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

1st Vegard Hellem
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
vhellem@gmail.com

2nd Madeleine Lorås
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
madeleine.loras@ntnu.no

***Abstract*—In a world in high demand of engineering professionals, higher education should be effective and quality conscious. A better understanding of what type of activities that are best suited for improving students' learning could enable further improvements. In this paper, the effect of mandatory assignments on students' learning outcome in introductory programming courses is explored through a quasi-experimental research study. One group of students were exempted from the mandatory weekly assignments and followed up via biweekly sessions. A control group was recruited to follow an assignment regime in parallel. Through pre- and posttests the learning outcome of the two assignment structures was statistically evaluated. The results indicated that the group of students exempt from mandatory assignments achieved the same learning outcome as the control group. Similarly, no difference was found between the two groups on exam performance. Students have individual learning behaviors and learn to program in different ways, and the instructional design should facilitate individual learning trajectories.**

***Index Terms*—assessment, performance, mandatory assignments, computing education**

I. Introduction

Increasing student numbers provides challenges for teachers and educators at higher education institutions across the world as the search for scalable and effective teaching designs continues. In the Norwegian engineering education, mandatory assignments are a common way to ensure student engagement in a course between lectures [1]. Most often, these assignments are done on a weekly or biweekly basis and are not counted towards the final grade. Instead, such assignments are assessed on a pass/fail basis, where students are required to pass a fixed amount of assignments in order to qualify for the exam. Having assignments besides the exam is for the assignments to address other learning outcomes of the course than those of the exam. The exam could be too short to test all that should be learned in the course, and such the assignments are needed as a supplement in certifying that the students have learned all that they are supposed to. For example, this could be practical knowledge like a chemistry lab, which is unfeasible to test during an exam. Math assignments in a math course are more straight forward learning to prepare students for the exam. Programming courses fall in between these two examples, with assignments often mainly focusing on preparing students, but may test larger collaboration projects and coding challenges for which the exam does not have enough time. In this paper, we will explore a different approach to this traditional instructional technique: removing mandatory assignments.

During the spring semester 2019, a research study was done exploring the effects of mandatory assignments in an introductory programming course. Extensive resources go into grading these mandatory assignments, resources that could be spent on more effective evaluations such as formative feedback [2]. Therefore, the focus of the study was to measure and compare how learning outcome and student performance was affected by having or not having mandatory assignments in an introductory programming course. The research questions were as follows:

- What is the effect of mandatory assignments on students learning outcome?
- What is the effect of mandatory assignments on students performance?

The difference here between learning outcome and performance is based on the measurements. Learning outcome is measured with pre- and posttest, whereas performance is measured with exam grades. Of course, exam performance also measures learning outcome, but we have found that differentiating the learning dimension in this way provides a more nuanced insight.

A. Assessment

In order to explore assignments, we need to discuss assessment, and an important distinction is made between formative and summative assessment [3]. This contrast was first described by Scriben in 1967 [4]. He explained summative evaluation as assessment used to judge the value of an educational program, what had the student learned. Formative assessment targeted improvement for the student, and how they could improve learning. Bloom extended this definition of the purpose of formative evaluation to "Provide feedback and correctives at each stage in the teaching-learning process"

978-1-7281-0930-5/20/$31.00 ©2020 IEEE

[5]. Multiple studies have found that a formative approach outperforms summative assessment [2], [6], [7]. An important finding is that when the number of formative evaluations increases, students will learn more [2], [7], also for the most low-performing students [8].

B. Introductory programming

In an evaluation of different teaching approaches to introductory programming from 2015, Koulouri et al. studied three distinctive factors of how to improve introductory programming [3]. The choice of programming language and teaching problem solving before programming were found to yield significant improvements in student performance; however, it had variable effects on the acquisition of basic concepts in programming. The last factor was how to use feedback effectively and formatively. Here, they found that formative feedback was not useful unless students actively sought out and responded to feedback. In order to be effective, feedback should be timed and targeted to specific features that one wants students to improve [9]. As computer programs are files that can be run by a computer, there has been significant research into how to automatically grade and correct programming assignments, reducing the strain on teaching resources. These have plenty of issues that need to be looked at, especially for a system that grades the student based on these assignments [10].

Numerous other studies have also investigated what type of activities are most useful to teach computer science. A systematic review by Luxton-Reilly et al. in 2018 found, among other things, that self-paced learning had few examples of usage in universities worldwide [11]. Self-paced learning is a form of mastery learning where students are supposed to demonstrate they have achieved an appropriate level of mastery of a topic before they can move on the next, more advanced, topic in the course. They also found that problem-based learning could increase motivation and social interactivity. However, little evidence that it increases the learning outcome of the students. Problem-based learning was mainly project-based, answering open-ended questions [12], [13]. Through the review, they found evidence that students preferred structured assignments [14].

II. Mandatory assignments

The reason for having mandatory assignments in a course is often twofold. Compulsory assignments could be there to qualify students for the exam, or it could be to ensure they learn skills and knowledge that can not be assessed by the exam. A Norwegian study from 2018 [15] argues why the number of mandatory assignments in engineering education should be reduced based on findings that the use of mandatory assignments has increased without any quality improvement in students learning outcome.

A. Previous work on homework

As the literature on assignments at university level is limited [16], it is interesting to investigate the studies done on homework, in general, from pre-university education. Multiple studies have found a positive relationship between achievement and homework [17]–[19] in mathematics, while others find a non-relation, or even a negative impact on achievement, among these a study from 2010 on 28 different schools, where neither frequency nor homework time had any relation to performance in class according [20]. Similar inconsistent results have been shown in studies linking homework and science achievement. Some found positive relations [21], however, others did not [22]. A variety of factors may have contributed to these inconsistent findings. For instance, the type of homework, grading, how achievement is measured, and what kind of homework indicators that have been used. Studies have been convened on different data, including total time spent on homework, the frequency of homework, the percentage that was completed, the effort needed to complete the work, or the grade given to the homework if being evaluated by the teacher. In summary, the research reviewed has not indicated that there is a clear correlation between feedback on homework and student motivation or achievement gain [23]. It should be noted that homework completion rate has been shown to have an effect, but not the actual deliverance of the homework.

B. Assignments at university level

An interesting study on university-level calculus investigated the relationship between compulsory, graded assignments and assignments with weekly quizzes [16]. The results revealed that there was no statistically significant grade difference between these two groups. This result builds on early results that monitoring assignment completion, rather than just giving them out as an aid in learning the curriculum, does not affect students' performance [24]. However, if students are not given any exercises to aid in learning the syllabus, some results put them at a disadvantage compared to students getting mandatory assignments [25].

Similar results were found for a college degree economic course in a study looking at feedback and grading of assignments [26]. They tried out a concept called selective grading, where only a few select assignments were graded, and it had no effect on students' learning outcome; they produced at the same quality and delivered the same number of assignments.

Research on whether mandatory assignments are helpful in programming courses are limited. A review from 2016 by Danielsiek et al. [27] about ways to teach computer science found no evidence that results on assignments were any indication of how students would perform on the exam. This was regardless of whether the assignments counted towards the final grade, or whether it was just a stepping stone for being allowed to take the exam.

A Norwegian analysis by Haugan and Lysebo from 2018 [15] argues why the number of mandatory assignments in engineering education should be reduced. They concluded with multiple important findings. Among them that the students with less mandatory work, spent more time on each course than before, one of the most important reasons for having mandatory work in the first place. They also found that the average grade increased after the restructuring of the teaching program. This also included, to their surprise, the result for the students with the worst results on a preliminary test.

III. Methodology

The work presented in this paper is based on a master thesis project from 2019 [28]. A quasi-experimental research design was set up and implemented in order to investigate the effect of an intervention on a research population but without random selection [29], [30]. The intervention, in this case, was not having mandatory assignments, and the aim was to measure the effect on learning outcome and performance. An overview of the experiment is shown in Fig. 1, and in the following section, the course, participants, experiment, measurements and analysis will be described further.

A. Course description

The experiment was set up in an undergraduate object-oriented programming (OOP) course at a large university in Norway. The course yields 7.5 ECTS and goes over 14 weeks with a final four-hour exam, in the end, accounting for the whole grade. The programming language used is Java, and the course covers topics such as classes and objects, encapsulation, object structures, exception handling and inheritance. Students generally take this course in their second semester and are required to have completed an introduction to information technology course, which includes programming in Python. The course is mandatory for all the various computer science engineering programs and serves as an elective course for all engineering programs.

During the semester, there are four hours of topic lectures and two hours of exercise lectures a week, as well as mandatory assignments evenly spaced throughout the semester. There are ten assignments in total. They do not count towards the final grade; however, each assignment is graded on a point basis between 50-100, and to qualify for the exam, the student has to reach 750 points. The assignments are based on the curriculum for the current week and the week before. Automatic tests are integrated with the assignments so that both students and teaching assistants (TAs) can easily check the code. To pass the tests, students have to code correctly for all edge cases, as well as name their methods according to the task description. The assignments are delivered online but have to be demonstrated in-person to a TA within a week after the deadline. TAs are generally older students who have completed the course, hired by the department to give feedback, and help students with their assignments, as well as assign points to each assignment. Each TA is responsible for 20 students, and are available in open labs at least six hours every week.

This course design has been in place for seven years, with revisions to the assignments being made regularly. The student feedback is generally positive; however, the workload has been criticized somewhat. On the other hand, students report that the amount of practice and experience with programming in the course is very useful.

B. Participants

Among over 700 students taking the course, 40 students volunteered to be part of the experiment, either as a part of the experimental group with no mandatory assignments or as a part of a control group. The experimental group were exempt from doing the mandatory assignments and were instead given the freedom to choose what learning resources to use. These resources could include the proposed assignments for the course, but the students were not required to deliver them. They were, however, required to attend biweekly meetings with a TA where they had to describe what they had learned in the previous weeks and explain how they reached the learning objectives for that week. These meetings along with the pre and posttest, served as the experimental group's qualifying activities for the exam.

The students participating in the experiment were from various study programs within computer science. 47.5% were from computer science engineering, 20% from computer science and business, 15% from computer science, 15% from communications, and 12.5% from engineering and ICT. The gender distribution of the participants was 50/50 male and female.

C. Experimental design and ethical concerns

Both the experimental group (N=22) and the control group (N=18) were required to hand in weekly reports, as well as take a pre and post programming test. As naturally, they have learned much more during the semester; the second test was more difficult and involved more object-oriented programming principles than the first test. Both these tests were corrected by one of the authors, using anonymized IDs that did not indicate to which group the writer of the answers belonged. In addition to the weekly reports and pre-/posttests, some of the participants also volunteered to attend an informal interview about their experience at the end of the experiment. Lastly, the participants consented to their exam answers and results being collected for analysis.

The reason the selection of students was not random, was because the teaching team had concerns about implementing such a change to the students without certainty that their learning would not be affected negatively. Therefore, we decided that students would have to volunteer to be part of the experiment, which subsequently limited the number of participants as well. The experiment was approved by the Norwegian Centre for Research Data.

D. Measurements

In order to analyze the difference in learning outcome and performance for students with and without mandatory assignments, we created two hypotheses'.

- H_{11}: There is a difference in learning outcome for students with mandatory assignments and students without.
- H_{12}: There is a difference in performance for students with mandatory assignments and students without.

The learning outcome was measured through either the change of learning or with a modified pretest. The change

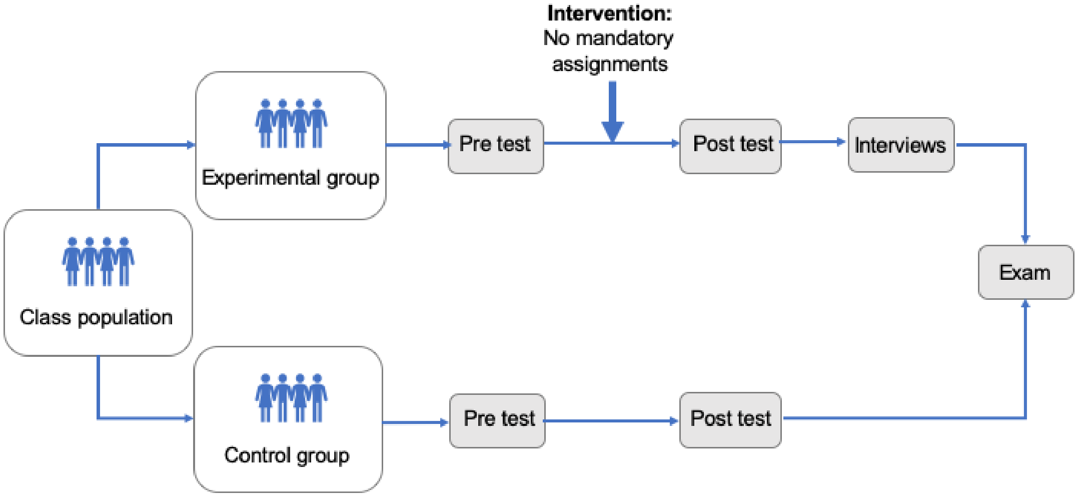

Fig. 1. Overview of experimental setup

of learning was measured as learning gain by subtracting the posttest score from the pretest score for each student. In order to deal with the quasi-experimental designs with non-randomized participants and the low number of participants, an adjusted pretest score was created in order to compensate for the nonequivalent groups design in a Reliability Corrected Analysis of Covariance model [31]. The reliability was calculated using Cronbach's Alpha, giving a reliability score of 0.817. This reliability was used to calculate adjusted pretest scores for feeding into the statistical model. The performance was measured with final exam grade, ranging from 0-5, where 0 is an F and 5 is an A.

Consequently, *post test score* and *exam grade* were the dependent variables, as indicated in bold in Table I. The independent variable *group* differentiated between the experimental and the control group. Additionally, *adjusted pretest score* acted as a covariate when analysing learning outcome and *grade in introductory programming (CS1)* for performance. All variables used to create these measures, as well as the variables used in the analysis, are summarized by group in Table I.

TABLE I
SUMMARY OF VARIABLES BY GROUP

	Experimental		Control	
Variable	μ	σ	μ	σ
Posttest score	49.54	20.69	53.36	18.28
Exam grade	2.43	1.65	2.17	1.82
Pretest score	54.30	17.04	61.17	16.73
Adjusted pretest	54.17	13.92	61.17	13.67
Gain score	-4.76	16.21	-7.81	11.62
Grade in CS1	3.74	0.96	3.83	1.04
N	22		18	

E. Analysis

In order to test the difference in learning outcome and performance, t-tests and ANCOVA models were run using posttest scores and exam grades as dependent variables, respectively.

Firstly, a t-test was used to compare the mean of the change between the posttest and the pretest to look for a statistically significant difference in learning gain. Secondly, an Analysis of Covariance (ANCOVA) was used to estimate the difference between groups on the posttest and exam, after having adjusted for initial differences in the pretest. Similarly, t-tests were performed to compare the exam performance of the two groups, using the grade from the previous introductory programming course (CS1) as a covariate.

IV. RESULTS

One of the assumptions of an ANCOVA test is that the covariate (adjusted pretest score and grade in CS1) does not vary among the groups. The interaction between group and adjusted pretest score was not significant, F(3,37)=12.15, p=0.799, similarly for group and grade in CS1 (F(3,37)=1.45, p=0.0549. Furthermore, conducting the statistical tests, the necessary conditions for normality and homoscedasticity were confirmed using the Shapiro-Wilk test and Levene's test. Since the assumptions were not violated, linear models were created for learning outcome and performance.

A. Learning outcome

The t-test for learning outcome found that there was no improvement or reduction in learning outcome for students that did not have mandatory assignments. Running the t-test on the results of the gain score, yielded no significant difference for these groups, t(39) = 0.672, p = 0.505. The means of both the control group and the experimental group were well inside the 95% confidence interval of these two variables, mainly due to a high standard deviation of the dataset. The t-test tries to explain whether there is a substantial statistically probability that the dataset differs because of the independent variable, the different treatment in assignments that the groups had. Running this test gave the result of it not being statistically probable that the group variable could explain the difference.

In order to verify the result of the t-test, reliability corrected analysis of covariance model was run. This yielded, like the t-test, no statistically significant differences for explaining the posttest scores based on the group (p=0.773, adjusted R^2 = 0.467). Although the model as a whole is statistically significant, the R^2-value comes mainly from the adjusted pretest score, which explains 47% of the differences in the posttest score. The results from the statistical analysis of learning outcome can be seen in Table II.

TABLE II
LEARNING OUTCOME MODEL

Post test score	β	σ	t
Group	-3.17	4.62	-0.69
Adjusted pretest score	0.998	0.165	6.05***
Adjusted R^2	0.469		
F(2,38)	18.65***		

* p<0.05, ** p<0.01, *** p<0.001

Consequently, we argue that there is not enough statistical evidence to accept the alternative hypothesis', so we accept the null hypothesis' that there was no improvement or reduction in learning outcome based on assignment regime.

B. Performance

The results of the t-test yielded no significant change in performance between students who did mandatory assignments and students who did not, t(39) = 0.494, p = 0.624. Like the t-test, the linear model indicated no significant differences in explaining the exam performance in OOP based on the exam performance in CS1 and the group (Table III). Consequently, we argue that there is not enough statistical evidence to accept the alternative hypothesis', so we accept the null hypothesis' that there was no improvement or reduction in performance based on assignment regime.

TABLE III
PERFORMANCE MODEL

Exam grade	β	σ	t
Group	-0.677	0.496	-1.36
Grade CS1	0.403	0.246	1.63
Adjusted R^2	0.038		
F(2,38)	1.80		

* p<0.05, ** p<0.01, *** p<0.001

C. Other models

To see if there were any other factors that could have played out on the results, several statistical tests were conducted with new models. Among others, an analysis of whether the mean of the groups' posttest scores, when looking at the grade in CS1. The result here is interesting, although a statistical t-test showed no significant statistical results, as seen in Table IV.

TABLE IV
LEARNING OUTCOME MODEL BY GROUP AND GRADE IN CS1

Group	N	μ	σ	t
Students with an A in CS1				
Experimental	5	42.5	12.7	
Control	6	58.9	7.63	-1.16
Students with a B in CS1				
Experimental	10	45.2	6.05	
Control	5	55.8	10.9	-0.926
Students with a C in CS1				
Experimental	5	62.0	6.70	
Control	5	51.1	6.49	1.10
Students with a D in CS1				
Experimental	3	55.0	4.27	
Control	2	38.6	0.750	2.931

* p<0.05, ** p<0.01, *** p<0.001

New models were also run with the modified ANCOVA model to see if any other variables better could explain the difference in the posttest score. The grade in CS1 was encoded into two groups of high-performing (A and B) and lower-performing students (C and D), to see whether this variable better explained the differences. This yielded approximately the same results as before, with the adjusted pretest score still being mainly responsible for explaining the difference, although now with an adjusted R^2-value of 0.47 (p=0.791).

Running other models on exam performance yielded similar results. Using the adjusted pretest score as a covariate instead of grade in CS1 increases the R^2-value (0.167); however, the model was still not significant (p=0.198).

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

D. Student feedback

Informal interviews were conducted with some of the students from the experimental group to get some qualitative insights into how the students experienced having no mandatory assignments, how they felt about the exam, and how prepared they felt that they were for further studies.

A majority of these students reported that they followed the assignments that the rest of the class did; however, they enjoyed not having to deliver the assignments. They reported that the lack of mandatory assignments made it more fun to work with the course compared to other classes. As exemplified by this statement:

> *"It has been inspiring to follow a different type of assignment scheme. I have had to work differently, more independently, I have taken responsibility for myself, and I have reacted positively to that. I get to decide for myself how I want to learn and what to learn."*

A number of the students reported that they followed web-based courses to learn the curriculum. Most of these courses were based on small videos explaining a subject and many practical assignments. Many of these felt that they were unsure whether the courses fulfilled the curriculum, and therefore ended up doing more work by looking at the exercises as well. Having to focus on the learning goals, and not assignments, meaning they focused more on what they were supposed to learn, and not just passing tests. For example, one student stated:

> *”I have completed the assignments to learn something, not just because I have to. I think I have learned more by that, and it has been more motivating and fun to work with the course. I’ve looked more at the learning goals of this course.”*

On the other hand, some felt that it was easier to neglect the course when they had other courses with deadlines coming up. For instance:

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

In addition, the students seemed very aware of their personal learning preferences. Many said something alongside, ”it works for me, but not necessarily for everyone else.” The biweekly meetings with TAs were pointed out by many as a good thing to enforce workflow when having to prepare for these meetings. They all mostly agreed that the motivation was high. However, it had gone up and down during the semester. Especially when other courses were deadline heavy, motivation to work with OOP was lower.

V. Discussion

A. Learning outcome

The results yielded no significant change in learning outcome between students that did mandatory assignments and students that were given autonomy and freedom. Assignments are an excellent way to check whether the student has learned something, and for following alongside similar tasks as will be given on the exam. The current study has indicated that measuring whether a student has done the assignment is not necessarily more helpful than merely helping them with the assignments or projects, letting them learn however they like.

The results found in this experiment is consistent with some of the results from the literature on the fact that monitoring assignment completion does not increase learning outcome [16], [24], [26]. These studies also showed no statistically significant difference for similar experiments, with some students having compulsory assignments, and others given more autonomy.

The formative assessment given by the TAs is well known for providing positive results for students that are open for feedback. However, students that are not open for feedback are being spent many resources on checking whether they have done the assignments. These resources could be better spent on more receptive students, focused on teaching the students what they need to learn when they are open for learning it, instead of a fixed schedule for every student that does not provide any autonomy.

The results from the interviews summarize that the majority of the students were happy to be free from mandatory assignments and that they felt this fitted better to their learning style. It is interesting that many chose to follow the assignments, even when not having to do them. It is noted that those who decided to do so did it because they wanted to be sure they learned everything that was related to the exam and not miss out on anything. This shows a considerable focus on the exam and the grade that is given there, while not the most important for a university to teach. The university wants students to have learned the learning goals of the course, and the exam is a summative way to measure that. Many things in a course are not asked about on the exam, due to time or practical constraints, and students choose not to focus their time on such knowledge. This is also consistent with previous findings, where students still delivered the assignment when only a select few were graded [26].

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

B. Performance

The results looking at the difference between the grades are particularly interesting, even though there were not a statistically significant enough difference. The assumption was that more autonomy and more freedom would be better for the higher-performing students, which manage to learn on their own, and are not in a significant need for guidance. However, the results indicate the exact opposite, with A and B students in the experimental group getting outperformed by A and B students in the control group, and the opposite for C and D students. Contrary to popular belief, that may indicate that students that lower-performing students may not require as much guidance, but rather need autonomy to work at their own pace, instead of being forced through a specific set of assignments. It could also mean that lower-performing students might find other and more unethical ways to complete assignments. As they are unable to do them, when given more autonomy, they complete the assignments without having the pressure of a deadline. This result is also more consistent with the findings from Haugan and Lysebos study, where the lower performing students in the pretest did even better on the exam [15]. It should be noted that the number of observations within each grade is very low, and a higher number of participants is needed to get a more meaningful result. It may also be that stronger students attribute more of their learning to the exam period and learn more in a shorter period of time, and therefore have delayed more of the work until the end of the semester.

When discussing lower and higher performing students, it should also be discussed how to allocate resources per student. In the current assignment system, all students have to meet their TA to demonstrate their code and understanding. They may meet as often as they want to get help in understanding the assignment and complete it. With resources that could focus less on the approval of exercises and more on teaching and guiding students, resources could be further utilized by the students that need them. Some mechanisms might also be in place to get the lower performing students to use the available resources. There will always be students that do not put in the effort needed when not giving strict guidelines for what and how to learn and when to deliver. The discussion should, therefore, be on whether it is more important to provide more autonomy to the students who want it than to force everyone through the same mandatory arrangement.

C. Implications

The results of this experiment have indicated there is no statistical difference in learning outcome or performance for students with mandatory assignments, and for students without mandatory assignments. Feedback from the students has also indicated that students being released from deliverance of compulsory assignments will do the assignment nevertheless. They do so because they want to learn the subject and prepare for the final exam. It is unsure whether they would have done that if they knew that their classmates not necessarily had done these assignments. It could be that when they knew everyone else had to do these, they were afraid of falling behind. Whether they did exercises or not, the result indicated that the average time spent on the course per week was less for the students not having to do assignments, even though they achieved the same learning outcome.

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

As multiple studies pointed out [2], [6], [7], the summative feedback of delivering homework or exercises does not give benefits for the students, and the results of this experiment support these statements. Assignments are a helpful tool for preparing students for the exam, guiding them into learning more about the curriculum of the week, and measuring their progress, but the assessment of the exercises does not necessarily benefit the students. It is interesting that a majority of the students in the control group believed otherwise, and that should also be taken into consideration before launching an all-out experiment testing such an arrangement. Lastly, a reasonable question to ask is whether one should consider grading the assignments and including these marks in the final grade. In this case, the Norwegian university law prohibits using TAs for grading that counts towards the final grade, which makes it nearly impossible to implement in a course with 700 students and 10 assignments. On the other side, there are course designs that could incorporate more formative grading throughout the course and these results on mandatory assignments can help inform these design regardless of grading scheme.

This experiment has been conducted on students from different study programs. All study programs have a high focus on computer science but are built up in different ways. Different study programs may learn and be motivated by different things, and this is important to keep in mind when designing a class. Students from different study programs may have a different learning style, while the same can also be said of students from the same study program. Designing a university course for different learning styles means having to give up inflexible systems for adaptable ones.

The most important implication is the need to give engineering students the best tool and guidance for learning and studying, and to educate the engineers for tomorrow. The world needs technologists in the future with the ability to learn and adapt, and educational institutions should take their part when it comes to finding the best possible way of teaching computer science.

Given that assignments, or at least mandatory assignments, do not seem to be any help in students learning, the focus onward should be on how students study and learn, and what is the best way to aid in their learning process. The students approach to learning could be helped along by various exercises or assignments, be them mandatory or not, to guide in this process. The choice of method could be exercises, group projects, pair programming, or other practical tools for teaching computer science. However, if only given compulsory assignments, that will not leave room for self-study and for learning styles that are not aligned towards exercises as a learning activity.

D. Limitations

Due to the quasi-experimental nature of the experiment, the small number of participants, and a variety of other factors, many biases could have affected the results of this experiment. Students may be colored by their experiences with other courses, and their extensive use of mandatory assignments in other classes parallel to the trial in this course. They may thus be tired of deadline sprints and give a more positive review to different types of learning approaches than what they usually would do.

Students who signed up for the control group have chosen to not sign up for the experimental group, and have as such chosen to do the assignments themselves. They would naturally be motivated to do assignments and are typically among the most motivated students. The same goes for the experimental group, especially when it comes to learning outcome, that they might be the type of students that learn best when given autonomy and freedom, and as such, does not represent the entirety of the student population sufficiently. The fact that many of them chose to do assignments anyway leads to thinking that they want assignments to learn anyway, and as such, discredits that bias. As for the entire experiment, conducted with such a low number of students, there are significant reasons why the result could be as it is. The students following alongside know full well that they work best given autonomy, and therefore signed up for the experimental group. The control group, while given the option of freedom, chose to follow alongside a strict schedule. There are, of course, outliers here, with the probability that several of the participants signing up for the experimental group because they did not want to do assignments, and wanted to have more free time and do less work throughout the semester.

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

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

VI. Conclusion

In summary, the experiment found that there were no statistically significant differences between learning outcome and performance for students following a mandatory assignment program and students that were given more autonomy to obtain the necessary course knowledge through their own means. This result indicates that mandatory assignments are not necessarily helpful for learning the course in introductory programming courses. The implication of this is that there should be a consideration of whether resources going into grading assessments are better spent otherwise. Assignments are also given out to test curriculum that can not be tested on the exam, but the emphasis on how much of the course is assignments, and how much is self-study should be reconsidered. Assignments along the semester helps to push students into effective study and learning behavior, and give them goals to work towards that are not too far into the future, like the exam. However, there should be more focus on formative evaluation and self-study throughout the semester.

Reducing the number of mandatory assignments in a course can be one way of bringing together the best of both worlds, avoiding students' procrastination while at the same time giving them time to focus on learning the curriculum through self-study. This study does not aim to get rid of assignments all together, as exercises are beneficial for gaining knowledge, and knowing what you have learned and what you have missed. However, collecting and grading the assignments may not be as helpful as we once thought.

References

[1] P. Lauvås jr. and T. Sandnes, "Mandatory coursework in higher Norwegian IT education," in *Norwegian Conference for Education and Didactics in IT subjects*, Narvik, Norway, 2019. [Online]. Available: https://ojs.bibsys.no/index.php/NIK/article/view/648

[2] J. Hattie and H. Timperley, "The power of feedback," *Review of educational research*, vol. 77, no. 1, pp. 81–112, 2007.

[3] T. Koulouri, S. Lauria, and R. D. Macredie, "Teaching introductory programming: A quantitative evaluation of different approaches," *ACM Transactions on Computing Education (TOCE)*, vol. 14, no. 4, p. 26, 2015.

[4] R. W. Tyler, R. M. Gagné, and M. Scriven, *Perspectives of curriculum evaluation*. Rand McNally, 1967, vol. 1.

[5] B. S. Bloom, "Some theoretical issues relating to educational evaluation," *Educational evaluation: new roles, new means: the 63rd yearbook of the National Society for the Study of Education*, no. part II, pp. 26–50, 1969.

[6] T. J. Crooks, "The impact of classroom evaluation practices on students," *Review of educational research*, vol. 58, no. 4, pp. 438–481, 1988.

[7] P. Black and D. Wiliam, "Assessment and classroom learning," *Assessment in Education: principles, policy & practice*, vol. 5, no. 1, pp. 7–74, 1998.

[8] A. Vihavainen, J. Airaksinen, and C. Watson, "A systematic review of approaches for teaching introductory programming and their influence on success," in *Proceedings of the tenth annual conference on International computing education research - ICER '14*. Glasgow, Scotland, United Kingdom: ACM Press, 2014, pp. 19–26. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2632320.2632349

[9] T. Koulouri, S. Lauria, and R. D. Macredie, "Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches," *Trans. Comput. Educ.*, vol. 14, no. 4, pp. 26:1–26:28, Dec. 2014. [Online]. Available: http://doi.acm.org/10.1145/2662412

[10] B. Cheang, A. Kurnia, A. Lim, and W.-C. Oon, "On automated grading of programming assignments in an academic institution," *Computers & Education*, vol. 41, no. 2, pp. 121–131, 2003.

[11] A. Luxton-Reilly, Simon, I. Albluwi, B. A. Becker, M. Giannakos, A. N. Kumar, L. Ott, J. Paterson, M. J. Scott, J. Sheard, and C. Szabo, "Introductory Programming: A Systematic Literature Review," in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE 2018 Companion. New York, NY, USA: ACM, 2018, pp. 55–106, event-place: Larnaca, Cyprus. [Online]. Available: http://doi.acm.org/10.1145/3293881.3295779

[12] M. Lykke, M. Coto, S. Mora, N. Vandel, and C. Jantzen, "Motivating programming students by problem based learning and LEGO robots," in *2014 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2014, pp. 544–555.

[13] N. Avouris, S. Kaxiras, O. Koufopavlou, K. Sgarbas, and P. Stathopoulou, "Teaching introduction to computing through a project-based collaborative learning approach," in *2010 14th Panhellenic Conference on Informatics*. IEEE, 2010, pp. 237–241.

[14] D. C. Cliburn, S. M. Miller, and E. Bowring, "Student preferences between open-ended and structured game assignments in CS1," in *2010 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2010, pp. F2H–1–F2H–5.

[15] J. Haugan and M. Lysebo, "Hvorfor antall arbeidskrav bør reduseres," *Uniped*, vol. 41, no. 03, pp. 347–360, 2018.

[16] B. Gutarts and F. Bains, "Does Mandatory Homework Have a Positive Effect on Student Achievement for College Students Studying Calculus?" *Mathematics and Computer Education*, vol. 44, no. 3, p. 232, 2010.

[17] S. Areepattamannil and B. Kaur, "Factors predicting science achievement of immigrant and non-immigrant students: A multilevel analysis," *International Journal of Science and Mathematics Education*, vol. 11, no. 5, pp. 1183–1207, 2013.

[18] J. R. Cheema and K. Sheridan, "Time spent on homework, mathematics anxiety and mathematics achievement: Evidence from a US sample," *Issues in Educational Research*, vol. 25, no. 3, p. 246, 2015.

[19] W.-C. Mau and R. Lynn, "Gender differences in homework and test scores in mathematics, reading and science at tenth and twelfth grade," *Psychology, Evolution & Gender*, vol. 2, no. 2, pp. 119–125, 2000.

[20] R. De Jong, K. J. Westerhof, and B. P. Creemers, "Homework and student math achievement in junior high schools," *Educational research and Evaluation*, vol. 6, no. 2, pp. 130–157, 2000.

[21] J. Wilson and J. Rhodes, "Student Perspectives on Homework." *Education*, vol. 131, no. 2, 2010.

[22] J. D. House, "The effect of student involvement on the development of academic self-concept," *The Journal of Social Psychology*, vol. 140, no. 2, pp. 261–263, 2000.

[23] E. F. Christensen, *The effect of homework choices on achievement and intrinsic motivation*, 2002.

[24] G. Weems, "The impact of homework collection on performance in intermediate algebra," *Research and Teaching in Developmental Education*, pp. 21–25, 1998.

[25] C. M. Cartledge and J. E. Sasser, "The Effect of Homework Assignments on the Mathematics Achievement of College Students in Freshman Algebra." 1981.

[26] E. Miller and G. Westmoreland, "Student response to selective grading in college economics courses," *The Journal of Economic Education*, vol. 29, no. 3, pp. 195–201, 1998.

[27] H. Danielsiek and J. Vahrenhold, "Stay on these roads: Potential factors indicating students' performance in a CS2 course," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. ACM, 2016, pp. 12–17.

[28] V. Hellem, "The Effect of Mandatory Assignments on Students' Learning Outcome and Motivation in Introductory Programming Courses," Master's thesis, NTNU, Trondheim, 2019. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2625825

[29] L. Cohen, L. Manion, and K. Morrison, *Research Methods in Education*, 5th ed. London, UK: Routledge, 2002.

[30] D. T. Campbell and J. C. Stanley, *Experimental and quasi-experimental designs for research*, 2nd ed. Boston: Houghton Mifflin Comp, 1967, oCLC: 247359300.

[31] Trochim, Donnelly, and K. Arora, *Research Methods: The Essential Knowledge Base*. Cengage Learning, Jan. 2015.