8BSYnLepi-05Kh_DuQi6k-LZUMkhHOUFTNzdVS0lUOUkwMzBKNjg4NDc0US4u

Vegard Hellem

The Effect of Mandatory Assignments in Introductory Programming

NTNU
Norwegian University of
Science and Technology