Jørgen Thorsnes

# Teachers' attitudes and self-efficacy towards teaching programming and the impact of continuing education

## A case study on in-service teachers' attitudes and self-efficacy towards teaching programming, and the impact of continuing education in programming on teachers self-efficacy in teaching programming

Master's thesis in Natural Science with Teacher Education

Supervisor: Monica Divitini

June 2020

Master's thesis

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

NTNU
Kunnskap for en bedre verden

Jørgen Thorsnes

# Teachers' attitudes and self-efficacy towards teaching programming and the impact of continuing education

## A case study on in-service teachers' attitudes and self-efficacy towards teaching programming, and the impact of continuing education in programming on teachers self-efficacy in teaching programming

Master's thesis in Natural Science with Teacher Education
Supervisor: Monica Divitini
June 2020

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

With the new curriculum in Norway and as a part of the Norwegian government's strategy for digitalization of the primary and secondary education, programming has been included as a something for all pupils to learn in school. Programming has been included into subjects like math, natural science, music, and arts and crafts, and teachers will have to acquire competence in programming to be able to teach it in school. For this reason, there is a need for research on how teachers feel towards teaching programming in school, and how well prepared teachers are for teaching programming. Self-efficacy refers to the belief in one's capability to perform the actions required for producing a given outcome. This thesis explores the attitudes and self-efficacy of teachers towards teaching programming, and the impact programming education might have on teachers' self-efficacy in teaching programming.

The aim of the research is two-fold: (1) supporting educational institutions in the education of new programming teachers, and (2) identifying challenges that emerge from the inclusion of programming in the curriculum. With this aim in mind, the research is set to answer the following research question: *How do in-service teachers with programming education perceive their attitudes and self-efficacy towards teaching programming, and the impact of programming education on their self-efficacy towards teaching programming?*

The research is based on a flexible design, and a qualitative approach within the case of a continuing education program in programming for in-service teachers. The research uses an explorative approach consisting of (1) a study of reflection notes written by in-service teachers as a part of the final delivery and assessment in the 2018-19 continuing education program; (2) a main interview study with ten teachers from the 2018-19 continuing education program; (3) a supplementary interview study with three teachers from the 2019-20 continuing education program. Both the study on the reflection notes and the interview studies look into the teachers' attitudes and self-efficacy towards teaching programming, as well as the impact of programming education on the teachers' self-efficacy in teaching programming. The main interview study also looks into the perceived effect of the programming education on the teachers' self-efficacy in teaching programming over time. The results are discussed in the context of self-efficacy and in light of relevant research in the domain of teaching programming.

The results indicate that the teachers in the study are positive towards programming in school, and perceive it as relevant for pupils to learn. There is a worry on how programming is included in the curriculum in Norway in that many teacher without competence in programming will have to teach it. It is also indicated that the teachers have a relatively high sense of self-efficacy in teaching programming. The teachers have lowest self-efficacy towards assessment in programming, and express a need for for quality tools and methods for assessment in programming. There is an indication that teachers don't need to be expert programmers to feel confident in their teaching of programming. It is indicated that contuining education in programming can improve teachers' self-efficacy in teaching programming. It is also indicated that the teachers' self-efficacy has increased with experience over time, and that it does not significantly decrease without experience over time. The teachers' perceived programming skill seem to lower over time when not used. It is indicated a need for more competent teachers of programming and local communities of practice for teachers of programming.

# Sammendrag

Med den nye læreplanen i Norge og som en del av den norske regjeringens strategi for digitalisering av norsk skole har programmering blitt innført som noe alle elever i skolen skal lære. Programmering har blitt inkludert inn i læreplanen til fag som matematikk, naturfag, musikk, og kunst og håndverk, og lærere blir nå nødt til å tilegne seg kompetanse i programmering for å kunne bruke og underivse i programmering i skolen. Det er derfor et behov for forskning på hvordan lærere opplever inkluderingen av programmering i skolen, og hvor forberedt de føler seg for å skulle undervise og bruke programmering. Begrepet *self-efficacy* handler om ens tro på egen evne til å utføre spesifikke handlinger som er nødvendige for å oppnå et gitt resultat. Denne avhandlingen utforsker læreres holdninger og self-efficacy inn mot å undervise programmering, og hvordan videreutdanning i programmering kan påvirke læreres self-efficacy i programmeringsundervisning.

Forskningens hensikt er todelt: (1) støtte utdanningsinstitusjoner i utdanningen av nye programmeringslærere, og (2) identifisere utfordringer som oppstår med innføringen av programmering i læreplanen. Med dette som utganspunkt sikter forskningen mot å svare på følgende forskningsspørsmål: *Hvordan ser lærere med utdanning i programmering på sine holdninger og self-efficacy i programmeringsundervisning, og hvordan utdanningen i programmering har påvirket deres self-efficacy i programmeringsundervisning?*

Forskningen er basert på en fleksibel design, og en kvalitativ tilnærming innrammet i et videreutdanningsstudie i programmering for lærere som en case for forskningen. Forskningen har en utforskende tilnærming bestående av (1) en studie av refleksjonsnotater skrevet av lærere som en del av siste innlevering og vurderingsgrunnlaget i videreutdanningsstudiet i 2018-19; (2) en hoved-intervjustudie med ti lærere som tok videreutdanningsstudiet i 2018-19; (3) en supplerende intervjustudie med tre lærere som tok videreutdanningstudiet i 2019-2020. Både studien av refleksjonsnotater og intervjustudiene ser på lærernes holdninger og self-efficacy inn mot å undervise programmering, i tillegg til hvordan videreutdanningen har påvirket lærernes self-efficacy inn mot å undervise programmering. Hoved-intervjustudien ser også på den oppfattede effekten av videreutdanningen på lærernes self-efficacy over tid. Resultatene diskuteres i kontekst av self-efficacy og i lys av relevant forskning innen undervisning av programmering.

Resultatene indikerer at lærerene er positive til programmering i skolen, og oppfatter programmering relevant for elevenes læring. Det er en bekymring mot hvordan programmering blir innført i læreplanen i det at mange lærere uten nødvendig kompetanse nå blir nødt til å underivse programmering. Resultatene peker på at lærerne har relativt høy self-efficacy i programmeringsundervisning. Lærerne har lavest self-efficacy når det kommer til vurdering i programmering, og uttrykker et behov for vurderingsverktøy og metoder i programmering. Resultatene peker på at lærere ikke behøver å være eksperter i programmering for å føle seg selvsikre i sin undervisning av programmering. Resultatene indikerer at videreutdanning i programmering kan forbedre læreres self-efficacy i programmeringsundervisning. Det indikeres også at læreres self-efficacy i programmeringsundervisning har økt med erfaring over tid, og at den ikke har noen signifikant nedgang over tid uten erfaring. Lærernes oppfattede programmeringsferdigheter virker å bli dårligere over tid når de ikke brukes. Resultatene indikerer et behov for flere kompetente programmeringslærere og lokale fagfellesskap for programmeringslærere.

# Preface

This thesis marks a somewhat strange end to my teacher education in informatics and mathematics at NTNU. Five years ago, I started the studies in Natural Science with Teacher Education along with many of those who now are my friends and fellow future teachers. While I write these words, I should have been sitting among my friends and fellow students in our study room at the university, sharing these last moments of our studies together. But I am not. It has been a strange semester. My main focus this semester should have been on my master's thesis, but I must admit that the coronavirus has stolen the focus of attention on many occasions. I actually believed that working in solitude without distractions would sharpen my focus and effectiveness in the work on the master's thesis, but I was very wrong. I'm still glad to look back at five wonderful years of studies, filled with learning and a rich social life, and I would argue that the joy of learning and research comes from sharing the process with others.

I would first like to give a big thanks to my supervisor Monica Divitini for invaluable guidance and understanding throughout the process of this research. Her expertise and follow-up have guided me through the process, and her understanding of the situation this semester have eased my mind and kept me working on the research in frustrating times.

I would also like to thank my co-supervisor Majid Rouhani for providing me with data, information, as well as thorough and specific feedback.

I would also like to give my gratitude to the teachers involved in the interview studies, there would be no thesis without them.

Finally, I would like to thank my buddy and roommate Robert for all the good times shared while working on our master's theses from home in our apartment.

Nøtterøy, June 2020

Jørgen Thorsnes

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| ICT | Information and communications technology |
| IT | Information technology |
| NTNU | The Norwegian University of Science and Technology |
| STEM | Science, Technology, Engineering and Mathematics |
| TSE | Teacher self-efficacy |
| TSES | Teachers' Sense of Efficacy scale |

# 1 Introduction

In the last twenty years, computers and technology have become a big part of our everyday lives. Our cars, kitchens, clocks, phones, all have been equipped with computers and program code. Computers and programming have definitely made an impact on our society in the past 20 years. Even though almost all of us are using computers, not many of us know how to program computers to do all the things they do for us. Several countries have now included, and are including, technological subjects with programming and computational thinking into their curriculum. The reasons why programming is now being included in many countries school curriculum are many. Sanne et al. (2016) summarize the arguments used by different european countries in five categories: Promote logical thinking, promote problem solving skills, attract pupils to ICT, promote coding skills, recruit future IT workers, promote other important competence. Programming already have a history in the educational perspective. For example, Seymour Papert, introduced LOGO in 1980, an educational programming environment for school children. Papert explained "The child programs the computer. And in teaching the computer how to think, children embark on an exploration about how they themselves think." (Papert, 1980, p. 414). It seems that now, 20 years later, programming and computational thinking are quickly on their way into the educational system. This is also the case in Norway. In late 2019 and early 2020, the new curriculum for the primary education (grade 1-13) in Norway was published (Utdanningsdirektoratet, 2020). In the new curriculum, programming and computational thinking have been included into several subjects such as math, natural science, arts and crafts, and music.

In a report on the status of the informatics education in Europe by The Committee on European Computing Education, it is reccomended that all students must have access to ongoing education in informatics in the school system, and that the teaching of informatics must be undertaken only by teachers who have formal education in informatics (Vahrenhold et al., 2017). The committee also recommend that the shortage of informatics teachers needs to be addressed by training and hiring informatics teachers, and that hiring of informatics teachers must follow the same standard as for all other disciplines without sacrificing formal requirements or methodological training.

In a report on technology and programming in school by Sanne et al. (2016), it is suggested that a compulsory subject in technology and programming is created in Norwegian schools. One of the reasons behind this suggestion is that technology is often downprioritized when embedded into other subjects, as well as teachers not feeling competent enough in teaching technology (Sanne et al., 2016). However, such a compulsory subject has not been created in the Norwegian school system, and programming has been included into existing subjects. Many teachers will most likely not feel competent enough in teaching programming in their respective teaching subjects.

## 1.1 Motivation

Many teachers will have to take continuing education or programming courses to prepare themselves for teaching programming. Both programming in school and programming courses for in-service teachers in Norway are still in their early years. New teachers of programming in primary, secondary and high school will have new challenges to face in their classrooms. To explore how teachers feel towards the recent inclusion of programming in the curriculum and how well prepared they are for teaching after taking programming education, the term of self-efficacy can help look into this field of interest. Self-efficacy refers to the belief in one's capabilities to organize and execute the courses of action required to produce a given result (Bandura, 1997, p. 3). In terms of teaching programming, the teacher's self-efficacy refers to their belief in their capabilities to teach programming, such that pupils achieve the desired learning outcomes. By exploring teachers' self-efficacy in teaching programming, and how programming education affects their self-efficacy, we can get an impression of what challenges teachers expect to meet, what challenges they are already facing, and how well prepared they feel to face these challenges after taking programming education.

Positive self-efficacy beliefs have been connected to both increased student and teacher outcomes, as well as positive influence on teachers' phycological well-being (Zee & Koomen, 2016). In two recent Swedish studies on teachers attitudes and self-efficacy towards programming, the researchers found that many Swedish teachers had a lack of confidence in teaching programming (Hartell, Doyle, & Gumaelius, 2019), and that the teachers felt insecure when it came to solving problems with programming (Mannila, Nordén, & Pears, 2018). Similar results were also found in England after they introduced the subject Computing in 2014, where 60% of the teachers in a survey felt not competent in teaching the new computing curriculum (YouGov, 2015). In another report three years later, in 2017, 48% of the teachers still reported low confidence due to lack of "sufficient theoretical and technical knowledge of computing that included aspects of programming and coding" (Royal Society, 2017, p. 55). There is reason to believe that the situation might be similar to this in Norway, and while it would also be interesting to explore how confident Norwegian teachers feel in teaching programming, it is also interesting to see how programming education affects their confidence, or self-efficacy, in teaching programming. Teachers with little experience and knowledge of programming will most likely have varied or little self-efficacy in teaching programming and will also have varied or little insight into what challenges that comes with teaching programming. Teachers with education in programming and some experience in teaching programming will have more insight into the different aspects of teaching programming. By exploring these teachers' self-efficacy in teaching programming, one can explore the challenges and opportunities that comes with the inclusion of programming into the curriculum, and better prepare new teachers of programming, as well as giving an indication of what is important to focus on in the education of prospective teachers and continuing education of in-service teachers of programming.

With this motivation in mind, this thesis is meant to support educational institutions, as well as initiatives that aims at helping teachers of programming, by indicating what can be beneficial to focus on in the training of new programming teachers. This thesis also aims at identifying challenges that emerge from the inclusion of programming in the curriculum, and by this indicating what areas of research might be needed in the years to come, in order to support educational policies in regard to programming in school.

## 1.2 Context

This study has been done as a master's thesis at the Department of Computer Science at the Norwegian University of Science and Technology (NTNU). The thesis is the work of a student in Natural science with Teacher Education (NTNU, 2020c) with ICT as the main field of study. The study looks at teachers with continuing education in programming's perceived self-efficacy and attitudes towards teaching programming. The research is based in the case of a continuing education program in programming for in-service teachers, at NTNU.

## 1.3 Problem and Research questions

As described, teachers in Norway and many other countries are going to teach and use programming in the years to come. Many of them have little experience and competence, and there is little research on how to deal with the challenges of teaching programming in today's educational system as this is a relatively new area of practice and research. One way of preparing teachers to teach programming can be by continuing education. Programming in the Norwegian educational system is relatively new and uncharted territory. What challenges do the teachers need to be prepared to face, and do they feel ready to face these challenges after taking programming education? Do the teachers find it relevant to use programming in school? This thesis explores these problems through the research questions seen below.

**Main research question:** *How do in-service teachers with programming education perceive their attitudes and self-efficacy towards teaching programming, and the impact of programming education on their self-efficacy towards teaching programming?*

The main research question is explored through the sub-questions seen below.

**RQ1:** What attitudes do in-service teachers with programming education have towards programming in school?

**RQ2:** How do in-service teachers with programming education perceive their self-efficacy in teaching programming?

**RQ3:** How do in-service teachers perceive that programming education has affected their self-efficacy in teaching programming?

**RQ4:** How do in-service teachers perceive the lasting effect of programming education in regard to their self-efficacy in teaching programming?

## 1.4 Method

This study is based on a flexible design as described in Robson and McCartan (2016, chapter 7), and it explores teachers own feelings and reflections towards teaching programming. In flexible designs data is often non-numerical, usually in the form of words, and often also a focuses on the participants views and opinions (Robson & McCartan, 2016). The study can be described as a case study since it focuses on a specific study program and in-service teachers taking this study program. In case studies, the case can for example be a group or an organization (Robson & McCartan, 2016). In this case the research focuses on in-service teachers that have participated or are participating in the continuing educational program in programming (NTNU, 2020d) at NTNU. At the same time as this can be described as a case study, the use of the teachers and reflection notes from the NTNU continuing education program in programming can also be described as a convenience sample. Convenience sample is one of the most commonly used non-random sampling techniques, and can be described as a group of participants who are easily accessible to participate in a given study (Fraenkel, Wallen, & Hyun, 2011). There are some disadvantages to this sampling technique, but it may be used when it is difficult to select a random or systematic nonrandom sample

(Fraenkel et al., 2011). As there are currently not existing many study programs in programming for teachers in Norway, in addition to the fact that the researcher in this study worked as a learning assistant in the continuing education program in programming for teachers in 2019 and thus had easy access to the study program, this specific case was chosen.

The research questions imply that the aspect of time has to be taken into consideration in the research process, as it tries to answer how the teachers' self-efficacy have been affected by the programming studies and also by the time that has passed since they finished the studies. A more optimal method in this case could have been to somehow measure the teachers' self-efficacy at different points of time, for example before their programming studies, right after their studies, and one year after they finished their studies. Due to this thesis being done in one semester, in the spring of 2020, such an approach was unfortunately impossible to go through with. A more explorative approach was chosen.

The study is based on an initial analysis of the study program description and material to get an overview of the context; an in-depth analysis of the reflection notes delivered by the students (the in-service teachers) as part of the final delivery in the study program in 2018-19; semi-structured interviews with ten of the students that wrote the reflection notes; and three supplementary interviews with students that were currently participating in the 2019-20 study program.

The reflection notes were chosen as a source of data, as they could indicate how the teachers felt towards their own learning in their studies, their results, and perceived challenges and confidence in their teaching of programming at the end of their programming studies. The reflection notes have been analyzed with a method inspired by grounded theory (Strauss & Corbin, 1998). This method was chosen due to the inconsistency of the reflection notes. With the grounded approach in the coding process, the researcher has tried to make the data manageable and find the relevant parts of the data through categories, relations, and themes that emerged from the analysis.

To get more in-depth into the results from the analysis of the reflection notes, as well as exploring how in-service teachers perceive their self-efficacy in teaching programming and the impact of the continued education on their self-efficacy, there was also conducted interviews with teachers that participated in the continuing education program in 2018-19. The researcher used semi-structured interviews to explore the research questions by capturing the teachers own reflections on their self-efficacy towards teaching programming. Due to COVID-19 the interviews were conducted through the online video communication application Zoom, recorded on an external recording device and transcribed by the researcher. The transcriptions were analyzed with two methods. First, thematic coding in regard to parts of a self-efficacy scale by Tschannen-Moran and Hoy (2001), then with a similar approach as with the coding of the reflection notes.

To be able to see how the results transfer to a different cohort of students and also to some extent address the validity of the study, the researcher has also interviewed three of the students participating in this year's (2019-2020) study program. The interviews and analysis were conducted in a similar way as with the interviews with the teachers from the 2018-19 study program.

## 1.5 Ethical aspects in the research

When carrying out real world research involving people, there are ethical aspects to take into consideration. Ethics is a process, and should be reviewed throughout the research process, not as a stand-alone, one time only, pursuit (Robson & McCartan, 2016, p. 208). The Norwegian Research Ethics Committees for Social Sciences and the Humanities, NESH, has developed research ethics guidelines for the social sciences, humanities, law and theology (NESH, 2016). Research pursuits the truth, but at the same time, it can never fully achieve this goal. In the humanities and social sciences, interpretations are often a part of the research process, and different approaches and theoretical positions may allow for different and at the same time reasonable interpretations of the same material (NESH, 2016). In the work of this thesis, the researcher has done interpretations of data material, and it has been important throughout the process to reflect upon how the researchers own values and attitudes can affect the choice of topic, data sources, and interpretations. For example, in this research, it has been important for the researcher to be aware of his own values and attitudes towards the relevance of teaching programming in school when interpreting the data and conducting the interviews. Even though it is impossible to completely put aside one's subjective values and attitudes when doing research, it is can be beneficial to be conscious about these aspects in the research process (Postholm, 2005). With this in mind, and as in most research, the results and interpretations in this thesis must be seen in some grade as contingent and limited, and not universally correct.

This research also deals with individuals in both the analysis of the reflection notes and the interviews. When dealing with individuals in research, one must take into account the ethical aspects of human dignity, privacy, duty to inform, consent, and confidentiality. These aspects have been reflected upon throughout the research, and at the same time as it is difficult to ensure them, it is important for the researcher to have them in mind. The reflection notes were anonymized by the institution responsible for them, before they were made available to the researcher. Both in the interviews and in the reflection notes, statements that indicate the teachers' school, living area, name, or personal information like health and medical related information have been anonymized or not transcribed, and are not used in this research. Interview studies are also filled with moral and ethical questions. This is written more about in the chapter 5.2.7 where ethical aspects in interview studies are described.

## 1.6 Results

The results indicate that the teachers perceive that the programming education has had a positive impact on their self-efficacy towards teaching programming. Many of the teachers did not feel capable of teaching programming before their studies, but express that they feel capable of teaching and using programming in their respective teaching subjects and grades after the programming studies. Some of the teachers report that the studies have also had a positive impact on their attitudes towards programming.

The teachers in this study are very positive towards programming in school, and perceive it as relevant for their pupils to learn and to use interdisciplinary. There is however an indication that the teachers are worried on how programming is included into the curriculum in Norway. Some of the teachers express a need for more competent teachers of programming. Some experience negativity towards programming among their colleagues. The teachers also find collaboration with other teachers of programming useful, and some want more colleagues with programming competence to work with, or communities of programming teachers.

The teachers in this study have lowest self-efficacy when it comes to assessment in programming. There is an indication that there is a lack of quality tools and methods for assessment in programming. Some of the teachers express that oral presentation is one of the better methods of assessment in programming.

Many of the teachers perceive their programming skill as relatively low, and many find learning programming hard. This does not seem to have a significant impact on their self-efficacy towards teaching programming, as most feel their programming skill is sufficient for their own teaching of programming. Some teachers find what and how to teach programming challenging, and there is an indication that there is a need for specifying what is relevant for the pupils to learn in programming. Even though some teachers express that they find how to teach programming challenging, most of the teachers seem to have a relatively high sense of self-efficacy when it comes to both conveying knowledge and developing suitable teaching material.

The teachers report that their self-efficacy in teaching programming increase with experience in teaching programming, and does not significantly decrease without experience over time. The teachers report that their programming skill lowers over time when not used, but that they can easily refresh their programming skills when needed. The teachers still find it important to maintain and further develop their programming skills.

The results indicate that continuing education in programming of in-service teachers can be a suitable way of preparing them for teaching programming in their respective teaching subjects and grades, as it can have a positive impact on their self-efficacy in teaching programming. It can also have a positive impact on their attitudes towards programming in school.

## 1.7 Outline

To help the reader there are two semantic issues to clarify in this thesis. When using the word "teacher" in this thesis, this refers to teachers in the primary education (grades 1-13). The word "lecturer" is used for the lecturers/teachers at university level, to avoid misunderstandings that might arise from the fact that the students in the case used in this study are also in-service teachers. The term "secondary school" refers to grades 8-10 in the Norwegian school system, and the term "high school" refers to the grades 11-13 in the Norwegian school system.

This thesis consists of 8 chapters. Chapter 2 provides a description of definitions used in this study, as well an overview on self-efficacy, and relevant research on teacher self-efficacy in the domain of programming and teaching programming. Chapter 3 describes the case, namely the continuing education program in programming at NTNU for in-service teachers.

Chapter 4 describes the study on the reflection notes written by the students in the continuing education program. Chapter 5 describes the main interview study with the teachers that participated in the 2018-19 continuing education program. Chapter 6 describes the supplementary interview study with the teachers that participated in the 2019-20 continuing education program.

In chapter 7, the researcher discusses the results from the study on the reflection notes and the interview studies in light of relevant research. Chapter 7 also provides a discussion on the implications of the results. Chapter 8 presents a summary of the research, and the conclusion of the research questions. Chapter 8 also present the contribution of this thesis, as well as what the researcher perceives as relevant to focus on in further work.

# 2 Programming and teacher self-efficacy

## 2.1 Chapter overview

This chapter aims to give the reader insight into the definitions and literature used in the research. The chapter presents definitions on programming, coding, computational thinking, as well as the term teacher self-efficacy. This chapter also present a short overview on teacher self-efficacy research. Lastly, it presents relevant research on teachers' attitudes and self-efficacy in the domain of programming and teaching programming.

## 2.2 Programming, coding, and computational thinking

Programming has been included in the Norwegian curriculum, but the term "programming" is rather open, and can be interpreted in different ways. In regard to the inclusion of programming in the Norwegian curriculum and this thesis, there are three different terms that need to be addressed and specified: Programming, coding, and computational thinking. This sub-chapter describes how programming, coding, and computational thinking is defined in this thesis.

### 2.2.1 Programming

Programming is often associated with writing computer code, but the term encompasses more than that. Programming is the process that embraces the development of a program that can be executed by a computer (Guzdial, 2015). Programming is "the process of identifying a problem and devising possible solutions to the problem, and then writing a code that can be understood by a computer, while also troubleshooting and continuously improving that code" (Sevik, 2015, p. 9, translated from Norwegian)

### 2.2.2 Coding

The terms coding and programming are often used interchangeably, and the term coding has become more prevalent in daily speech in recent years, especially with regard to children and young people learning programming, for example through the voluntary Norwegian initiative Lær Kidsa Koding (Lær Kidsa Koding, 2020). In this thesis, the term coding means constructing program code that can be executed by a computer. This involves writing both text-based programming languages (for example Python, Java, C#), and block-based programming languages (for example Scratch or Micro:Bit).

Coding is thus a part of programming and is the part of programming that deals with writing the program code. One can think of coding as formulating solutions to problems so that the solution can be executed by a computer.

### 2.2.3 Computational thinking

Computational thinking can be defined as the thought process involved in the formulation of problems and their solutions, so that the solutions can be effectively executed by an information-processing agent (Wing, 2011). An information-processing agent can be both a human and a machine.

Computational thinking is a problem-solving method/process/strategy that deals with breaking down complex problems into smaller solvable sub-problems, logically organizing and analyzing information and creating step-by-step procedures for solving problems. It also deals with abstracting, modeling, and generalizing solutions so that they can be applied to similar problems (Utdanningsdirektoratet, 2019)

Computational thinking is thus a kind of generalization/abstraction of the programming process. It concerns how one can go about solving problems, not only with a computer and in computer science, but by using programming methods and principles to solve problems in general.

## 2.3 Teacher self-efficacy

Perceived self-efficacy refers to the belief in one's capabilities to organize and execute the courses of action required to produce a given result (Bandura, 1997). Teacher self-efficacy refers to the same definition in the context of teaching. In other words, it refers to a teacher's belief in his/her capabilities to perform certain actions such that a given outcome is produced.

Teacher self-efficacy (TSE) has been a focus of educational research since the late 1970s when the Rand Corporation studied teachers' sense of personal efficacy (Armor, 1976), and Albert Bandura published his early works on self-efficacy (Bandura, 1977). Since the 1970s there have been conducted several studies on TSE beliefs, or teachers' self-referent judgements of capability. Using various measures and definitions, studies imply that "teachers with an assured sense of self-efficacy set the tone for a high-quality classroom environment by planning lessons that advance students abilities, making efforts to involve them in a meaningful way, and effectively managing student misbehavior" (Zee & Koomen, 2016, pp. 981-982).

Zee and Koomen (2016) presents in their literature review study an overview of research on teacher self-efficacy from the past 40 years. In addition to TSE affecting classroom quality, Zee and Koomen also points to studies that imply that TSE has also been found to exert influence over student and teacher outcomes, as well as showing some links to academic achievement and self-efficacy on the student side. Positive TSE beliefs have also been shown to improve psychological well-being of teachers in terms of job satisfaction, commitment, and lower levels of stress (p. 982).

The attribution-based theory of locus of control by Rotter (1966) and Banduras work on TSE (Bandura, 1977, 1986, 1997) have made a made an enormous impact on TSE research over the years (Zee & Koomen, 2016, p. 983). The emphasis in these theories is that humans are able to exercise control over actions that affect their lives. In attribution-based theory of locus of control, Rotter assumes that individuals differ in their perceptions of whether outcomes are contingent on external control (luck, fate, or others), or on internal control, more specific, a result of their own actions (Zee & Koomen, 2016, p. 983). Bandura argued that an individual's behavior is influenced not only by generalized expectancies for control, but also by the individual's perceived capabilities, their self-efficacy to perform certain actions that produces a certain outcome in a particularized domain (Zee & Koomen, 2016, p. 983).

Bandura made a distinction between outcome expectancies and efficacy expectations. Outcome expectancy is defined as an individual's "estimate that a given behavior will lead to certain outcomes" (Bandura, 1977, p. 193). Efficacy expectation is defined as "the conviction that one can successfully execute the behavior required to produce certain outcomes" (Bandura, 1977, p. 193). The difference between them is also visualized in figure 2-1. Bandura explains that they are differentiated because individuals can believe that certain actions will produce certain outcomes, but that such information does not influence their behavior if they have serious doubts about whether they can perform the necessary actions (Bandura, 1977, p. 193). Hence, their efficacy expectations, or self-efficacy expectations, will affect their behavior.

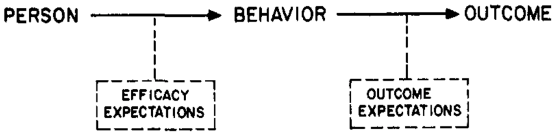


**Figure 2-1: Diagrammatic representation of the difference between efficacy expectations and outcome expectations (Bandura, 1977, p. 193)**

Most researchers have, since Banduras writings, underscored the differentiation between self-efficacy and outcome expectancies (Zee & Koomen, 2016, p. 984).

Measurement of TSE has also been a central part of TSE research. Gibson and Dembo (1984) extended the Rand measure (Armor, 1976), and their measure consisted of two factors, a measure of personal teaching efficacy (teachers competency beliefs), and another measure of general teaching efficacy (expectancy beliefs that their effectiveness is limited by environmental obstacles) (Klassen, Tze, Betts, & Gordon, 2011, p. 22). Gibson and Dembo's (1984) measure remained popular among researchers until the late 1990s, when it was criticized due issues with the validity of general teaching efficacy (Klassen et al., 2011, p. 22; Zee & Koomen, 2016, p. 984).

With Bandura's sociocognitive framing, researchers started conceptualizing TSE as task- or situation-specific rather than generalized, as Rotter does. They acknowledged that TSE beliefs could be dependent on different types of tasks, students, and circumstances in class. This acknowledgment lead to particularized self-efficacy scales that have been argued to have higher predictive validity, since they take variation in TSE across different domains into account, and therefore focus on the breadth of teachers' role in the classroom and not only on student outcomes (Zee & Koomen, 2016, p. 984). The often used Teachers' Sense of Efficacy scale (Tschannen-Moran & Hoy, 2001), which is also used in the interview study in this thesis, is an example of this. In this scale, TSE is treated as a task-specific, three-dimensional construct reflecting instructional practices, classroom management, and student engagement. Researchers using this instrument, have reported satisfactory reliability and construct validity evidence, across grades and several countries (Zee & Koomen, 2016, p. 984).

Other educational researchers have developed separate self-efficacy scales for literacy (Tschannen-Moran & Johnson, 2011), science (Riggs & Enochs, 1990), inclusive practices (Malinen et al., 2013), technology (Sang, Valcke, Van Braak, & Tondeur, 2010), and discipline (Brouwers, Evers, & Tomic, 2001)[1]. Notable is also the Norwegian Teacher Self-efficacy scale (Skaalvik & Skaalvik, 2007), where the teachers' psychological well-being has been connected to TSE.

[1] The sources in this paragraph have been found in the literature review study by Zee and Koomen (2016, p. 985)

## 2.4 Relevant research on teacher self-efficacy in the domain of programming

Measuring self-efficacy in relation to specific content and context (like teaching programming) is difficult, as the construct needs to be situated in a specific curricular context (Hartell et al., 2019). Bandura (2006) points out that "scales of perceived self-efficacy must be tailored to the particular domain of functioning that is the object of interest (pp. 307-308). In this subchapter I present research on self-efficacy in the domain of programming as a skill, and the domain of teaching of programming.

### 2.4.1 Teachers' self-effiacy in the domain of programming as a skill

Ramalingam and Wiedenbeck (1998) presents a computer programming self-efficacy scale based on the theory by Bandura (1977, 1896). The scale by Ramalingam and Wiedenbeck (1998) focuses on the self-efficacy of students in a C++ programming course, and they find that there was a growth in self-efficacy between two administrations of the scale twelve weeks apart. The scale is focused on programming as a skill, with four factors: Independence and persistence, complex programming tasks, self-regulation, and simple programming tasks. The scale has little focus on teaching of programming.

Korkmaz (2013) adapts Ramalingam and Wiedenbecks (1998) scale and conducts a study on prospective teachers' self-efficacy perceptions on computer programming. Programming skills are considered as one of the primary parts of both computer science, and computer teacher education. In the context of teacher education, it is important for prospective programming teachers to attain skills in programming (Korkmaz, 2013, p. 639). Korkmaz (2013) points to a study on mathematics teachers' computer self-efficacy (Askar & Umay, 2001) and argues that self-efficacy perceptions of prospective teachers is a significant variable in terms of teaching them computer programming skills (Korkmaz, 2013, p. 640). Korkmaz (2013) finds that prospective programming teachers' self-efficacy perceptions on programming are at a medium level. A similar study finds that preservice IT teachers have medium level programming self-efficacy perceptions, and that their complex programming skills improve when being trained with a Scratch programming tool (Yukselturk & Altiok, 2017). Another study also finds that students in computer teacher education have medium level of self-efficacy in programming, and that their self-efficacy can be improved with training in programming (Mazman & Altun, 2013).

### 2.4.2 Teacher self-effiacy in the domain of teaching programming

The studies presented in the last sub-chapter have mainly focused on programming as an important skill for teaching programming. Another aspect is teachers' self-efficacy in teaching programming, in other words, their belief in their capabilities to teach programming rather than only their perceptions of their programming skills. A recent Swedish study finds that primary school teachers in Sweden lack self-confidence in teaching programming, and that the teachers questioned "the why" behind teaching programming in Swedish primary school. On the other hand, the teachers did perceive programming as something relevant for pupils to engage with in their primary education (Hartell et al., 2019).

In a study of Swedish teachers' self-efficacy in digital competence (Mannila et al., 2018), the researchers found that teachers had least confidence with regard to competences related to programming and copyright/licenses, and that sixty percent of the 530 respondents felt insecure when it came to solving problems using programming (Mannila et al., 2018, p. 83). Mannila et al. (2018) finds a large spread in teachers' self-efficacy in digital competence, but that common to all teachers was a need for guidance in programming (p. 84). They conclude that their results "imply strongly that helping teachers develop their self-efficacy in digital competences is important, as studies show that teachers with a high self-efficacy in the subject they teach are more likely to persist longer, provide a better teaching environment and not burn out as easily" (Mannila et al., 2018, p. 84).

A similar study done in England when they introduced the new subject Computing in 2014, showed that 60% of the teachers felt not competent in teaching the new computing curriculum (YouGov, 2015). In another report three years later, in 2017, 48% of the teachers still reported low confidence due to lack of "sufficient theoretical and technical knowledge of computing that included aspects of programming and coding" (Royal Society, 2017, p. 55).

Yadav, Gretter, Hambrusch, and Sands (2016) interviewed twenty-four high school computer science teachers in USA to find out what challenges the teachers face in the classroom and what support systems they perceive would be helpful. The study found out that the teachers struggled with teaching computer science due to limited content knowledge. The teachers in the study also found it difficult meet to the pupils' needs on an individual level, for example because of the uniqueness of problem solving approaches each pupil might use. Another challenge was assessment, especially in terms of a lack of quality computer science assessment tools and teachers' lack of background or content knowledge in computer science. Yadav et al. (2016) also present three compounding factors that influence teaching computer science: Lack of CS teacher preparation (that teachers do not have the necessary training in computer science), isolation (that computer science teachers work in solitude without peers in one's content area), and IT challenges (unavailability of new and necessary technology and resources for teaching computer science). To address these challenges Yadav et al. (2016) also presents what the teachers identified as their needs. The teachers in the interviews specified a need for a more organized repository of teaching resources. The teachers expressed that while there is an abundance of teaching resources online, it can be challenging to assess their quality, and that there is a need for a better organization of online teaching resources and assessment tools. The other need presented in the study is the need for a community of practice in order to address the isolation of computer science teachers within their school. One of the main implications presented in the study is "the need for developing communities of practice for teachers to meet their curriculum needs (both content and pedagogical) and address the lonely enterprise of teaching computer science" (Yadav et al., 2016, p. 248).

Kadirhan, Gül, and Battal (2018) presents a study on Turkish K-12 computer science teachers perceived benefits and challenges in coding education, as well as their perceived self-efficacy skills to teach coding. The researchers find that the teachers perceived challenges are infrastructural issues related to computer resources and internet connection, lack of suitable resources for coding activities, lack of competent teachers of coding and a need for in-service training, overcrowded classes, and misperception of CS teachers' role in school. In regard to self-efficacy skills to teach coding Kadirhan et al. (2018) found that the teachers perceived content knowledge, personal characteristics and enjoyment in teaching, being able to motivate pupils, pedagogical knowledge, classroom management, and the ability to develop appropriate instructional material, as essential self-efficacy skills that teachers should possess for effective coding education. The study also showed that teachers perceived coding education as "important due to its affordance of improving student's computational thinking skills" (Kadirhan et al., 2018, p. 221).

### 2.4.3 Summary of research on teacher self-efficacy and attitudes towards teaching programming

This sub-chapter summarizes the relevant research and present what one can bring forward with regard the research done in this thesis. There exists self-efficacy in the domain of programming as a skill, but there seems to be lack of well-defined and tested scales of teacher self-efficacy in the domain of teaching programming. In regard to teachers' self-efficacy towards programming as a skill and competence, the research presented in this chapter show that prospective computer science and programming teachers have a medium level of self-efficacy in programming, but that their self-efficacy can be improved with training in programming (Korkmaz, 2013; Mazman & Altun, 2013; Yukselturk & Altiok, 2017). These studies do not say much about the prospective teachers' self-efficacy towards teaching programming. There is an indication that training in programming can increase prospective computer science and programming teachers' self-efficacy in the domain of programming skill.

There has been done some studies on teachers' attitudes and self-efficacy towards teaching programming as well. Two Swedish studies show that Swedish teachers in primary school have low confidence in teaching programming and using programming to solve problems, and that there is a need for guidance of the teachers in programming (Hartell et al., 2019; Mannila et al., 2018). One of these also show that the teachers in the study were positive towards programming in school, but also insecure towards why it was relevant (Hartell et al., 2019). Two studies in England, done with a three year gap in-between showed that nearly half of the teachers in the study did not feel confident in teaching the new computing curriculum, due to lack of competence in programming (Royal Society, 2017; YouGov, 2015). These studies indicate a need for raising prospective programming teachers' competence in teaching programming. This can also be seen in relation to the report on informatics education in Europe (Vahrenhold et al., 2017), and the recommendation that teaching of informatics, in this case programming, should be done by qualified teachers with competence in programming.

A study on US K-12 computer science teachers showed that the teachers found teaching computer science challenging due to limited content knowledge. The teachers also found adapted teaching and assessment in computer science challenging. The study found three factors of challenges in teaching computer science: Lack of competence, few peers to work with for the teachers in computer science, and lack of IT resources for teaching computer science. The study expresses a need for quality teaching resources and assessment tools in computer science, and communities of practice for computer science teachers (Yadav et al., 2016). A study on Turkish computer science teachers find that the teachers have infrastructural challenges like lack of suitable computer equipment and resources, lack of suitable teaching material for coding activities, and lack of competent teachers of coding. The study also showed that the teachers perceived coding education as relevant to have in school (Kadirhan et al., 2018). The teachers' perceived challenges in these studies are relevant to compare to the results in this study. There is also again an indication that the lack of competent teachers of programming is a challenge towards computer science and programming education in school.

# 3 Case: Continuing education in programming for teachers

This chapter describes the case used in this research. The case used in this study is a continuing education program in programming (NTNU, 2020d) at the Norwegian University of Science and Technology. The study program is aimed at in-service teachers in grades 8-13 that need competence in programming, and the study qualifies for teaching programming in grades 8-13 (and earlier).

The continuing education program is an online study, with web-based lectures and weekly activities such as online lectures and regular compulsory work exercises. The study program consists of two online courses. The first course is called Introductory Programming for Teachers and focuses on programming as a subject and skill, and giving insight into to how programming can be used to create digital solutions (NTNU, 2020a). This course is done in the fall semester, and will be referred to as the *fall course* or *introductory programming course* from now on. The learning outcomes of the introductory programming course can be found in table 3-1.

**Table 3-1: Learning outcomes: Introductory Programming for Teachers (NTNU, 2020a, translated from Norwegian to English by the researcher)**

| | |
|---|---|
| Knowledge | • have basic knowledge of constructions and structures in modern programming<br>• have basic insight in techniques and methods for testing and debugging in short, basic programs |
| Skills | • develop and run basic programs<br>• install and use a programming environment<br>• be able to use and explain basic programming examples in teaching<br>• understand how creativity and cooperation can be utilized to promote learning of basic programming |
| Generic competence | • convey basic programming and the didactic of the subject to others, both written and oral<br>• discuss, describe and evaluate basic solutions<br>• plan varied exercises with basic programs<br>• demonstrate basic digital competence |

The second course is called Applied Programming for Teachers and focuses on more complex programs and applied programming in areas like games, control of simple electronics and robots (like Arduino and Micro:bit), programming used in computing and simulations in subjects like math and physics (NTNU, 2020b). The course has as a goal to "give students a deeper understanding of basic programming and how this can be applied to solve issues within different subject areas" (Olstad & Rouhani, 2019, p. 603). Furthermore, the course focuses on how in-service teachers can ease pupils' learning processes and understanding of programming, as well as give guidance on how to use programming to support learning in school (Olstad & Rouhani, 2019). This course is done in the spring semester, and will be referred to as the *spring course* or *applied programming course*. This course is also a web-based course and focus on a learner-centered structure[2] (Rouhani et al., 2019). The learning outcomes of the applied programming course can be found in table 3-2.

**Table 3-2: Learning outcomes: applied programming for teachers (Olstad & Rouhani, 2019, pp. 603-604)**

| | |
|---|---|
| Knowledge | • detailed knowledge of constructions and structures in modern programming<br>• knowledge of programming languages, tools and methodology, both pedagogically oriented solutions and solutions that are used professionally<br>• basic understanding of the software's function in electronics and robots<br>• basic insight into the technique and methods for testing and misfire in major program projects |
| Skills | • develop and test programs with some complexity<br>• utilize modern programming tools and assess their suitability in teaching and learning<br>• identify and evaluate programming that should be included in simple technological solutions, understand the difficulty, scope and suitability in a teaching situation<br>• understand how creativity and collaboration can be utilized to promote programming learning |
| Generic competence | • convey knowledge of programming and the didactics of the subject to others, both written and oral<br>• discuss, describe and evaluate solutions with some complexity<br>• plan varied work tasks and programming projects, focusing on creativity and social learning<br>• evaluate ethical issues related to programming<br>• demonstrate good digital competence |

The students in the 2018-19 continuing education program are in-service teachers in primary, secondary, and high school. Their teaching subjects vary, but the majority are secondary and high school teachers in STEM subjects. The same applies to the students in the 2019-20 continuing education program in programming, with the majority being high school teachers in STEM subjects.

As a part of the exam delivery in the applied programming course, students in the course delivered a reflection note on their thoughts and reflections towards their own learning and results. 80 students started in the applied programming course in 2019, and 65 anonymized reflection notes were made available for the researcher in January 2020. These reflection notes have been analyzed as a part of this thesis. The study on the reflection notes is presented in chapter 4.

[2] To read more about learner-centered design, see Guzdial (2015) and Rouhani, Divitini, Vujosevic, Stai, and Olstad (2019)

10 students from the 2018-19 continuing education program in programming have been interviewed as a part of the work in this thesis. This is the main interview study, and is presented in chapter 5.

3 students from the 2019-20 continuing education program in programming have also been interviewed as a part of the work in this thesis. These students were still participating in the course when the interviews were conducted. This supplementary interview study is presented in chapter 6.

# 4 Study of reflection notes

## 4.1 Chapter overview

This chapter presents the study of reflection notes from the applied programming course. The reflection notes are part of the final delivery and assessment in the course and are written by in-service teachers that were students in the continuing education program in programming in 2018-19 (see chapter 3 to read more about the education program).

This chapter first presents the data, then the process and method used in the study on the reflection notes. Then the results are presented, and lastly the results are summarized and put into the context of self-efficacy.

This chapter explores the research questions:

**RQ1:** What attitudes do in-service teachers with programming education have towards programming in school?

**RQ2:** How do in-service teachers with programming education perceive their self-efficacy in teaching programming?

**RQ3:** How do in-service teachers perceive that programming education has affected their self-efficacy in teaching programming?

The results indicate that the teachers are positive towards using and teaching programming in school, and that programming can be relevant to use interdisciplinary. There are some worries on how programming is included in the curriculum. There is an indication that the teachers perceive a need for competent teachers of programming, and they want colleagues with programming competence to work and collaborate with.

The results indicate that the teachers have a relatively high sense of self-efficacy in teaching programming. The teachers seem to perceive adapted teaching and assessment as challenging in programming. The teachers also perceive technical issues as a challenge in teaching programming. It is indicated that the teachers feel they can motivate their pupils, as well as developing and adapting suitable teaching material. Some teachers perceive how to teach programming and how to teach programming as challenging. Many teachers perceive their own programming skill as relatively low, but this does not seem to have a significant impact on their self-efficacy in teaching programming.

The results indicate that the programming education has had a positive impact on both the teacher attitudes and self-efficacy towards programming and teaching programming. Some teachers thought the learning curve in the introductory programming course was steep.

## 4.2 Data: Reflection notes

The reflection notes are part of the final delivery and assessment in the applied programming course. In the delivery/project, students design and develop a lesson plan in the domain of applied programming which will be used in their teaching. In other words, they design and implement one or several teaching lessons for their pupils where they focus on programming and applied programming.

After finishing the project and, if possible, implementing the project in school, the students/teachers also had to hand in a reflection note on their thoughts and reflections towards the result of the project and their own learning in the course.

The reflection notes have no common structure and are in written form in Norwegian. They vary in length from 44 words to 1864, with an average word count of 625. This posed a challenge on how to analyze the reflection notes, as there was no quick or simple way to extract the information. The requirement given to the students regarding the content of their reflection notes was that it should contain reflections towards the results of the project/delivery and their own learning in the course. This has resulted in very varied content and little consistency across the reflection notes. This is also the reason behind the method used in the analysis.

Of 80 students enrolled in the applied programming course in 2019, 73 received a grade after delivering the final project. 65 anonymized reflection notes have been made available for the researcher and been analyzed.

32 of the 65 teachers have implemented their lesson plans or parts of their lesson plan in class. 27 have not implemented it, and 6 do not report if they have implemented it or not.

## 4.3 Method

### 4.3.1 Collecting the data

The reflection notes were made available for the researcher by the lecturer in the applied programming course. Before they were made available for the researcher, the reflection notes were anonymized due to privacy issues. The reflection notes do not specify the teachers' respective grade and teaching subjects. This could be deduced by some of the reflection notes, but in such a small amount that this was not taken into account in the analysis.

### 4.3.2 Analysis of reflection notes

As mentioned, the reflection notes have little consistency in terms of content, and also vary in length. This study focuses on the participants views, and instead of trying to "measure" the teachers' self-efficacy with numbers, the study looks into the participants own views and perspectives from a qualitative perspective. To be able to make the data manageable, a method for coding and structuring the data was chosen, inspired by grounded theory (Strauss & Corbin, 1998).

The reflection notes were first read through to get a feeling and some ideas on how to deal with the data. Then it was used open coding where phenomena were named and categorized through careful and intense review of the data. This is done to be able to develop precise concepts that can be linked to various elements of the data material (Postholm, 2005). NVivo 12 (QSR International, 2018) has been used in the coding process. There are several ways of selecting what to code during open coding, in this study the researcher has coded whole sentences and sections to be able to extract the content of these. When new codes appeared, the researcher went back to previously coded reflection notes to see if the new code also appeared in these. After the process of open coding, the codes were categorized. Even though the categorization was done after the process of open coding, the researcher saw some patterns and potential categories during the open coding. Categorization is when codes that seemingly cover the same phenomenon are gathered in categories (Postholm, 2005). The categories that emerged from the analysis can be seen below in table 4-1:

**Table 4-1: Categories after open coding and categorization**

| Category | Description |
|---|---|
| Attitudes towards programming in school | The students' general reflections and experiences towards programming in school |
| Teaching programming | Reflections regarding teaching of programming |
| Programming skill | Reflections regarding the students' perceived programming skills |
| External challenges | Challenges that are not directly connected to the student's teaching competence |
| Course specific reflections | Reflections about the course in general |

After the open coding and categorization the researcher connected categories to their sub-categories, the process called axial coding (Postholm, 2005). The goal in axial coding is to specify a category by the different circumstances that create them. To specify these sub-categories the researcher can ask himself questions about when, why, and under what circumstances a specific category emerged (Postholm, 2005). These questions reveals relations between categories and their subcategories (Strauss & Corbin, 1998). In the axial coding the researcher went through the data material once again with a focus on the five categories and coded the material into each category while also creating the sub-categories. When a new sub-category emerged, the researcher went back to previously coded reflection notes to see if the new sub-category was also present in previously coded reflection notes. The analysis was first done in the researcher's native language, Norwegian, to be able to describe the data most precise, and then translated to English. The categories with their sub-categories can be seen below in table 4-2:

**Table 4-2: Categories and their sub-categories after axial coding**

| Category | Sub-category |
|---|---|
| Attitudes towards programming in school | Positive towards programming in school |
| | Positive towards using programming interdisciplinary |
| | Community and collaboration |
| | Teaching programming requires competent teachers |
| | Looks forward to teaching programming |
| | Motivated to learn more |
| | Worried on how programming is to be included in school |
| Teaching programming | Positive towards own teaching |
| | Adapted education |
| | Positive towards own project |
| | Motivation |
| | Can improve teaching |
| | Learning resources |

| | |
|---|---|
| | Importance of experience |
| | Didactical issues |
| | Assessment |
| Programming skill | Positive towards own learning |
| | Perceives own programming skill as relatively low |
| | Believes some pupils have better programming skill than themselves |
| | Has learned basic programming |
| | Able to increase programming skill |
| | Maintaining programming skill |
| External challenges | Time is an issue |
| | Technical issues |
| | School resources |
| | Math in programming |
| | Vast amount of internet resources |
| | Pupils digital competence |
| Course specific reflections | Positive feedback |
| | Negative feedback |

Sub-categories, sample sentences/sections and frequency of coded elements into sub-categories are found below in table 4-3:

**Table 4-3: Subcategories, frequency of codes, and sample sentences**

| **Sub-category** | ***Freq.*** | **Sample sentence** |
|---|---|---|
| *Attitudes towards programming in school* | | |
| Positive towards programming in school | 27 | My overall reflection is that it will be very interesting to develop the subjects in Norwegian school now that programming will gradually gain more space and significance. It is very relevant to use programming as a tool in high school. (Teacher 22) |
| Positive towards using programming interdisciplinary | 19 | But I clearly see how much potential programming can have in understanding of math and hope future curricula can enable me to apply this lesson plan. (Teacher 16) |
| Community and collaboration | 7 | It is a challenge to learn yourself programming and stay up to date. Therefore, it is good to be at a school with several teachers interested in programming which you can exchange experiences and lesson plans with. (Teacher 43) |
| Teaching programming requires competent teachers | 7 | We see that this motivates the pupils very much. But we also see that it requires a lot of knowledge and expertise from the teachers, and that it is important to be well prepared. (Teacher 14) |
| Looks forward to teaching programming | 6 | I look forward to trying out and develop the lesson plan further. (Teacher 47) |

|  |  |  |
|---|---|---|
| Motivated to learn more | 6 | I see that I have become more and more interested in programming through this study, and look forward to many hours of programming in the future. (Teacher 23) |
| Worried on how programming is to be included in school | 2 | There is a danger that the finished code will become 'black boxes' which does something one does not quite what is. But that works. I am therefore worried about the implications of new curricula for pupils and teachers without programming skills in the first place. (Teacher 4) |
| *Teaching programming* |  |  |
| Positive towards own teaching | 46 | I am pleased with the result. The aim of the project was to create a program that could be adapted to the individual pupil's interest and needs, in addition to being fun … The parts of the program I have tested worked very well, the pupils were positive and engaged. (Teacher 33) |
| Adapted education | 43 | Especially when you have in mind that the pupils have no prior knowledge and that it is new to the teacher. I spent a good deal of hours thinking about how this can be done in a way that will meet all the pupils, without seeming overwhelming. (Teacher 6) |
| Positive towards own project (lesson plan) | 29 | The end result of this lesson plan has become very good, due to of the fact that this should be something that sparks interest in programming. Pupils get to do something practical in their subject which they can also use later, while at the same time pupils get an understanding of how programming can be done. (Teacher 6) |
| Motivation | 14 | I wanted them to see a result, which created some mastery and motivation to continue with Arduino programming. (Teacher 7) |
| Can improve teaching | 14 | In retrospect, I think the lesson plan should have been split in two. Where the main part becomes a separate lesson, but a lesson that to a certain extent requires pupils to know the basic concepts of programming. (Teacher 15) |
| Learning resources | 13 | It is no problem to find pre-written code and good ideas for Arduino projects, but the point here was to teach the pupils some simple coding and not to show everything one can use Arduino for (Teacher 62) |
| Importance of experience | 8 | I certainly don’t have the full overview even though I have become quite good at programming in Python. I expect to learn a lot myself and not least about the pupils' learning during the first year as a programming teacher. One can probably argue with some truth that it is when teaching the subject that real learning begins. (Teacher 26) |
| Didactical issues | 7 | In a subject like programming, I think the didactic is challenging. How should we organize the teaching so that the pupils learn as best as possible, and what is important to focus on? (Teacher 4) |
| Assessment | 6 | I think it is difficult to set a mark on the pupils' work - when have they learned enough of what they are supposed learn? What exactly is enough and how can we measure |

| | | |
|---|---|---|
| | | that they have learned it? In retrospect, we see that we were not clear enough on what competence we wanted the pupils to remain with after completing this project. We had set many sub-goals along the way, but we could be even more concrete both to help pupils understand what they should be working towards and to help teachers with the assessment of pupils. Assessment in programming something I want to work on in the future. (Teacher 50) |
| *Programming skill* | | |
| Positive towards own learning | 22 | My result after completing IT6203 is increased competence in programming. Although the course this fall (IT6203) was a basic course, it was brand new to me and the learning curve was steep. Previously, I had some experience with block-based programming at the pupil level, but no experience with text-based programming. Now that I have acquired some expertise in basic programming, I am able to recognize features across block-based and text-based languages. I also recognize some features from other languages, even though I haven't worked actively with them. This makes me more confident in myself and my programming skills, at least for school use. (Teacher 48) |
| Perceives own programming skill as relatively low | 17 | The main reason why it took a lot of time before I starting to make the lesson plan is probably my own feeling of security in programming. Since I do not feel confident in programming, it became challenging to see what and how to teach. (Teacher 6) |
| Believes some pupils have better programming skill than themselves | 8 | Many pupils know a lot and learn very quickly and often pass the teacher. It will be a major challenge for teachers to keep pace in the future. A teacher must almost be constantly undergoing further education. I think many school leaders and politicians don't know how demanding this can be. (Teacher 43) |
| Has learned basic programming | 7 | The education has given me the basic knowledge I needed to become curious of programming, and on the side of the education I have now started various projects using programming in the context of teaching. (Teacher 31) |
| Able to increase programming skill | 6 | There was a moment when I was unsure if I would finish at all. It was probably in the borderlands for my competence, but at the same time I learned a lot and gained the knowledge about using python for graphical scripting that I was looking for. (Teacher 37) |
| Maintaining programming skill | 5 | However, I have had to spend some time rehearsing and reading up on some basic methods used on lists, matrices and in plotting. This has been a small eye opener for me. Something that was really easy in the fall I had to spend some time getting back in. Coding is obviously not quite like cycling - you forget if you don't use it regularly. (Teacher 8) |
| *External challenges* | | |

|  |  |  |
|---|---|---|
| Time is an issue | 18 | Time consumption became quite extensive, and I chose to run only the first two tasks in this project. (Teacher 23) |
| Technical issues | 13 | After everything was mounted, we connected the first circuits. There are several things that can be connected incorrectly, and there are few error messages if the error lies in how the components are connected. This quickly became a challenge, and I ran around to the pupils to find out where the error was. (Teacher 32) |
| School resources | 5 | We have previously worked a lot individually and with Scratch or similar because we did not have enough equipment at the school to do anything else. (Teacher 9) |
| Math in programming | 4 | The challenge in some of the programs is to understand the programming and especially the mathematics, which is behind some of the conversions (Teacher 7) |
| Vast amount of internet resources | 3 | During this exam project, I have once again realized that there is an ocean of good exercises on various online sources. All these resources are great as a starting point, but the problem is almost that there is too much to choose from. (Teacher 17) |
| Pupils digital competence | 2 | At the same time, I know that some pupils in secondary school often find it difficult to find information themselves. This, I think, will be one of the biggest challenges (and therefore also for the learning) for the pupils. (Teacher 27) |
| *Course specific reflections* |  |  |
| Positive feedback | 8 | It has been very positive to be able to work at your own pace, with quick feedback in Slack, Webinar and on assignments along the way. (Teacher 21) |
| Negative feedback | 2 | The pre-project had submission too late. The time from which the pre-project was approved to<br>the project was supposed to be delivered was too short. For example, I do not manage to implement<br>the lesson plan in class before the submission deadline. (Teacher 11) |

A tree-map of categories and their sub-categories can be seen below in figure 4-1. The tree-map shows the most frequently coded sub-categories after the axial coding. The bigger the square, the greater number of sentences or sections are coded into the specific sub-category. Different colors represent the different categories.

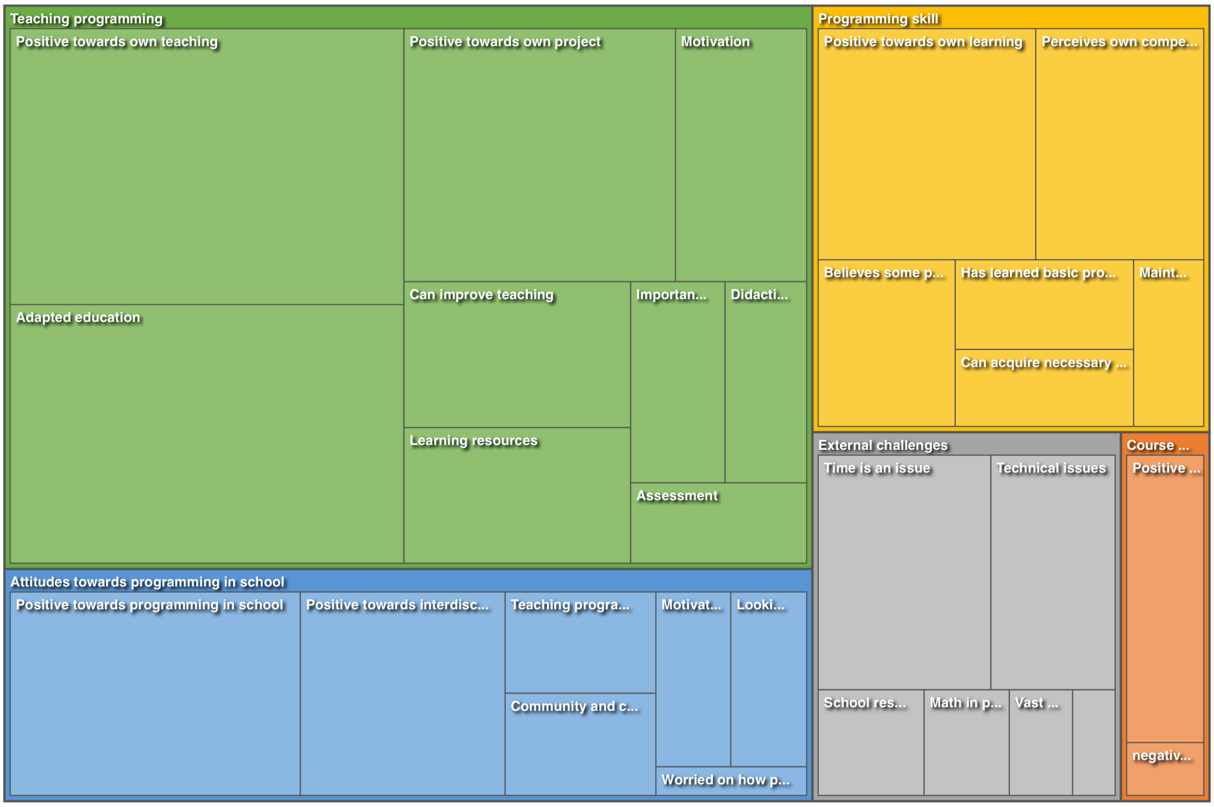


**Figure 4-1: Tree-map of categories and sub-categories, nodes compared by number of coding references. Generated in NVivo 12**

## 4.4 Result

This sub-chapter presents the results from the analysis if the reflection notes. All quotes are indented for readability. Quotes have been translated from Norwegian to English by the researcher.

### 4.4.1 Attitudes towards programming in school

The reflection notes indicate that teachers in the course are positive towards the inclusion and use of programming in school. Many of the teachers find programming as something useful for the pupils to learn and use in school. For instance, teachers expressed that:

> Programming is engaging, it creates commitment and collaboration. In educational settings we experience that pupils talk a lot to each other, and that there is a positive "work-noise" in the classroom. (Teacher 46)
>
> This is one of the things I find exciting about the inclusion of programming in school, that it creates space for the pupils to be creative and that they can see how there can be several trajectories to a good and correct solution. (Teacher 42)
>
> My eyes have been opened for the educational possibilities that unfold with the inclusion of programming in school. (Teacher 30)
>
> This exercise offered good theoretical insight, and to me, a personal persuasion for why programming will be an important subject in school in the years to come. (Teacher 63)

As we see here, teacher 30 and teacher 63 also indicate that the programming education has had a positive impact on their attitude towards programming in school. Other teachers also reported that the programming education has impacted their attitude towards programming in a positive way:

> After this study, I have also become far more observant of the importance of programming in school, and believe it can help make the school life for many pupils easier and more interesting. (Teacher 60)

Some of the teachers are also positive towards the use of microcontrollers and physical devices with programming in school:

> At the same time, I as an electro physic teacher think that using Arduino in teaching about different sensors and components is a fantastic tool. (Teacher 7)
>
> Arduino has met my expectations. It provides a good way to use programming on physical components. I think this can give pupils more insight into how one can use programming in different places. (Teacher 32)

Many of the teachers are also positive towards using programming interdisciplinary and find it relevant to use in their own teaching subjects. For example, some of the teachers express:

> The work on equation solving has been particularly interesting. This goes straight into the core of what programming in mathematics should be and provides exciting experiences towards new curricula in Fagfornyelsen[3]. (Teacher 25)
>
> The competence I have acquired will also be useful in my other teaching subjects, mathematics and natural science. Through working on the project assignment, I have not become so much better at programming. But I have become more aware of how I can communicate what I have learned to my pupils. (Teacher 26)
>
> As I read through the curriculum learning objectives with programming in mind, I saw that it is relevant in many subjects (as shown in the pre-project) and that there are very many of the competence goals that argue for using programming in school. (Teacher 27)

Some of the teachers express that they find it very useful to collaborate with colleagues when dealing with programming:

> In particular, it has been valuable to carry out the project with a fellow student and colleague. (Teacher 25)
>
> It is a challenge to learn yourself new programming and stay up to date. Therefore, it is nice to have colleagues that are interested in programming which you can exchange experiences and lesson plans with. (Teacher 43)

Others also state that it will be important to focus on increasing programming competence in school staff and want more colleagues with programming competence, for example:

> This also gives us some thoughts on that it is important to raise competence in the staff groups if we are to succeed in all pupils learning programming, regardless of grade and what teacher they have. It requires a lot from the educators in regard to aquiring knowledge in the subject. It will be important to set up such learning arenas for staff in the future. (Teacher 46)
>
> In addition, our school will focus more on programming so that there will be a new teacher at the school who will follow the same studies I have taken. Thus, I get a colleague I can spare with when we are developing school projects. What I miss is that language and social science teachers are also expanding their programming skills. Then it will be easier to make projects more interdisciplinary if we can expand the project with more competence goals, and where the correct subject teachers can assess the achievement of goals in these competence goals. (Teacher 55)

As seen, some of the teachers think it will be important that more teachers increase their competence in programming, in relation to this some teachers express that teaching

[3] «Fagfornyelsen» is the name of the new curriculum in the Norwegian school system

programming and succeeding in including programming in school requires competent teachers. For example, one of the teachers expresses that:

> We see that this motivates the pupils very much. But we also see that it requires a lot of knowledge and expertise from the educators, and we see that it is important to be well prepared. (Teacher 14)

After finishing the project by creating the lesson plan, and by this also finishing the continued education in programming, some of the teachers express that they are looking forward to trying out the lesson plan in their class:

> I look forward to trying out and developing the lesson plan further. (Teacher 47)

> It is a pity that I have not had the opportunity to try out the lesson plan with the pupils yet, but it will be exciting when implement it this fall. (Teacher 29)

Another observation is that some of the teachers report that they have gained interest in programming, and that they are motivated to learn more, this is indicated through statements like:

> As a teacher, I am ready and want to learn more. I am much more secure, clearer and more inspired to use this now than I was before the programming study. (Teacher 7)

> I have become more and more interested in programming through this study, and look forward to many hours of programming in the future. (Teacher 23)

We see that most of the teachers are very positive towards programming in school. A few of the teachers is however worried about how programming is included into the curriculum, for example in regard to teachers that have little or no programming competence:

> There is a danger that the finished code will become 'black boxes' which does something one does not quite what is. But that works. I am therefore worried about the implications of new curricula for pupils and teachers without programming skills in the first place. (Teacher 4)

## 4.4.2 Teaching programming

### 4.4.2.1 Feelings towards own teaching of programming

The reflection notes indicate that the programming education in general have impacted teachers' feelings towards teaching programming in a positive way. Many report that they feel more secure in their teaching of programming:

> I am much more secure, clearer and more inspired to use this now than I was before the programming study. (Teacher 7)

> After a hard work, challenging and educational half year with IT6204 studies, I now feel a little better equipped to enter the IT classroom. I know that I have a greater sense of security in the choices that should be made regarding getting pupils motivated and engaged. And I know there are challenges in getting girls to start with, and to stay in the subject. (Teacher 17)

> My programming skills in the fall of 2018 were near zero. Through the studies in Basic Programming and Applied Programming I believe that I have been given a good enough foundation to be able to teach elective programming. (Teacher 26)

Most of the teachers who implemented their lesson plan are also positive towards their own teaching, results and experience. This is indicated through statements like:

> I am pleased with the result. The aim of the project was to create a program that could be adapted to the individual pupil's interest and needs, in addition to being fun … The parts of

> the lesson plan I have tested worked very well, the pupils were positive and engaged. (Teacher 33)
>
> Our experience with this method of introducing programming for pupils has been positive. We see that the pupils got motivated and interested in trying out the different program codes we introduced. (Teacher 34)
>
> By the end of the lesson, the pupils were able to program two movements. They asked when they could continue with this, and we will try to carry out this exercise, even though my deadline is soon over. In other words, looks like the lesson plan is a success so far. (Teacher 44)

Also, many of those who did not implement their lesson plan are very positive towards the lesson plan they created, and believe it will work well in their teaching:

> I believe the two lessons can be implemented in the classroom with great success, but the two projects also have very different degrees of difficulty. (Teacher 2)
>
> The end result of this lesson plan has become very good, due to of the fact that this should be something that sparks interest in programming. Pupils get to do something practical in their subject which they can also use later, while at the same time pupils get an understanding of how programming can be done. (Teacher 6)
>
> When I look back at the lesson plan I have made, I am happy with the results and think it will work well in my class. (Teacher 42)

Thirteen of the teachers also explicitly state how they can improve their lesson plans further, to benefit the pupils and themselves even more. For example, one of the teachers explain that:

> In the next implementation of the lesson plan I will do the last part of the plan, the programming of the BitBot, a little different … by further developing the plan in this way, a more natural level differentiation of the plan is created, and the students can work at their own pace without running out of work for a longer period than during the first implementation. (Teacher 53)

Several of the teachers also express that they will benefit and learn more from getting more experience in teaching programming. This is indicated in statements like:

> I expect to learn a lot myself and not least about the pupils' learning during the first year as a programming teacher. One can probably argue with some truth that it is when teaching the subject that real learning begins. (Teacher 26)
>
> I believe time is a key word here and that I (as with everyone else) will continuously discover new opportunities while working with Lego Mindstorms in the classroom. Then, my knowledge and "resource repository" will increase with completed teaching hours and I will gain better and broader expertise over the years. (Teacher 39)

#### 4.4.2.2 Adapting teaching to the pupils

The aspect of adapting teaching to the pupils is something several of the teachers bring up. The reflection notes indicate that most of the teachers focus on adapted education when dealing with teaching of programming. Many see pupils’ prior knowledge in programming as a challenge, and that it is important to take into account. For example:

> But before implementing this lesson plan, it is important to know something about the pupils' programming skills. (Teacher 2)
>
> As with almost every other teaching, I think that adapting the teaching to each student's level will be the biggest challenge. Many students will need some time to understand the computational thinking, find it overwhelming and maybe feel I am moving too fast. (Teacher 26)

Even though many of the teachers see adapted teaching as a challenge, some also express that they have ways of addressing this challenge with different methods, for example by letting the pupils work together in specific groups:

> During the implementation I have learned that it may be a good idea to put them together in groups on the first programming tasks to assure the pupils who are extra insecure and then challenge them to create their own codes. (Teacher 9)

Or bringing extra challenges and exercises for the more capable pupils:

> Many will need a lot of follow-up and help, while other pupils will also be able to get it done quickly, and then you have to bring some extra challenges to give to these pupils. (Teacher 13)

Some of the teachers that implemented their lesson plan also experienced that adapting their teaching was not that problematic, and that they succeeded in adapting and differentiating to the pupils' needs. For example:

> The main impression I am left with after finishing the project period is that the task was comprehensive and demanding, but doable, fun and very educational. The lesson plan differentiated well and gave all pupils, regardless of their level of competence, challenges that suited them (Teacher 4)

> In addition, the group I teach is very varied, but it still works well in programming because they adjust the level of difficulty themselves based on how they want to solve a given task. The more expertise-the more advanced codes and solutions. (Teacher 9)

Another teacher expresses that it is easy to differentiate programming exercises, but that it can be more difficult to create good exercises for the pupils that are not particularly interested:

> The most important thing I learned was how easily one can differentiate programming tasks. It is very easy to add more challenging tasks to pupils who have a solid background in programming and who are interested in the subject. The hardest part is probably to make simple enough tasks, which in addition gives a sense of mastery, to those who do not find this particularly interesting. (Teacher 16)

#### 4.4.2.3 Assessment and didactical issues

A few of the teachers also talk about assessment in their reflection notes, and that this is a more challenging aspect in teaching programming. One of the teachers that included summative assessment as a part of the lesson plan had difficulties with landing on a grade that seemed fair:

> Even though the assessment criteria were defined beforehand, where oral presentation was an important part of the assessment, it turned out to be very difficult deciding a grade I experienced as fair. (Student 19)

Others reflect upon summative assessment as a challenge more in general terms:

> I think it is difficult to set a mark on the pupils' work - when have they learned enough of what they are supposed learn? What exactly is enough and how can we measure that they have learned it? In retrospect, we see that we were not clear enough on what competence we wanted the pupils to remain with after completing this project. We had set many sub-goals along the way, but we could be even more concrete both to help pupils understand what they should be working towards and to help teachers with the assessment of pupils. Assessment in programming something I want to work on in the future. (Teacher 50)

> In a subject like programming, I think the didactic is challenging. How should we organize the teaching so that the pupils learn as best as possible, and what is important to focus on? (Teacher 4)

As teacher 4 expresses here, that the didactic in teaching programming is challenging is also shared by a few other teachers. For example, one teacher explains that:

> The didactic approach to the subject material has been more challenging - how much should we explain to pupils in advance and in plenary? How detailed should the explanations possibly be? What kind of help is useful for student learning? (Teacher 50)

Of those of the teachers that reflect upon assessment in their teaching of programming, most of them find the summative assessment as challenging.

#### 4.4.2.4 Motivation

In the reflection notes, several teachers point out motivation as an important factor in teaching programming. For example, one of the teachers believe that it will be important to make programming seem relevant for the pupils in order for them to get motivated:

> If we can find problems that pupils can relate to and they can see the relevance to their own and others' everyday lives, I hope and believe that we can get them engaged and activated. (Teacher 29)

Others experienced that they managed to motivate their pupils in their teaching. One of the teachers verbalize that:

> We see that the students got motivated and interested in trying out the different program codes we introduced. We believe the mastery and the will to further develop the coding to get the robots to perform more advanced actions is connected to the fact that the pupils received this step by step introduction. (Teacher 18)

Of the teachers that reflect upon pupils' motivation in programming seem to be positive towards how they can motivate their pupils. For example one of the teachers believes that using micro:bit can help motivate the pupils:

> By using micro: bit, we open up for students to have something physical to relate to along the way. This can help motivate students at the same time as it opens up to using the theory in practice. (Teacher 1)

#### 4.4.2.5 Learning resources and material development

As seen already many of the teachers are positive towards their own lesson plan. Many have based their lesson plan on existing learning resources found online. Some express that there are very many good resources online, and that they find this useful in the development of learning material. For example, one teachers state:

> The part of the course that I think I will bring with me is simply the collection of online resources in the syllabus part of the course. There were a lot of very specific good tips there that I have to work with after completing the course. (Teacher 63)

At the same time as the teachers think there are a lot of good online resources, some also express that they think it might be too much to choose from:

> During this exam project, I have once again realized that there is an ocean of good exercises on various online sources. All these resources are great as a starting point, but the problem is almost that there is too much to choose from. (Teacher 17)

### 4.4.3 Programming skill

In terms of the teachers own programming skills, most of the teachers were happy with their own learning of programming and increase in skills through the programming education. This is indicated through statements like:

> Before this study I had not worked so much with programming, neither text programming such as Python, nor programming of microcontrollers such as Arduino, MicroBit or

> Raspberry Pi. Through the IT6203 and IT6204 programming courses, I have gained a good insight into programming structure and basic knowledge of programming languages, tools and methodology. (Teacher 34)

> My result after completing IT6203 is increased competence in programming. Although the course this fall (IT6203) was a basic course, it was brand new to me and the learning curve was steep. Previously, I had some experience with block-based programming at the pupil level, but no experience with text-based programming. Now that I have acquired some expertise in basic programming, I am able to recognize features across block-based and text-based languages. I also recognize some features from other languages, even though I haven't worked actively with them. This makes me more confident in myself and my programming skills, at least for school use. (Teacher 48)

Even though most of the teachers report that they have learned a lot throughout the continued education, many of them also express that they feel their programming skill or programming competence is relatively low. For example:

> The main reason why it took a lot of time before I starting to make the lesson plan is probably my own feeling of security in programming. Since I do not feel confident in programming, it became challenging to see what and how to teach. (Teacher 6)

> As of today, I master a rather limited repertoire of programming methods and platforms. I certainly do not have the full overview even though I have become quite good at programming in Python. (Teacher 26)

Another aspect in relation to this is that quite a few teachers believe that some pupils might have better programming skills than themselves. One of these teachers states that:

> Many pupils know a lot and learn very quickly and often pass the teacher. It will be a major challenge for teachers to keep pace in the future. A teacher must almost be constantly undergoing further education. I think many school leaders and politicians don't know how demanding this can be. (Teacher 43)

> There will still be pupils who know more than me, but I now have more opportunity to challenge them further in the subject, as I myself have more expertise in the products we work with. (Teacher 49)

On the other hand, none of these teachers express that they think this is a major challenge in their teaching. Contrary, some of them think this can also be a strength in some ways. For example:

> This caused that I, as a teacher, was given a different role in the classroom than I am used to. I became a "co-walker" in the learning process for pupils rather than an "oracle" that has the facts about everything. This also caused my relationship with the pupils to change in a positive way, I felt that we were becoming more equal. (Teacher 50)

Others also see the possibility for using these capable pupils as a resource in the classroom:

> Therefore, it is important that one can look at this as a learning arena between pupils and not just between pupil-teacher. In this way, the pupils who already have programming knowledge and interests in the subject can positively influence the rest of the teaching group. (Teacher 6)

Some of the teachers have managed to acquire new programming skills outside the curriculum of the programming education on their own. One of the teachers verbalize that:

> Having basic skills in a subject makes it much easier to further develop your skill on your own. (Teacher 48)

A few others point out that they had to “re-fresh” their programming knowledge when working with the project, and that the programming skills they learned the previous semester had maybe faded somewhat. One of the teachers stated it this way:

> However, I have had to spend some time rehearsing and reading up on some basic methods used on lists, matrices and in plotting. This has been a small eye-opener for me. Something that was really easy in the fall I had to spend some time getting back into again. Coding is obviously not quite like cycling - you forget if you don't use it regularly. (Teacher 8)

### 4.4.4 External challenges

Many of the teachers also reported some challenges not directly connected to their own competence in teaching or programming. The most stated challenge was the well-known factor of time. Many teachers reported that they did not have the time to implement their project in class, but others also found it difficult to find time in general for programming into the current curriculum. For example, one of the teachers find it difficult to find room for programming in math:

> One reason why this was difficult is that I consider this lesson plan not to be feasible in my classes at the moment. I only teach math at high school this year and in these subjects I can't defend spending a bunch of hours on basic programming. (Teacher 16)

Some others agree that there was not much room for programming in the old curriculum, and hope there will be more room for this in the new curriculum:

> As it is today, the amount of learning goals we will go through during the year does not allow for programming as a supplementary activity. I hope that the new curriculum will be less comprehensive, so that one can spend some time on activities with coding. (Teacher 61)

The second most stated external challenge was technical difficulties. Many teachers experienced that the microcontrollers, like micro:bit and Arduino, did not always behave as they expected and caused difficulties in their teaching. For example:

> Technical problems are always a challenge. We quickly found that neither the micro:bit nor the editors behaved the way they should. There was quite a lot of frustration with code that worked one day but not the next, signals that did not transmit, measurements that did not start and micro:bits that did not respond. (Teacher 4)

The third most stated external challenge was the school resources, in this case directly connected to the schools programming equipment like micro:bits or computers. Six of the teachers reported that they did not have optimal access to programming equipment at their school. One of the teachers expresses it this way:

> My school has been under savings for the last few years and it has been difficult to obtain funds for the purchase of new equipment. This is why I searched the web this school year and tried to find other opportunities to get support for purchasing new technological equipment. (Teacher 55)

A few of the teachers also experienced that the math in programming could be challenging, both for some of the pupils:

> The challenge in some of the programs is to understand the programming and especially the mathematics, which is behind some of the conversions (Teacher 7)

And sometimes for the teachers:

> Numerical methods of derivation, integration and differential equations were new to me - partly also mathematical, but I am beginning to approach an understanding of this now - even if I have to work more on programming this. (Teacher 50)

Two of the teachers state that the pupils' digital competence can be a challenge, more specifically they express that pupils may have trouble finding information online or having trouble saving files in a good way. For example, one of them state:

> At the same time, I know that some pupils in secondary school often find it difficult to find information themselves. This, I think, will be one of the biggest challenges (and therefore also for the learning) for the pupils. (Teacher 27)

## 4.5 Summary of results in the context of attitudes and self-efficacy

### 4.5.1 Placing the results in the context of self-efficacy

The results from the reflection notes are used as an indication towards how teachers perceive their self-efficacy in teaching programming. Teacher self-efficacy refers to the teachers' belief in their ability to perform certain actions in order to produce a given result. To place the results of the reflection notes in the context of the teachers' self-efficacy, the researcher uses the teachers' statements about their feelings and attitudes towards teaching programming as an indication to how they perceive their self-efficacy. For example, teacher 26 expressed "Through the studies in Introductory Programming and Applied Programming I believe that I have been given a good enough foundation to be able to teach elective programming". This can be seen as a statement connected to how able teacher 26 feels towards programming, and can indicate that teacher 26 have a relatively high sense of self-efficacy towards teaching elective programming in general. If teachers perceive something as challenging in their teaching, this is used as an indication that their self-efficacy towards that specific aspect is not at an optimal level.

There are also some blurry lines in the assumption that the results can be interpreted in this way. For example, when teacher 42 say "When I look back at the lesson plan I have made, I am happy with the results and think it will work well in my class", this can be interpreted as both an efficacy expectation and an outcome expectation. It can be interpreted as an efficacy expectation in the way that the teacher has developed the lesson plan himself, and therefore with his own capabilities in mind. When the teacher expresses that they believe it will work well, one can interpret this in the way that the teacher believe it will work well because he also believe he has the capabilities of implementing the lesson plan (performing the actions necessary for it working well). It can also be interpreted as an outcome expectation, since the statement simply say that the teacher believes the lesson plan will work well, and not specifically anything about whether he feels capable of performing the actions necessary for the implementation to work well. In this specific example, the researcher assumes that the teacher also to some extent feel capable of implementing the lesson plan, due to the teacher being the one that actually created the lesson plan with his own teaching and pupils in mind. Such assumptions have been made in the analysis of the reflection notes, and it is therefore important to note this due to the validity and reliability of the study, as another researcher might had interpreted the statements in another way. These results should therefore also be seen as an indication towards the self-efficacy of the teachers, and not a precise and correct measure.

### 4.5.2 Attitudes towards programming in school

From the analysis of the reflection notes it is indicated that the teachers are positive towards the inclusion of programming in school, and perceive it as relevant to use and to teach to their pupils. There is also an indication that the programming education has made a positive impact on some of the teachers' attitudes towards programming, and teachers report that they have gained interest in programming and are motivated to learn more. There is also a positive trend towards the use of physical devices and microcontrollers in teaching of programming. Many teachers also express that they are positive towards using programming interdisciplinary, and that programming can be a useful tool also in teaching of other subjects than programming itself.

Some of the teachers express some worry on how programming is included into the curriculum. Some teachers express that they believe it will be important to focus on raising teachers' competence in programming in the years to come, and that they would like more colleagues to work with in their teaching of programming.

Some teachers express that they find collaboration with other teachers very useful in teaching and learning programming.

### 4.5.3 Teachers self-efficacy in teaching programming

There is an indication that the programming education has had a positive impact on the teachers' self-efficacy towards teaching programming in general. Many teachers report that they feel more secure in teaching programming, and most of those who implemented their lesson plan are positive towards the implementation and the result. Most of those who did not implement their lesson plan in class also express that they believe it will work well, and some state that they look forward to teaching programming. There is an indication that the teachers have a relatively high sense of self-efficacy when it comes to their belief in their teaching of programming in general. Part of the teachers report how they can improve their lesson plans, and their teaching. Part of the teachers also believe they will get better in teaching programming by getting experience. Some of the teachers report that they find the didactics in teaching programming challenging, namely what to teach and how to teach it.

Many of the teachers perceive adapted teaching as a challenge in teaching programming, especially in terms of pupils' varied or low prior knowledge in programming. Some of the teachers also express that they have ways of addressing this problem, by differentiating by skill level, let the pupils work in groups, or have exercises with different level of difficulty. Some of the teachers that implemented their lesson plans experienced that adapted teaching was not a very big challenge, and that their lesson plan differentiated well. One teacher expressed that it was easier to give suitable challenges to more capable pupils than to pupils that do not find programming very interesting. The teachers' self-efficacy in adapting their teaching to their pupils in programming seem to be varied.

Some of the teachers included some kind of assessment in their lesson plan. One of these found it challenging to give the pupils a grade that was fair. Some others reported that they found assessment in programming challenging more in general, and that it is important to have clear learning objectives when teaching programming. It is a small indication that the teachers' self-efficacy when it comes to assessment in programming is not very high.

Part of the teachers talked about motivation as an important aspect when teaching programming. Many of these teachers expressed that they believed they somehow could motivate their pupils in programming, an indication that the ones that talked about motivation had a relatively high sense of self-efficacy when it comes to motivation in programming.

All of the teachers have made a lesson plan, as part of the final delivery and project in the spring course applied programming. Many of the teachers reported that they were happy with what they had made, and that they either believed it would work well, or that it did work well. Many of the teachers based their lesson plan on existing teaching material, and some express that there are very many good teaching resources in programming online. A few report that there are so many teaching resources that it can be difficult to navigate and locate the most suitable resources. The teachers seem to have a relatively high sense of self-efficacy when it comes to developing and adapting teaching material in programming.

### 4.5.4 Programming skill

Most of the teachers report that they feel positive towards their own learning of programming in their studies, and a part of them express that they possess basic programming skills. Even though many of the teachers are positive towards their own learning of programming, many perceive their own programming skill as relatively low, and that they will probably have pupils that have better programming skills than themselves. There is an indication that the teachers do not perceive their low programming skill as a challenge to their teaching. Some also report that they can use the more capable pupils as a resource in their teaching of programming.

Some of the teachers report that they can increase their programming skill on their own, some others report that they had to refresh their programming knowledge in the spring course because their programming skill had "faded" a bit over time.

The teachers seem to have a varied and relatively low sense of self-efficacy when it comes to their own programming skills, but this do not seem to have a significant impact on their self-efficacy towards teaching programming.

### 4.5.5 Challenges in teaching programming

Many of the teachers reported that time was an issue, and that they believed it would be challenging to find time to include programming into the current curriculum, and especially in other subjects. Many also experienced technical issues in some form. Either with the microcontrollers, computers or computer programs. Another challenge reported in the reflection notes was school resources, that the teacher's school did not have sufficient equipment for their teaching of programming. A few of the teachers reported that the math in programming could be challenging, both for the pupils and sometimes for the teacher. Two of the teachers reported that pupils' digital competence could pose a challenge, as some pupils have trouble finding information online, as well as structuring files and saving files in a good manner.

# 5 Main interview study: Teachers from the 2018-19 programming education and their self-efficacy and attitudes towards teaching programming.

## 5.1 Chapter overview

This chapter presents the interview study with teachers that took the continuing education in programming at NTNU in 2018-19 (see chapter 3 for more information on the continuing education program). Semi-structured interviews have been conducted with the teachers, the interviews have been transcribed and analyzed based on a self-efficacy scale and a grounded theory approach.

This chapter first describe the process and method used in the interview study. Then results are presented, and lastly the results are summarized in the context of self-efficacy.

This chapter explores the research questions:

**RQ1:** What attitudes do in-service teachers with programming education have towards programming in school?

**RQ2:** How do in-service teachers with programming education perceive their self-efficacy in teaching programming?

**RQ3:** How do in-service teachers perceive that programming education has affected their self-efficacy in teaching programming?

**RQ4:** How do in-service teachers perceive the lasting effect of programming education in regard to their self-efficacy in teaching programming?

A summary of the results from the main interview study can be seen below in table 5.1.

**Table 5-1: Summary of results from interview study**

| | |
|---|---|
| **Teachers' attitudes towards programming** | • The teachers are positive towards programming in school<br>• The teachers perceive programming as relevant in school and in several subjects<br>• Some teachers believe the inclusion of programming in the curriculum will not be a quick and simple process<br>• Teachers want colleagues to collaborate with in programming<br>• The teachers perceive that their colleagues' attitudes towards programming are very varied. One experienced resistance from colleagues when it came to programming |

| | |
|---|---|
| | • Some of the teachers mention gender differences in relation to programming |
| **Teachers’ self-efficacy towards teaching programming** | • The teachers can teach programming in their own teaching subjects and grades<br>• The teachers have a high sense of self-efficacy in adapting their teaching to their pupils<br>• The teachers have a more varied sense of self-efficacy towards assessment in programming<br>• There is a need for assessment tools and strategies. Oral presentation is indicated as a good method for assessment in programming<br>• The teachers have a high sense of self-efficacy towards motivating their pupils<br>• The teachers have a moderately high sense of self-efficacy towards explain and conveying programming knowledge<br>• The teachers have a high sense of self-efficacy towards developing lesson plans<br>• The teachers feel their programming skill is sufficient for their teaching<br>• The teachers perceive their programming skill as relatively low<br>• The teachers find programming hard to learn<br>• Pupils digital competence and computer skills are a challenge in teaching programming<br>• Technical issues are a challenge in teaching programming |
| **Impact of programming education on teachers’ self-efficacy** | • Positive impact on teachers’ self-efficacy in teaching programming<br>• The introductory programming course had a steep learning curve and a big workload. Learning programming is hard<br>• Second course had more balanced workload. Two teachers felt this course was too easy or not interesting enough<br>• Some want more focus on didactics |
| **Lasting effect of programming education on teachers’ self-efficacy** | • Teachers feel their programming skill lowers over time, but can easily be re-trained<br>• The teachers’ self-efficacy in teaching programming increases with experience over time<br>• The teachers’ self-efficacy in teaching programming does not significantly decrease without experience over time |

## 5.2 Method

### 5.2.1 Semi-structured interview

This study explores teachers' views and feelings towards their capability of teaching programming. Interviews can be used to gain insight into the research participants' experiences, views and opinions (Kvale, Brinkmann, Anderssen, & Rygge, 2015). To be able to answer the research questions, semi-structured interviews were chosen. With semi-structured interviews it is possible to have a structure in the interview while there is room for participants to elaborate, and for the interviewer to ask follow-up questions where appropriate. Semi-structured interviews are also most appropriate when the interviewer is closely involved with the research process, for example when the researcher is also the interviewer (Robson & McCartan, 2016), as in this study, because the researcher will have insight to what follow-up questions are relevant to ask, as well as what subjects to elaborate on.

### 5.2.2 Selection of participants

To be able to connect the results from the analysis of the reflection notes from the 2019 applied programming course, the main group targeted for interviews were students from the 2019 applied programming course. In addition, students from this year's (2020) applied programming course were also targeted for interviews (see chapter 6), to be able to see if compare the results with last years' students/teachers, and also for strengthening the quality of the research.

### 5.2.3 Recruitment

For the students of the 2019 course, the unit of continuing education at NTNU was contacted and asked if they could reach out to former students of the applied programming course through email. Emails were sent out to the 2019 students, and those who were interested in participating responded to the researcher's email address, which was attached in the email they received. Interviews were then planned with each individual participant.

### 5.2.4 Participants

All participants are Norwegian in-service teachers in either secondary school (grade 8-10) or high school (grade 11-13). The teachers' main subjects vary.

**Table 5-2: Overview of interviewed teachers from the 2019 course, grade, teaching subjects, subjects they expect to use or are using programming in, and if they teach in general or vocational education.**

| **Participant**<br>Number-gender-subject-grade<br>HS: High school<br>SS: Secondary school | **Teaching grade** | **Subject(s)** | **Programming subject(s)** | **General/Vocational education** |
|---|---|---|---|---|
| 1-Female-Economics-HS | High school | Economics | Economics | Vocational |
| 2-Male-Math-HS | High school | Math, physics | Math, physics | General |
| 3-Female-Math-HS | High school | Math, natural science | Math, natural science | Vocational |

| | | | | |
|---|---|---|---|---|
| 4-Male-Math-SS | Secondary school | Math, natural science, programming | Math, natural science, programming | General |
| 5-Female-English-SS | Secondary school | English, social studies, gymnastics, programming | Programming | General |
| 6-Male-Math-HS | High school | Math, physics | Math, physics | General |
| 7-Female-Math-HS | High school | Math, computer and electronics | Math | Vocational |
| 8-Female-Math-SS | Secondary school | Math, natural science, religion, programming, work-related training | Math, natural science, programming | General |
| 9-Female-Math-HS | High school | Physics, math, programming, natural science, technology and research | Math, programming, natural science | General |
| 10-Female-Construction-HS | High school | Construction- and control technique | Construction- and control technique | Vocational |

### 5.2.5 Interview guide

The interview guide was constructed based on four main aspects: Attitudes towards programming in school, self-efficacy in teaching programming, self-efficacy in programming skill, and the outline and content of the courses introductory and applied programming. The interview guide used for interviews with teachers from the 2019 course can be found in appendix A.

In regard to attitudes and RQ1, open questions were created by the researcher, so the teachers could answer freely on how they felt towards programming in school, and the researcher could follow-up with questions when appropriate. These questions were first focused on programming in school in general, and then towards the teachers own teaching subject and the inclusion of programming in these.

There exist several instruments for measuring self-efficacy, but very few in the context of teaching programming (Hartell et al., 2019, p. 197). No suitable results were found by the researcher either. To be able to look into teachers self-efficacy in programming through interviews, the often-used Teachers' Sense of Efficacy scale (TSES) (Tschannen-Moran & Hoy, 2001) was used as an inspiration for creating interview questions towards self-efficacy in teaching programming and RQ2. This study adapts the items from factor 1: Efficacy for instructional strategies, from TSES, as these are the items relevant for this study. The other two factors in the scale goes into classroom management and student engagement, and are not explored in this study. The items used have been translated to Norwegian by the researcher, for the purpose of making the scale more accessible for the teachers in their native language. A second alteration done, is placing the items in the domain of teaching programming. This was done to tailor the scale to the particular domain of functioning that is the object of interest (Bandura, 2006, pp. 307-308). In the interview guide, the words "to what extent" was also changed to "do you feel that you", since the interview study wanted to explore how the teachers felt towards their capabilities rather than asking them to rate their skills. This was also done to make it easier for the teachers to elaborate on what they found difficult or easy. The items adapted from TSES factor 1: Efficacy for instructional strategies can be seen in table 5-3.

**Table 5-3: Items adapted from TSES factor 1. In English and Norwegian.**

| Item | TSES-English | Adapted-Norwegian |
|---|---|---|
| 1 | To what extent can you use a variety of assessment strategies in programming? | Føler du at du kan bruke varierte vurderingsmetoder i programmering? |
| 2 | To what extent can you, in your teaching of programming, provide an alternative explanation or example when students are confused? | Føler du at du i programmeringsundervisning kan komme med alternative forklaringer eller eksempler når elever har behov for dette eller ikke forstår? |
| 3 | To what extent can you, in your teaching of programming, craft good questions for your students? | Føler du at du i programmeringsundervisning kan lage gode oppgaver for elevene? |
| 4 | How well can you implement alternative strategies in your teaching of programming? | Føler du at du kan bruke alternative læringsstrategier i din programmeringsundervisning? |
| 5 | How well can you respond to difficult questions regarding programming from your students? | Føler du at du kan svare på vanskelige spørsmål fra elever i programmeringsundervisning? |
| 6 | How much can you do to adjust your teaching of programming to the proper level for individual students? | Føler du at du kan tilpasse din programmeringsundervisning til den enkelte elevs behov? |
| 7 | To what extent can you gauge student comprehension of what you have taught in your teaching of programming? | Føler du at du kan vurdere hva elever har forstått din programmeringsundervisning? |
| 8 | How well can you provide appropriate challenges for very capable students in programming? | Føler du at du kan tilby tilpassede utfordringer til veldig kompetente elever i programmering? |

In addition to these questions, a similar question about motivation was also added, based on the results from the analysis of the reflection notes. The questions were also adjusted in terms of wording to make them more understandable after running three test interviews with pre-service math teachers. In regard to the teachers' programming skill, which is also relevant to teachers ability to teach programming (Korkmaz, 2013), questions were created to answer how the teachers felt towards their own programming skill.

In regard to RQ3, questions about teachers perceived self-efficacy before and after the programming education were formed, as well as questions on how they perceived that the programming education had impacted their self-efficacy towards teaching programming. These questions were first focused on how the teachers felt towards their capability of teaching programming before and after the education in programming, and later on, focused on how the programming education had impacted their self-efficacy towards teaching programming. The teachers were also asked how content they were with the two courses in the programming education.

In regard to RQ4, questions about teachers perceived change in self-efficacy in the time after finishing the programming education were created the same way as the questions for RQ3, but with a focus on the time after they finishes the programming education.

The questions and interview guide were tested through three test-interviews with three pre-service math teachers, and the wording of the questions were adjusted so that they would be more understandable. The order of the questions were also changed to some extent after the test interviews to make the interview process more coherent.

To see how the COVID-19 situation might had impacted the teachers' answers in the interviews, questions regarding this was added at the end of the interview guide.

The interview guide used in the interviews with the teachers from the 2018-19 programming education can be found in appendix A.

### 5.2.6 The interview process

All interviews were conducted through the digital video communication computer program Zoom (Zoom Video Communications Inc, 2020). To ensure the privacy and security of the communication, the researcher first made sure that the use of Zoom would be compliant with GDPR and privacy issues by contacting the NTNU IT department, as well as finding information that confirmed the security of using Zoom (Uninett, 2020). To further secure the online meetings in Zoom, precautions were made by securing the meeting from any other participants than the interviewer and the interview subject. The interviews were recorded on an external digital recorder with no access to internet to make sure personal information would be securely stored.

The interviews followed a sequence structure found in Robson and McCartan (2016, p. 290). The interviews were started by an introduction where the researcher presented himself, explained the purpose of the interview and the study, assured confidentiality, and asked permission to record the conversation. This was followed up with starting the recording and easy non-threatening warm-up questions, to create a safe atmosphere. Then the main part of the interview followed, as specified in the interview guide. The main part of the interview was ended by some cool-off questions, like asking the participant whether there was something they would like to add or clarify. The interviews were ended by stopping the recording, thanking the participant, making sure they knew how to get in touch if they had questions or wanted to withdraw from the study, and saying goodbye. The only part of the interview structure that differed from the sequence in Robson and McCartan (2016) was an addition in the cool-off part where the researcher also asked if the participant thought that the COVID-19 situation could have impacted their answers in any way.

### 5.2.7 Ethical aspects in interview studies

Interview studies are filled with moral and ethical questions (Kvale et al., 2015), and it is therefore necessary to make ethical considerations in such research. The Norwegian Research Ethics Committees for Social Sciences and the Humanities, NESH, has developed research ethics guidelines for the social sciences, humanities, law and theology (NESH, 2016).

Two important ethical considerations in interview research are consent and confidentiality. The consent requirement must ensure the research participants' freedom and self-determination, in addition to the consent being based on information about the project's purpose and consequences that may have an impact on the participants (NESH, 2016). The participants in this study gave consent via email, through a document of consent (see appendix C), which they also received their own copy of. Confidentiality is about not disseminating information in ways that can identify the informant (NESH, 2016). In this study, the names of the participants were anonymized, in addition to the school and the area and personal information, to secure the participants' identity.

As the recording of an interview causes personal data to be processed, this master's project was reported to NSD (NSD, 2020) and approved (see appendix D). The recordings have only been stored on the digital audio recorder provided by the responsible institution for the research, and the recorder has, when not being used, been kept in a locked drawer where only the researcher had access to the key. After the analysis of the interviews, the recordings were deleted.

### 5.2.8 Analysis

The interviews were first transcribed. Transcribing an interview-recording involves structuring the interview so that it is better suited for analysis (Kvale et al., 2015). The interviews were first transcribed roughly verbatim. Filler words ("uhm", "eeh", etc.) were not transcribed. After the transcription, all transcriptions were proof-read and corrected while listening to the recording corresponding to each transcript.

The coding of the transcript was done in an experimental way that included two methods. The first coding method can be seen as a theoretical thematic analysis where the data was coded specifically for a research question (Braun & Clarke, 2006). In this first coding method, the data was coded based on RQ2 and the Teachers' Sense of Efficacy scale, factor 1 (see table 5-3), as well as the aspect of motivation that emerged in the analysis of the reflection notes. In this coding process the researcher started out with the main category *teaching programming self-efficacy*, and the 5 sub-categories seen in table 5-4. The data was then coded as sentences and sections into these sub-categories, based on what sub-category the data represented or had relation to. The coding process was done with the computer program NVivo (QSR International, 2018).

**Table 5-4: Sub-categories with sample sentences from the theoretical thematic analysis.**

| Sub-category | Sample sentence (I: Interviewer, P: Participant) |
|---|---|
| Assessment (TSES items 1 and 7) | I: Do you feel that you can use different assessment methods in your teaching of programming?<br>P: Yes.<br>I: Yes, easily?<br>P: Yes, written, oral, and practical.<br>Interviewer: Do you feel that works well too?<br>Participant: Yes, I would say that.<br>(Teacher 9-Female-Math-HS) |

| | |
|---|---|
| Adapted teaching (TSES items 6 and 8) | I: Can you adapt your programming teaching to your pupils?<br>P: Yes. I can.<br>(Teacher 2-Male-Math-HS) |
| Motivation (From analysis of reflection notes) | I: Do you also feel that you can motivate pupils with different needs?<br>P: Yes, I believe so. There is not much difference between programming and other subjects really, so that's ok.<br>(Teacher 10-Female-Construction-HS) |
| Explaining and conveying knowledge (TSES items 2 and 5) | I: If pupils do not understand a concept, do you feel that you can come up with an alternative explanation?<br>P: Yes, I have actually done that.<br>(Teacher 3-Female-Math-HS) |
| Developing teaching material (TSES items 3 and 4) | I: Do you feel that you can create good lesson plans, that include programming?<br>P: Yes, I can, because there is so much online, and it is great that so many share online. I use a lot of these pages that are out there, and download, so there's a lot.<br>I: So you find it easy to find such…<br>P: Yes, that's no problem.<br>I: Do you find it easy to adapt those exercises to…<br>P: Yes.<br>(Teacher 10-Female-Construction-HS) |

The second coding method used was the same as in the analysis of the reflection notes (see chapter 4.3.2). Like with the reflection notes, the researcher first used open coding, then categorization and axial coding. This was done to find new categories, sub-categories and relations in the data material. After this process the results from the two coding methods were merged together. The final categories can be found in table 5-5.

**Table 5-5: Categories after merging results from the two coding methods**

| **Category** | **Description** |
|---|---|
| Attitudes towards programming in school | Statements regarding teachers' attitudes towards programming in school |
| Teaching programming self-efficacy | Coding from the theoretical thematic analysis (see table 5-4) |
| Programming skill | Statements regarding the teachers' perceived programming skills |
| Impact of programming education | Statements regarding the impact of programming education on the teachers' programming skills and self-efficuacy |
| Impact of time after programming education | Statements regarding the impact of the time after programming education on the teachers' programming skills and self-efficacy |

| | |
|---|---|
| Digital competence | Statements regarding pupils' and teachers' digital competence |
| Course-specific reflection | Statements connected directly to the course IT6203 and IT6204 |
| Covid-19 | Statements regarding the impact of Covid-19 on the teachers' answers |

## 5.3 Results

In this sub-chapter results from the analysis of the transcription of the interviews are presented. All quotes are indented for readability. All the quotes have also been translated from Norwegian to English by the researcher.

### 5.3.1 Attitudes towards programming in school

#### 5.3.1.1 Teachers' personal attitude towards programming in school

Common for all ten teachers are that they are in general positive towards programming in school. One of the teachers verbalized:

> I think it's great. I'm convinced it's terribly important, and perhaps crucial for many in future jobs, that they have an understanding of and can use programming. I believe it is greatly underestimated how many places you need to know about programming. There are many who do not realize how many professions involve programming. And we can't put our pupils in a situation where they haven't seen programming before coming to high school. (Teacher 5-Female-English-SS)

Most of the teachers are also positive towards teaching and using programming in their own teaching subjects. One of the teachers put it this way:

> I think it seems very positive, because computational thinking is very important, considering if you look at algebra for example, where splitting a problem into smaller problems, which you can actually do in programming, which again is very easy to look at or somehow understand if you know some programming … getting this thinking into more subjects, especially math, is profitable. But also in natural science, because in natural science programming will be a good tool for conducting experiments. (Teacher 4-Male-Math-SS)

One of the teachers is positive towards pupils learning programming, but is not convinced that it necessarily will benefit the pupils' math skills:

> I believe it is right that pupils need to learn programming, more coding than they do today. But I am not convinced that they - not necessarily get any better at math from it, in the subject of math. I might be wrong, we have computational thinking and things that of course are relevant. But I'm not quite convinced yet (Teacher 2-Male-Math-HS)

Two of the teachers also express that they do not think implementing the new curriculum with programming into different subject will be easy or a "quick-fix". For example:

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

#### 5.3.1.2 Community and collaboration

One of the sub-categories that emerged from the analysis of the interviews was teachers' community and collaboration in programming in school. Four of the teachers expressed that they found collaboration with colleagues important when dealing with programming in school. Those who had someone to collaborate with expressed that this was very useful to them:

> I have one colleague whom I took the course with last year, and we were very happy that we were two at that time. And I know I'm happy we're two now as well. So that we can – we are doing this somewhat together. (Teacher 9-Female-Math-HS)

Two of the teachers expressed that they would like a colleague to work with in programming, one expressed it this way:

> I feel that I should have had more knowledge, I should have more, I should have a bigger bank of knowledge myself, or at least had a colleague to work with. I am missing that. (Teacher 5-Female-English-SS)

When it came to how the teachers perceived colleagues' attitudes towards programming in school, the answers were very varied. Two of the teachers felt that their colleagues and leaders were to some extent positive towards programming, for example:

> Maybe I'm a little more positive since I have the subject elective programming. We are two teachers who have the subject at our school. We also have a management that is a bit focused on it. So it was probably that way it was started, as well as some others being willing as well. (Teacher 8-Female-Math-SS)

One teacher also expressed that teaching programming would be a challenge for teachers without competence in programming, but that at their school they were relatively well prepared:

> But the teachers who have no knowledge in programming, they will have a challenge. But at our school in general, we are pretty well equipped, in science especially, there are quite a few who can program now. (Teacher 8-Female-Math-SS)

Four of the teachers expressed that their colleagues did not share their positive attitude towards programming and were more negative than positive. One of these teachers expresses that:

> I have two colleagues who played a bit with an old Lego League set, but other than that is programming a "no-no" subject. And I've tried, in different rounds, to engage management and other colleagues in programming. But have resigned a bit and have come to terms with that I have to sit back and wait in anticipation for how it will be solved when the new curriculum is implemented. Because I have tried. (Teacher 5-Female-English-SS)

Two of these teachers express that the age of their colleagues might be the reason why they are not as positive towards programming, and that younger science teachers see more possibilities with the inclusion programming. One of the four teachers also experienced resistance from other teachers when it came to programming:

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

Two of teachers perceived that their colleagues' attitudes towards programming were fairly varied positive and negative.

Two of the teachers also mentions gender differences in the context of teaching programming in school. One of the male teachers expresses that male teachers might be more interested in programming than female teachers, and also the implementation of it:

> I notice - it is probably a bit traditional this stuff - that boys are more experimental than girls, and thus also are male colleagues are more interested in programming than female colleagues, and in implementing this. (Teacher 4-Male-Math-SS)

One of the female teachers experienced that even though she was the only one teacher at her school with education in programming, male teachers got much the responsibility of teaching programming and ordering programming equipment. She summarizes this:

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

## 5.3.2 Teaching programming and instructional self-efficacy

The results in this sub-chapter is the combination of the two coding processes explained in chapter 5.2.8 This sub-chapter describes how the teachers perceive their own capability of teaching programming.

### 5.3.2.1 General feelings towards teaching programming

In general, all ten teachers express that they feel they can teach programming in the grades and in the subjects that are relevant for themselves. This is indicated in statements like:

> I feel that I could have taught programming after finishing the course before Christmas. (Teacher 2-Male-Math-HS)
>
> *Interviewer:* So, after you finished the continued education studies in programming, did you feel that you could teach programming then?
> *Participant:* Yes, I did. (Teacher 6-Male-Math-HS)

Some of the teacher that do not teach a pure programming subject express that they do not feel competent enough to teach in subjects other than the ones relevant for themselves. For example:

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

When asked how comfortable the teachers felt towards teaching programming, five of the teachers expressed that they felt comfortable teaching. One of the teachers for example said:

> *Interviewer:* Do you feel comfortable teaching programming?
> *Participant:* Yes, I do. Because there are not so many people who can. (Teacher 3-Female-Math-HS)

Some of the teachers also expressed that they felt comfortable with teaching programming, but only in their own teaching subjects and grade. For example:

> Yes, I feel comfortable teaching at secondary school. I feel that high school probably requires a little more. I have little experience, but I believe it's fine at secondary school, if the pupils are "on", and want more, I hope I will be able to help them so they can feel mastery at all levels. (Teacher 4-Male-Math-SS)

### 5.3.2.2 Adapted teaching

Adapting teaching to pupils were one of the aspects the teachers in the interview study were asked about. Of the ten teachers, nine of them expressed that they felt confident in adapting their teaching of programming to their pupils. This is seen in statements like:

> Interviewer: Do you feel that you can adapt your teaching of programming to your pupils
> Participant: Yes, I do. (Teacher 6-Male-Math-HS)

Differentiating to the different pupils and creating open exercises are the two main methods teachers mention when talking about adapted teaching. For example:

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

One of the teachers expresses that she finds it easy to adapt teaching in programming:

> After all, it is perhaps the easiest subject to give students challenges at different levels … so yes, it's very easy to differentiate. (Teacher 9-Female-Math-HS)

One of the teachers feels she cannot adapt her teaching very well due to not being confident enough in programming:

> No, I think it's because I am not confident enough, and then it's harder to adapt, because I don't know how to adapt because I'm not familiar enough with it. (Teacher 1-Female-Economics-HS)

All ten teachers express that they can provide appropriate challenges for very capable pupils in programming. One of the teachers verbalizes it this way:

> Yes, I can. I have some plans for that, and it's all about providing - for example in code.org, there are probably some that, when we've gone through this, spent a month or so, that think this is simple and ridiculously easy. So that they can follow their own pace. Like last year … I could demand a little more of them, but they wanted a little more as well. (Teacher 4-Male-Math-SS)

Also the teacher that did not feel she could adapt her teaching very well, expressed that she could provide appropriate challenges for very capable pupil through giving them freedom in what they were doing:

> Yes. One of the pupils went above and beyond all ability I had in the app he was going to create. And this is about the fact that they had free reins to create whatever they wanted, as long as there was content within the curriculum goal. So he - I hadn't been able to program what he was doing, to put it that way. But I believe that he received his appropriate challenges too, without me as the one laying out the guidelines. (Teacher 1-Female-Economics-HS)

#### 5.3.2.3 Assessment

Teachers feelings towards assessment in programming were more varied. Four of the teachers express that they find assessment in programming difficult. One teacher answer that she cannot use any methods of assessment in programming, because there has been no focus on it at her school, and that she will have to find out when programming is implemented more:

> As of today, it has not been any assessment in programming, we have just used it kind of to "spice up" the lessons. It will not begin until the autumn, but I see, at least at our school, that there is no plan on how programming should be included, so this is something I will find out next year. I don't know. (Teacher 10-Female-Construction-HS)

The three others believe that assessment is, and will be, challenging, for example because it is easy to find solutions online for the pupils, and that they need strategies and tools for assessment. For example:

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

Five of the teachers express that they feel they can assess their pupils, and gauge pupil comprehension. The teachers especially find oral presentation as a method for assessing useful, also to unveil whether the pupils understand their solutions or if they have copied it. One of the teachers explains it this way:

> What we have often done if they have a submission for work they have done, is that they must explain the code orally. “Explain why it is like this”. They either deliver an audio file, or they explain to me. And then I think it is revealed pretty quickly, because you can for example say: "in that block there it starts with “if”, what does it mean?" and then I think that their understanding clearly shows at least, especially when it comes to copying code and such. (Teacher 8-Female-Math-SS)

One of the teachers that finds assessment challenging feels that it was little focus on this in the programming education as well:

> Also, regarding the course we took, there was little about assessment. So, it's not something we've worked on. (Teacher 6-Male-Math-HS)

#### 5.3.2.4 Motivation

When it came to motivation, and if the teachers felt that they could motivate their pupils in programming, nine of the teachers expressed that they felt they could motivate their pupils in programming. Some also felt that it was easier to motivate pupils in programming than in other subjects. For example:

> Yes. In a way, they get to try something that is not in the curriculum, they get to try something new. And those who have some prior knowledge get to do something they master. So yes, I think it is easier to motivate them in programming than in accounting, for example. (Teacher 1-Female-Economics-HS)

One of the teachers find it difficult to motivate some of the pupils in his elective programming class because they are in the programming class because they did not get into the elective class they wanted to:

> It's about how you meet the students. And in discussing with them attempt to find angles of attack that motivate them. I feel that I manage that with some pupils, but then there is a problem in that not all pupils in elective programming are necessarily motivated to learn programming. It is not ideal, this stuff. Some of the pupils are there just because they did not get into the elective they wanted, and they choose to do nothing - that's a challenge. (Teacher 4-Male-Math-SS)

#### 5.3.2.5 Explaining and conveying programming knowledge

Explaining concepts and conveying knowledge such that the pupils understand can be connected both to adapted teaching and assessment, as the teacher need to know what the pupil knows, and also how to explain things in a way the pupil can understand. Of the ten teachers, eight of them express that they feel they to some extent can explain programming concepts and also believe that they can come up with alternative explanations when pupils do not fully understand. This is seen in statements like:

> Yes, I believe so. I get them to use the concepts at least, but I have sometimes wondered "Do they really understand what they are doing now?". And I work towards making them think about it. But I’m not always absolutely sure, but they use what I explain, so I have to trust that they understand it (laughs). And I believe I can explain on different levels. (Teacher 5-Female-English-SS)

The two other teachers are not sure if they can come up with alternative explanations for all concepts, but one of them explain that she can utilize other pupils to help explain when others do not understand:

> I don't know, I would think that I try at least, but if I have such a conscious relationship to it, I'm not so sure. It is not a question I have had so much on my mind really. But I probably do it. I often use other pupils in that situation. I mean like, "how did you understand it?", right. (Teacher 10-Female-Construction-HS)

When asked if they could answer difficult or more advanced questions from more capable pupils, six of the teachers answered that they could not. Still, nine of the teachers explained that they could either come back to the pupil with the answer later or find the answer together with the pupil. One of the teachers explains:

> No. But the pupils are also quite understanding when you say "I can't do this very well, but I find it very fun! And I want to show you, and then we can figure it out together". They understand that much, kind of. (Teacher 1-Female-Economics-HS)

#### 5.3.2.6 Developing teaching material

When it comes to development and design of lesson plans and exercises, eight of the teachers believe that they can create good lessons and exercises to their pupils, but many of the teachers also add that this is time consuming, and that they want more time to prepare for programming classes. For example:

> I can. But it takes time. If I had even better time, I could have made even better lessons. (Teacher 3-Female-Math-HS)

Two of the teachers explain that they cannot create good lessons from scratch, but that they can create good lessons by adapting existing teaching resources. One expresses:

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

Four of the teachers express that they would like more teaching resources in programming.

Most of the teachers feel that they can also create varied lessons in programming. Two of the teachers explain that they stick to what they feel is working rather than varying their lessons very much, for example:

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

### 5.3.3 Programming skill

When asked if they felt that their own programming skill were sufficient for their own teaching of programming, eight of the teachers answered that they felt it was. Some seemed very confident that it was good enough, others expressed that it was not sufficient for teaching subjects or grades other than their own. For example:

> Yes, as long as it isn't a subject that will have it throughout the whole year, but if it is part of the subject and a curriculum goal, then yes, then it is good enough. But as in a subject like technology and research, then no, it's not good enough. But for some of the subjects, yes. (Teacher 1-Female-Economics-HS)

One of the teachers felt that if her programming skills were sufficient was dependent on the day she was asked, and that if she did not feel comfortable in teaching programming, it was because of her programming skill and not didactical or pedagogical aspects:

> Interviewer: Do you have an example of when you don't feel comfortable? What is the teaching content, or what are the challenges?
> Participant: For example, it is when we have embarked on slightly larger projects, which I

> may not have complete control over the development in. But so far, I have not been on extremely thin ice, but I have felt that "oh, I have to go home and pick up the book and read some more", I have had some of those rounds with myself.
> Interviewer: Is that connected to your programming skill or didactic probl…
> Participants: My skills.
> Interviewer: So mainly your programming skills, and not didactic challenges?
> Participant: No, it's the skills. The programming skills. (Teacher 5-Female-English-SS)

Another common theme is that a part of the teachers perceived their programming skill as relatively low. Six of the teachers express that they feel like they have relatively low programming skills, and for example that pupils might be better than them. This is seen in statements like:

> I could have been much better in the elective programming. I could have had much greater skills, to be able to, like I talked about, giving challenges to the capable pupils and such. It's a little strange to say that, because I say that I'm not good enough in elective programming. (Teacher 8-Female-Math-SS)

Many of the teachers also found it very hard to learn programming and thought the learning curve in the first part of the education, where they learned introductory programming was very steep. Seven of the teachers expressed that they thought the introductory programming course was very hard, for example:

> In the first half year I didn't feel like I was mastering anything, I thought it was a steep learning curve and I thought the pace was fast. (Teacher 1-Female-Economics-HS)

Most of the teachers also report that they could, when necessary, increase their programming skill on their own, but time is often mentioned as the biggest challenge towards training and getting better at programming. Three of the teachers express that they would like more training in programming in form of education or courses. One of the teachers is currently taking more education in programming at a university, and one is participating in an online programming course.

Two of the teachers explain that they would have liked to have more follow-up in term of exercises or a local programming group after the education, to get better at programming as well as to maintain their programming skill. One of them state that:

> Yes, that it becomes just like an anonymous alcoholics group, that you have a follow-up group, "anonymous coders" who need some follow-up. Get some challenges and keep up. It's like playing football on the oldboys team, right. Then you keep it maintained, right. (Teacher 4-Male-Math-SS)

### 5.3.4 Challenges in teaching programming

When the teachers were asked what they perceived as the biggest challenges in teaching programming, two themes were most prominent: Pupils' digital competence and technical issues.

Three of the teachers talked about pupils' computer skills as a challenge, and that many pupils had trouble with using computers in a good way, maintaining a file structure, saving and naming files, and retrieving said files. For example:

> But that is because the pupils are terribly poor at structure from the beginning - that suddenly things have to be done in the right order. And if they only had structure for example on their file system in the file explorer then it would have helped. They do not have structure, they do not even know how to save. Several of them name their files "one" "two" "three" and such, they do not have any systems. So it will be difficult when you need to help them find a structure in programming when they can't even structure other things, so I think that might be most challenging. (Teacher 1-Female-Economics-HS)

Two of these teachers also add that they believe the pupils computer skills will eventually get better with the inclusion of programming into the curriculum.

Three of the teachers also talk about technical challenges. One of the teachers find that the computer programs for programming are difficult to use and cause technical problems, and that she would like more user-friendly programs for her pupils. Two other teachers explain that there is a lot of technical issues connected to the pupils using different versions of Python, different libraries and modules, and that such problems are hard to solve and time-consuming. One state that:

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

### 5.3.5 Impact of programming education on self-efficacy in teaching programming

Overall, the programming education seem to have had a positive impact on the teachers' self-efficacy towards teaching programming. Six of the teachers expressed that they felt they could not have taught programming before the course, but that they could teach programming to some extent after the education. For example:

> Interviewer: But did you feel you could teach programming before studying?
> Participant: No. Absolutely not.
> Interviewer: Connected to that, did you feel that you could teach programming, in general, after your studies?
> Participant: Yes. I think so. I have, and I do. I was even so "cocky" that I started teaching an elective programming class at the same time as I was studying. Of course, I am left with slightly different experiences from it, that may not have been the smartest thing I have done. But it went ok. A little bumpy, but it worked somehow. We came through the year. (Teacher 5-Female-English-SS)

Of the teachers that did not feel they could teach programming before the studies, three of them also had some prior knowledge in programming.

Three of the teachers felt that they could have taught or did teach programming before or during the programming studies, but that the studies helped them get better. For example:

> Participant: We felt that it was much trial and error, and that we had to try our way forward. After all, we relied heavily on "Kidsa Koder" and the resources available there. And it goes at least ok in a way, but you have nothing to give to those who are capable and such, because they are better than you already. But that is fine, and they still are. You know, there are some who are very good at programming, and that's fine.
> Interviewer: When you had completed the programming studies, those two subjects at NTNU, did you feel that you could teach programming then?
> Participant: Yes, but still it is kind of narrow you can say, there were a only a few fields we went through. But, I'm more confident in, what should I call it, the progression in the subject, what should we choose first, how should we go forward. It used to be more random or that we followed a website. (Teacher 8-Female-Math-SS)

One of the teachers felt more capable of teaching programming before the studies. She explains that she had been teaching some programming for two years before the programming education, but that the education had a negative impact on her confidence in teaching programming. She explains:

> Interviewer: So before you took the studies at NTNU, did you feel that you could teach programming back then?
> Participant: Yes, I did teach. Two years before that. But I didn't quite know what I was doing.
> …
> Interviewer: So, after the studies, did you feel more comfortable in..
> Participant: No. I was much more confused.
> Interviewer: In what way were you confused by the education?
> Participant: Well, because, I was brand new, and I started at NTNU, and the first course with python was aimed at science and math, and I'm not there at all. And I struggled terribly. The first and second tasks were ok, and then came the third and fourth and fifth tasks, then it became so overwhelming that I went into a (unclear recording). I couldn't sort it out, and it became so much that I just, "what should I here, and what should I use there?". (Teacher 10-Female-Construction-HS)

The same teacher did however specify that this was mainly in the first half year of the studies which focused on programming skills, and that the second half and course did have a more positive impact. The teacher also felt that she had sufficient programming skills for her teaching of programming, but that she would like to be more competent in programming.

Five of the teachers express that they studies have also had a positive impact on their attitudes towards programming in school. Especially towards why the inclusion is relevant, and that it can have benefits also interdisciplinary. For example:

> But yes, my attitude has changed. From a general attitude that programming is useful, to realizing what it can be used for, and that it can especially useful in math be used in mah. Python, for example, is an incredibly nice language, in which you easily can do a lot of calculations, like doing 10,000 calculations is ridiculously easy in Python, versus doing it by hand on paper. (Teacher 4-Male-Math-SS)

As seen in chapter 5.3.3, many of the teachers felt that the learning curve in the fall course, introductory programming, was very steep. Most of the teachers still felt that they learned a lot from the first course, and most were happy with both the learning outcome and the workload in the second course. Two of the teachers however, reported that they thought the introductory programming course was very good, and that they felt the applied programming course was either too easy or had little impact on their own competence:

> I think the course after Christmas affected minimally, I simply think it was bad. I had little benefit from it, and I had quit on February first had it not been that I knew there were money in the pot if I finished the exam. It's awful to say, but I'll be honest. (Teacher 2-Male-Math-HS)

> But after Christmas, it was too easy. It seemed as if the assignments given could be done on a few weekend courses, so there was simply too little to do, and too little connection between the exam and the assignments. (6-Male-Math-HS)

Two of the teachers expressed that they wanted more focus on how to teach programming and didactical aspects, for example:

> Yes, I think there could have been a lot more focus on what we were talking about earlier, how to do teach it in school. And don't just teach yourself programming. (Teacher 6-Male-Math-HS)

### 5.3.6 Impact of time after course

The teachers were also asked about how they perceived their competence in teaching programming had been affected by the time that had passed since they finished the programming education. The most common theme found was that the teachers perceived their programming skill lowered over time, when not used actively. Eight of the teachers stated that their programming skills lowers when they are not using it. For instance, one of the teachers stated that:

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

At the same time, these teachers expressed that they could “refresh” their programming skill and knowledge with little trouble.

The teachers that have taught or used programming in their classes in the last year, explain that they also feel more capable in teaching programming due to the experience. One of the teachers verbalized:

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

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

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

## 5.4 Results summarized in the context of attitudes and self-efficacy

In this sub-chapter, the results from the interview study are summarized and related to the context of attitudes and self-efficacy. As part of the interview questions were based of a teacher self-efficacy scale (see chapter 5.2.5), these result can easier be connected to self-efficacy than the results from the reflection notes (see chapter 4.4), but some of the issues of interpretation still persists, as the statements can also be interpreted in different ways here.

### 5.4.1 Teachers’ attitudes towards programming

The teachers are very positive towards the using and teaching programming in school. They also express that they are positive towards using programming interdisciplinary, and many report that they through the programming education see more possibilities and the relevance of programming in their own subjects. One of the teachers does not feel convinced that programming necessarily will help pupils get better at math.

Even though the teachers are very positive towards programming in school, some believe that the inclusion of programming will be a long process and that it is not a quick fix. Some of the teachers believe that it will be difficult for many of the teachers that do not have any competence or education in programming.

Teachers that have someone to collaborate with in programming express that they find this very useful. Some other teachers indicated that they do not have colleagues to work with in programming, and that they would like to have that.

How the teachers perceived their colleagues' attitudes towards programming varied. Two of the teachers felt that colleagues were relatively positive and that they had support from their management. Four of the teachers expressed that their colleagues were more negative than positive towards programming. One of these teachers also felt that she experienced resistance from some teachers when she talked about programming. Two teachers felt their colleagues' attitudes towards programming were fairly balanced positive and negative. Three of the teachers talked about gender differences related to programming. One male teacher expressed that male teachers were more interested in programming than female teachers. One female teacher experienced that male teachers got much of the responsibility of programming related tasks in school, when she was the only teacher with programming education.

## 5.4.2 Teachers' self-effiacy towards teaching programming

All of the teachers feel to some extent that they can teach programming in their own grade and teaching subjects. Some state that they could not have taught a programming course, but that they can teach programming as a part of their own teaching subjects. Five of the teachers expressed that they felt comfortable in teaching programming, some of the others stated that they only felt comfortable teaching programming within their own subjects and grades.

### 5.4.2.1 Instructional self-efficacy

Nine of the teachers felt that they could adapt their teaching of programming to their pupils, it is indicated to some extent that adapted teaching is not a big challenge for the teachers, and that many teachers had a high sense of self-efficacy towards adapting their teaching of programming. However, one of the teachers expressed that she did not feel she could adapt her teaching very well, due to not being confident in programming. All of the teachers expressed that they could provide suitable challenges to capable pupils in programming. Some explained that even though some capable pupils perhaps were better than them in programming, this was not a big challenge because they could usually find suitable challenges together with the pupils, or the capable pupils got suitable challenges through a freedom of choice in what they were working on.

If the teachers felt they could use different methods of assessment in programming was more varying. Three of the teachers wanted more tools and strategies available for assessment in programming. Five of the teachers expressed that they could use different methods of assessment in programming. Oral presentation was indicated as one of the better methods of assessment in programming. One of the teachers stated that there had been little focus on assessment in the programming education. Most of the teachers felt that they could gauge pupils' comprehension when in programming, some expressed that they could do this to some extent. It is indicated that the teachers had a more varied sense of self-efficacy when it came to assessment in programming.

Nine of the teachers felt that they could motivate different pupils with different needs in programming. One for example expressed that programming was something new for the pupils and that this could increase motivation in itself. One other experienced that motivating pupils in an elective programming class was challenging since some of the pupils were not motivated to be in the programming class in the first place. In general, the teacher seemed to have a relatively high sense of self-efficacy when it came to motivating their pupils in programming.

Most of the teachers felt that they to some extent could explain and come up with alternative explanations in programming. Two of the teachers expressed that they could explain some programming concepts, but probably not all. Six of the teachers believed that they would not be able to answer difficult programming questions from the more capable pupils, the other four believed they could answer some difficult questions, but not all. However, nine of the teachers believed that they could either come back to the pupil with the answer later or find the answer together with the pupil. The teachers seemed to have moderately high sense of self-efficacy in this theme. The main challenge towards answering difficult answers and conveying knowledge indicated is that the teachers do not perceive their programming competence as very advanced.

Eight of the teachers expressed that they can create good lesson plans and exercises. Some of these also add that this is time consuming, and that it would be beneficial with more time for planning lessons. Two of the teachers stated that they could not create good lessons from scratch, but that they could create good lessons by adapting existing teaching resources. Most of the teachers state that they are using and finding teaching material online, and adapt it to their classes. Four of the teachers expressed that they would like more relevant teaching material resources available, but one of these teachers also added that this view is not exclusive to the subject of programming. In general, the teachers indicated that they had relatively high sense of self-efficacy when it came to designing lessons in or with programming when there was relevant teaching material available that they could adapt to their own teaching.

#### 5.4.2.2 Programming skill

The teachers were asked how they perceived their own programming skill. Eight of the teachers expressed that their programming skills were sufficiently good for their own teaching in their subjects and grades. Six of the teachers indicated that their programming skill was relatively low, and that some pupils probably had better skills than themselves. It is indicated that the teachers did not have a very high sense of self-efficacy towards doing programming, but that their self-efficacy towards teaching programming was not severely impacted by this.

Most of the teachers stated that they could also increase their programming skill. Three of the teachers would like more training in programming, and two teachers are already getting more training. It seems that even though many teachers feel their programming skills are relatively low, they perceive that they can increase this skill. An indication of a higher sense of self-efficacy towards getting better in programming.

Seven of the teachers expressed that they thought it was hard to learn programming in the first half of the programming education which focused on learning introductory programming. But many of them also felt that they learned a lot.

### 5.4.3 Challenges in teaching programming

Three teachers perceived pupils' computer skills as one of the biggest challenges. That many pupils do not know how to use computers in a good way when it comes to structuring files, saving, and comprehension of how one should use a computer as a tool.

Three other teachers expressed that technical aspects could be very challenging. What programs to use in programming, and how to fix technical issues that appear.

### 5.4.4 Impact of programming education on teachers’ self-efficacy

Six of the teachers expressed that they could not teach programming before the programming studies, but that they felt they were able to teach it in their subjects and grades after the studies. This indicates a very positive impact on the teacher’s self-efficacy towards teaching programming. Three of the teachers felt that could have taught or taught programming before the programming education, but express that they felt more secure in their teaching after the studies. One teacher felt that the first half year of studies, namely the introductory programming course, had a negative impact on her ability to teach programming because the big workload and difficult exercises made her more confused than more competent. Five of the teachers also stated that the programming education had a positive impact on their attitudes towards programming in school. Especially in terms of why programming in school can be relevant, and that it can be beneficial in other subjects.

Most of the teachers felt that the introductory programming course had a very steep learning curve, and that it was hard to learn so much programming in such little time. Most teacher were more content with the workload in the applied programming course. Two of the teachers were not content with the applied programming course. One thought it was too easy and too little to do, the other expressed that it was bad and not very interesting. The same two teachers expressed that they thought the introductory programming course was very good.

Two of the teachers expressed that they wanted to have learned more about what to teach in programing and how to teach programming, namely, didactics of programming.

### 5.4.5 Lasting effect of programming education on teachers’ self-efficacy

The interviews were conducted close to one year after the teachers finished their programming studies. They were also asked how the time that had passed since the studies had impacted their capabilities to teach programming.

Eight of the teachers stated that their programming skill lowered over time, when not used. The same teachers also felt that they would have little trouble refreshing their skills and competence in programming. It is indicated that the teachers sense of self-efficacy towards their programming skill lowers to a certain extent when not used regularly, but that this might not have a significant impact on their self-efficacy in teaching programming.

Teachers that had taught programming in some form during the last year felt that their capability in teaching programming had become better due to the experience. The others felt that their capability to teach programming was the same as right after they finished their studies, except one teacher that stated he might was a bit more capable right after the studies, but that it would not be a problem when he started planning for teaching programming again. It is indicated that the teachers sense of self-efficacy in teaching programming has increased with teaching programming experience, and not decreased significantly over time without experience.

# 6 Supplementary interview study: Teachers from the 2019-20 programming education and their self-efficacy and attitudes towards teaching programming.

## 6.1 Chapter overview

This chapter presents the supplementary interview study with teachers that participated in the programming education for teachers at NTNU in 2019-20 (see chapter 3 for more information on the programming education program). Semi-structured interviews have been conducted with three teachers that were still participating in the study program when the interviews were conducted. The interviews have been transcribed, and analyzed based on a self-efficacy scale and the categories that emerged from the analysis of the main interview study.

This chapter first describes the process and method used in the supplementary interview study. Then the results are presented, and lastly the results are summarized in the context of self-efficacy.

This chapter explores the research questions:

**RQ1:** What attitudes do in-service teachers with programming education have towards programming in school?

**RQ2:** How do in-service teachers with programming education perceive their self-efficacy in teaching programming?

**RQ3:** How do in-service teachers perceive that programming education has affected their self-efficacy in teaching programming?

**RQ4:** How do in-service teachers perceive the lasting effect of programming education in regard to their self-efficacy in teaching programming?

The results indicate that the teachers from the programming education program in 2019-20 are also positive towards using and teaching programming in school, and perceive it as relevant to include programming in the curriculum. The teachers are however skeptical towards how programming is included into the curriculum, in terms of how beneficial it will be for the pupils to include programming in natural science and mathematics. One teacher is worried that the inclusion of programming into different subjects will result in significant differences in teachers’ programming competence. One teacher experience that colleagues are skeptical to the inclusion of programming in school, but that he has managed to ease their minds to some extent. One teacher believes she will benefit from collaboration with other teachers with programming competence.

The teachers feel they are able to teach programming to some extent, but want to feel more secure through experience. All three teachers perceive adapted teaching as a challenge in teaching programming, due to pupils’ varied prior knowledge and programming being a new field of study for both the teacher and the pupils. The teachers did not perceive assessment, motivation, or conveying programming knowledge as big challenges in their teaching of programming, but they also express that they would need more experience to say for sure. Two of the teachers feel their programming skill is sufficiently good for their teaching, one does not feel she is there yet, but believe she will get there with time. Two of the teachers perceive their programming skill as relatively low.

In addition to adapted teaching, the teachers experience technical issues and pupils' digital competence as a challenge in teaching programming.

The teachers report that the studies in programming has had a positive impact on their self-efficacy in teaching programming, but they still do not feel fully comfortable with teaching programming. One teacher report that the studies has had a positive impact on his attitudes towards programming in school.

## 6.2 Method

The method used in the supplementary interview study is the same as in the interview study seen in chapter 5.2. One notable change is that in the supplementary interview study the teachers were not asked about how the time after their studies had impacted their self-efficacy in teaching programming, since they at the time of the interviews were still not finished with the programming education. The same interview guide as in the main interview study was used in this study, with some minor changes. The changes were that questions that regarded the time after the end of the programming studies were removed. The interview guide used in the supplementary interview study can be found in appendix B. The only other aspects that are different from the main interview study are the participants, the recruitment method, and a small change to the analysis method.

### 6.2.1 Partcipants

**Table 6-1: Overview of interviewed teachers from the 2020 course, grade, teaching subjects, subjects they expect to use or are using programming in, and if they teach in general or vocational education.**

| **Participant**<br>Number-gender-subject-grade<br>HS: High school<br>SS: Secondary school | **Teaching grade** | **Subject(s)** | **Programming subject(s)** | **General/Vocational education** |
|---|---|---|---|---|
| 1-Male-Math-HS | High school | Math | Math | General |
| 2-Male-Math-SS | Secondary school | Math, natural science, programming | Math, natural science, programming | General |
| 3-Female-Math-HS | High school | Math, geoscience, physics | Math | General |

### 6.2.2 Recruitment

The researcher was given access to a Slack-channel (Slack Technologies, 2020) where all of the students from the 2020 applied programming course had access. There was posted a message in the Slack channel stating that all interested students could send a private message to the researcher on Slack. Interviews were then planned with each individual participant.

### 6.2.3 Analysis

The transcriptions from the supplementary interviews were analyzed in a similar way as in the main interview study (see chapter 5.2.8). The difference is that the supplementary interviews were coded directly into the categories already defined in the main interview study. The categories can be seen again below in table 6-2, note that the category "Impact of time after programming education" has been removed.

**Table 6-2: Categories from the analysis of supplementary interviews**

| Category | Description |
|---|---|
| Attitudes towards programming in school | Statements regarding teachers' attitudes towards programming in school |
| Teaching programming self-efficacy | Coding from the theoretical thematic analysis (See table 5-4) |
| Programming skill | Statements regarding the teachers' perceived programming skills |
| Impact of programming education | Statements regarding the impact of programming education on the teachers' programming skills and self-efficuacy |
| Digital competence | Statements regarding pupils' and teachers' digital competence |
| Course-specific reflection | Statements connected directly to the course IT6203 or IT6204 |
| Covid-19 | Statements regarding the impact of Covid-19 on the teachers' answers |

## 6.3 Results

This sub-chapter presents the results from the interview study on the teachers from the programming studies at NTNU in the class of 2019/2020. All quotes are indented for readability. All the quotes have also been translated from Norwegian to English by the researcher.

### 6.3.1 Attitudes towards programming in school

All three teachers are positive towards programming in school, but also express that they are somewhat skeptical to the way it is included into the curriculum. One of the teachers is skeptical towards if programming in math is the most beneficial way of including programming in school:

> I think it gives us some opportunities in the subject, so that's good. Whether it is good for these programming skills in itself is a little more uncertain … But, I'm not negative towards the inclusion of programming into math, I'm just a little unsure if it is the most beneficial way of doing it, regarding the skill set. (Teacher 1-Male-Math-HS)

One other teacher is very positive towards programming math, but more unsure about programming in natural science. He also states that the inclusion of programming in school will take time:

> I think it is absolutely great in math, I believe that it can add quite a lot to the teaching. I don't think it is done in one, two, three though, I think it's a project that's going to take a long time. But eventually, I think it will be very good. I am maybe a little more awaiting when it comes to natural science, on my own part. I think it is positive, but I have to wait a little longer to see how good I think it is. (Teacher 2-Male-Math-SS)

The third teacher also expressed that she is positive towards programming in school, but more skeptical towards how it is included. Especially towards if it should be included:

> I think it's good that programming is included, but I'm also unsure - I haven't quite landed yet on the way it is included, whether I think it's good or not. Programming should probably have been included as a separate subject in the previous curriculum, and integrated into subjects in this curriculum now. (Teacher 3-Female-Math-HS)

The third teacher is also worried that because programming is included into different subjects, many teachers that are going to teach it are not competent enough:

> But what I am a little afraid of is that there is a lack of competence, perhaps not only in primary school, but the entire 1-13 grades, and that it makes a very big difference in terms of who the pupils have as a teacher. And then the individual differences become so great, because of course, if you had your own subject called programming, then not any math teacher would have said that "I can teach that", or science teacher or social science teacher or whatever. (Teacher 3-Female-Math-HS)

One of the teachers experience that her colleagues are also positive towards programming to some extent, while another teacher expresses that some of his colleagues are more skeptical towards programming in school, but that he have tried to ease their minds by explaining that programming isn’t necessarily just to write complicated code, but that it’s much broader:

> Participant: And it makes it much more accessible too, when I've tried to tell colleagues who are skeptical towards this, tried to tell them, "well, this could be programming, and this could be programming", and so on. And then many of them of have lowered their shoulders a little.
> Interviewer: Do you feel that you can ease their minds, kind of?
> Participant: At least give some comfort to those who were very uncertain about how this was going to go, with the fact that it is not necessarily about acquiring such immensely great skills that one does not already possess. (Teacher 1-Male-Math-HS)

One of the teachers explain that she will have colleagues to work with when working on teaching programming, and state that she believes she will work well with them in regard to teaching programming:

> But as I said, I have some good colleagues who are taking programming education now, and I have worked well with them in the past, so I know I will work well with them in the future as well. So there are in particular a couple of the colleagues that I have, what should I say, “played good ball” with in relation to developing exercises in mathematics, where we have, in a way, done it together. So, I think, we already have the foundation from the beginning when starting with programming. (Teacher 3-Female-Math-HS)

### 6.3.2 Teachers’ self-efficacy towards teaching programming

All three teachers feel that they can teach programming to some extent, but that they do not feel very comfortable yet, and that they would like to become more secure in teaching programming. For example:

> Interviewer: Do you comfortable with teaching programming?
> Participant: No, not quite yet. It's a bit like that in a way, that it's not a traditional school

> subject if you know what I mean, that the teaching is quite different than, for example, a math lesson or an English lesson or, yes, anything really. So, I am not quite comfortable with it yet because I have yet to figure out how to teach in the best possible way, so hopefully I find out, but so far I am not quite so comfortable with it. (Teacher 2-Male-Math-SS)

#### 6.3.2.1 Adapted teaching

All three teachers perceive adapted education in programming as a challenge in some form. One of the teachers is not sure how well he can adapt his teaching to his pupils yet, and expressed that he will need experience before he can know. He perceives pupils' prior knowledge as a challenge as it can be difficult to find a suitable point of entry when introducing programming, such that all pupils are met on their level:

> That is probably to be able to meet the pupils with the learning material - find out where to start. Because I think you quickly can start either too high or too low. So if you start a course in programming by talking about data types and some different ways numbers are stored, and strings and stuff, then you risk losing a lot of pupils already there, and if you start too high with file processing and stuff like that, a program that will do some file processing, then it will also be on a too high level. So, you have to be able to adapt it so that it is just right. I think that can be difficult, it has to be done with trial and error. (Teacher 1-Male-Math-HS)

The second teacher also do not know how well he will be able to adapt his teaching of programming, due to programming being a new subject to both him and his pupils:

> It is difficult because, what to say, it is difficult to know, and getting to know what the pupils think is challenging in programming versus, for example, more traditional subjects like mathematics. When you are relatively new to a field of study yourself, you do not have the mindset of what can be experienced as difficult and what can be challenges, and what some pupils are struggling with and what others are not struggling with. (Teacher 2-Male-Math-SS)

The third teacher also perceives adapted teaching as a challenge due to the varying prior knowledge of the pupils:

> I see a challenge in what they can do and know from before. Because those who can do a lot of programming, let's say they've been playing with Scratch for 5-6 years, they know a lot of things about loops and all these things, and tests that they should do, but that doesn't necessarily mean they understand the math behind what we have to do with programming in math. So, I might lose them according to the mathematical understanding of what we're going to do in math, but many of those who, those who may have never touched any programming may have much better mathematical understanding, but they have no competence in programming. So, adapted teaching in the classroom, I think that can be challenging. (Teacher 3-Female-Math-HS)

All three teachers believe they can provide suitable challenges for more capable pupils in programming. One for example verbalized:

> In fact, I think so, because in a lot of cases we will be able to simply add additional functionality … so that a student who has better skills, or greater skill sets, may face such as additional challenges. (Teacher 1-Male-Math-HS)

#### 6.3.2.2 Assessment

All three teachers express that they feel they will be able to use different methods of assessment in programming, and that they believe they to some extent can assess their pupils in programming, but that they will need some more experience before they can say for sure. For example, one teacher answered when asked if he could use different strategies of assessment in programming:

> Yes. I would say that. Without that I have, like I said, tried so much myself yet. But I would believe that I can as far as I know. (Teacher 2-Male-Math-SS)

#### 6.3.2.3 Motivation

None of the three teachers brought up motivation as a challenge in teaching programming during the interviews, or that they felt insecure towards motivating their pupils in programming.

#### 6.3.2.4 Explaining and conveying programming knowledge

All three teachers believe that they would be able to explain and also come up with alternative explanations in basic programming, but that if they were to explain more complicated programming concepts and how to use them, they would have more trouble. For example, one teacher expressed when asked if he could come up with alternative explanations in programming if his pupils did not fully understand:

> Yes, I believe so. Obviously, you can get a question that you are a little unprepared for, so I don't know if I can cough up a good explanation then and there, but I think I would be able to give alternative explanations for a lot of the basic programming concepts, I believe so. Yes. I would say that. Without that I have, like I said, tried so much myself yet. But I would believe that I can as far as I know. (Teacher 2-Male-Math-SS)

When asked if they could answer more complicated or difficult questions from the more capable pupils in programming, all three teachers answered that they would be able to do this to some extent. They also explain that if they cannot answer the question, they would probably be able to find the answer later or with the pupil. For example:

> Yes, and if I can't answer it, then I can at least help them figure it out. (Teacher 2-Male-Math-SS)

One of the teachers also state that if she if has capable pupils, she can also use them as a resource in class:

> I used something like Lego Mindstorms with my pupils, and I had never played with that myself. Two of the students had been doing this for many years and they were very good at it, so I said "ok, you two are my assistants, you get to help all the other students with how Lego Mindstorms and the robots work". So, what should I say, I'm academically confident enough in the subjects of math and physics, so it's not a problem that I can't do everything in programming. (Teacher 3-Female-Math-HS)

#### 6.3.2.5 Developing teaching material

All three teachers express that they can develop good programming lesson plans and exercises for their pupils to some extent. One of the teachers feels confident in being able to develop lessons in math that contain programming, but not so confident in creating pure programming lessons:

> I feel I can do it well if I connect it to math somehow. But when it comes to making a more general programming lesson, which in a way does not have any other academic foundation, then think it would be a little more challenging. And I am not quite sure if I would have been able to do it in a good way right away. (Teacher 2-Male-Math-SS)

The two other teachers believe they will be able to create good lessons in programming, one of the them points out that there are many good existing teaching resources to adapt and use:

> Yes, I believe so. To be honest, there are so many ideas out there already. Whether you straight out copy a lesson plan that someone else has made, or if you use it as a foundation for your own lesson, I feel pretty confident that it will be possible to do to a good extent. (Teacher 1-Male-Math-HS)

### 6.3.3 Programming skill

Two of the teachers express that they feel their own programming skill is sufficient for teaching programming. For example, one of them expresses:

> Interviewer: In terms of programming skills, the skills you have now, do you feel they are good enough for your teaching of programming?
> Participant: Yes. I do. (Teacher 2-Male-Math-SS)

The third teacher expressed that she feels her programming skills are not good enough yet, but that they will be with more experience in doing programming:

> No, not yet. But I think they can be. I feel in many ways that they are not there yet, but the expectations of what the students will learn in the beginning in relation to the new curriculum are not very high … but in programming I do not have the same amount of training, which means that I may not be able to easily create my own exercises, but I think it will come with time. (Teacher 3-Female-Math-HS)

Two of the teachers perceive their programming skill as relatively low. One of the teacher stated:

> In terms of skills, when it comes to programming, I still have a way to go. I am on the level above beginner, perhaps that is the best description. On my way from beginner to - I don't remember what the next step is called - novice. (Teacher 1-Male-Math-HS)

All three teachers believe they are able to increase their programming skill on their own. For example, when asked if she felt she could increase her programming skills on her own or in collaboration with colleagues, one of the teachers stated:

> Yes, both on my own and in collaboration with colleagues, especially the latter. (Teacher 3-Female-Math-HS)

One of the teachers expressed that she thought it was a sudden steep learning curve in the first course, introductory programming:

> … the first exercises were really easy, and suddenly there a huge leap, it was not natural progression … (Teacher 3-Female-Math-HS)

### 6.3.4 Challenges in teaching programming

When asked what they perceive as the biggest challenges in teaching programming, the teachers mainly presented three challenges. The first is pupils' prior knowledge and thus adapted teaching. The second and third are pupils' digital competence and technical challenges, for example one teacher state it this way:

> Many of the pupils fail due to technical issues, so we rely more on giving them specific examples, and we also hope they discover these things themselves. We also point them out along the way, but we can't spend three weeks on how to type things in this computer program, to make them understand. And ten years ago, I had not exactly imagined that this would be a challenge. (Teacher 1-Male-Math-HS)

Technical issues is also one of the main challenges perceived by another teacher:

> I may stumble through it, but I still don't feel very secure. Some of the reason is that there are so many technical challenges that I see. (Teacher 3-Female-Math-HS)

### 6.3.5 Impact of programming education on self-efficacy in teaching programming

In general, the teachers report that the programming education has had a positive impact on their self-efficacy in teaching programming. Two of the teachers state that they feel they could not have taught programming before the programming education. For example:

> No, I could not have taught programming. I think I would have been able to fumble my way through something, with how programming is included in math now in mind. Or, no, I would not have managed to do it with confidence, and I would very much like to know that I master something before I teach anything, so I answer no. (Teacher 3-Female-Math-HS)

And both of these teachers also express that they believe they now can teach programming to some extent, but also that they do not feel very comfortable yet. For example:

> I don't know if I'm quite there yet, that I feel fully confident in being able to teach it now, but I am much more confident than before. Among other things, there are some didactic issues that have emerged that I had not thought of at all. (Teacher 1-Male-Math-HS)

The other teacher had already been teaching programming before the education, and also felt that he could teach programming before the studies. He also expressed that the programming education has had a positive impact on his capability in teaching programming, but not so much on his programming skill:

> I can do a bit more programming, but that's not where the main part of the development is. I feel I have gotten better yes, but mostly in the activities and exercises we have had in this study, then I feel I have become better at, what should I say, teaching programming, but I do not feel that I have learned so much more programming. But that wasn't the main goal either. (Teacher 2-Male-Math-SS)

There is also an indication that the programming education has had a positive impact on the teachers’ attitudes towards programming, for example:

> I've belonged to those who thought of programming quite specifically as coding. But I've expanded that view a bit. I look at it as - I think there are several things, and some things we already do, which can actually be called programming. You just have to be aware of it. (Teacher 1-Male-Math-HS)

## 6.4 Results summarized in the context of attitudes and self-efficacy

### 6.4.1 Teachers’ attitudes towards programming

The teachers from the programming education of 2019/2020 are also very positive towards programming in school, but seem more skeptical towards how programming is being included in the curriculum. One of the teachers is worried that since programming is included the way it is, there will be significant differences in how competent each programming teacher is, which again can have consequences for the pupils and their learning. One of the teachers experience that some of his colleagues are somehow skeptical towards programming, but that he also manages to ease their minds by explaining what programming can be. One of the teachers explain that she will have colleagues with programming competence to collaborate with, and that she believes this collaboration will work well.

### 6.4.2 Teachers’ self-efficacy towards teaching programming

The teachers from the programming education of 2019/2020 all express that they feel able to teach programming to some extent, but that they would also like to feel more secure in their teaching. In other words, they do not have a very high sense of self-efficacy in teaching programming, but not a low sense either.

All three teachers perceive adapted teaching and/or pupils' prior knowledge as a challenge in teaching programming. They express that they believe it will be challenging to adapt their teaching to all pupils. All teachers believe they can provide suitable challenges for more capable pupils. It is indicated that they have a moderately high sense of self-efficacy when it comes to adapted teaching, but also that they perceive this as one of the bigger challenges in teaching programming.

All three teachers believe that they to some extent can use different methods of assessment in programming, but that they will need more experience to find out for sure. It is indicated that they have a rather high sense of self-efficacy when it comes to assessment in programming.

All three teachers believe they can explain and come up with alternative explanations in basic programming, but they are more insecure when it comes to more complicated programming concepts. All three teachers also believe they to some extent would be able to answer more difficult questions from more capable pupils in programming, or find the answer later or with the pupil. One teacher also explain that she would be able to use more capable pupils as a resource in class. It is indicated that they have a quite high self-efficacy in conveying basic programming knowledge, and also towards finding answers to more difficult questions in programming as long as there is time for it.

All three teachers express they can make and adapt teaching material that includes programming. One of the teachers are however less confident in his capability in developing pure programming lessons. One of the teachers also points out that there is a vast amount of good programming teaching resources to use and adapt.

### 6.4.3 Programming skill

Two of the teachers state that they believe their programming skills are sufficient for their own teaching. The third teacher feels her skills are not good enough yet, but believe they will be with more time and experience, and by working together with other programming teachers. Two of the teachers perceive their programming skill as relatively low. One of these expressed that the first course, introductory programming, had a steep learning curve.

### 6.4.4 Challenges in teaching programming

The three main challenges presented by the three teachers is adapted teaching/pupils' varied prior knowledge, technical issues and pupils' digital competence. The technical issues both include that there can be many problems with using digital equipment and programs, as well as pupil's digital competence in form of lack of computer skills and experience with using computers.

### 6.4.5 Impact of programming education on self-efficacy in teaching programming

In general, the programming education seems to have had a positive impact on the teachers' self-efficacy in teaching programming. Two of the teachers express they could not have taught programming before the study, but that they to some extent feel they can teach programming in their respective subjects now. They also express that they still do not feel fully comfortable with teaching programming. The third teacher elaborate that even though he felt he could, and did teach programming before the programming education, he now feels more secure in the pedagogical and didactical aspects, but that his programming skills are not that much better. He also stated that he feels his programming skills are sufficient for teaching programming. One teacher also indicates that his attitude towards programming has been positively impacted by the programming education, as he now has a broader view on what programming actually can be.

# 7 Discussion

## 7.1 Chapter overview

The previous chapters explore teachers' attitudes and self-efficacy towards teaching programming through a study of reflection notes from a programming course for teachers, and two interview studies. In this chapter, the researcher explores the research questions through a discussion on the results and in light of relevant research.

## 7.2 Teachers' attitudes towards programming

### 7.2.1 Programming's place in school

The results from the study of the reflection notes and the interview studies indicate that the teachers with programming education are positive towards the inclusion of programming into the curriculum in the Norwegian school system, and that many of them perceive programming as something relevant for the pupils to learn and use. That programming can be relevant for the pupils to engage with is also found in the study by Hartell et al. (2019), but the teachers in the study by Hartell et al. (2019) also questioned the "why" behind teaching programming in Swedish primary school. This does not seem to be the case in this study, as the teachers in this study are almost exclusively positive towards programming in school, and several teachers express that they see many possibilities with the inclusion of programming, similar to that K-12 computer science teachers in Turkey find coding education beneficial (Kadirhan et al., 2018).

There are however worries on how programming is included into the curriculum. From the reflection notes and the supplementary interview study there is an indication that the teachers are not completely convinced that the inclusion of programming into math and natural science is the most suitable way of including programming. There are also some worries on that many teachers will not have the sufficient competence to teach programming. A result that is also found in several of the studies presented in chapter 2.4 (Kadirhan et al., 2018; Royal Society, 2017; Yadav et al., 2016). Like in these studies, the lack of competent teachers of programming seems to also be a worry among the teachers in this study. Seen in relation to the recommendation that a compulsory subject in technology and programming is created in Norwegian schools (Sanne et al., 2016), and the recommendation by Vahrenhold et al. (2017) that informatic subjects should only be taught by qualified teachers with formal education in informatics, one can ask the question of why programming is being included the way it is in the Norwegian educational system. The case is that very many Norwegian teachers in math, natural science, and music will have to teach programming, without formal education or training in programming. The teachers' worries on how programming is included in the curriculum found in this study seem to have some roots in reality, as they correspond with relevant reports and findings from other studies.

The teachers from the main interview study, that took the programming education one year ago, are however a little less worried on the way programming is included and perceive it as relevant to use programming interdisciplinary, but they also report that the inclusion of programming is not a quick fix, and that it will be a long process. There is an indication that teachers with programming education have positive attitudes towards the inclusion and use of programming in school, but that some question the way programming is included. There has also been resistance towards the inclusion of programming in math from Norwegian teachers earlier (Utdanningsdirektoratet, 2017), the teachers in this study are however more positive than negative towards the inclusion of programming in math. It is indicated that the teachers perceive it relevant to give programming a place in the curriculum, but there is more uncertainty towards where that place is.

### 7.2.2 Programming community and perceived collective attitude towards programming

How the teachers in this study perceive their colleagues' attitudes towards programming is very varied. Some report that they feel their school and colleagues are positive towards programming in school, but several teachers also report that they experience some form of resistance or little interest in programming from their colleagues. There is also an indication that teachers want colleagues to work with in programming, and that those who have someone to collaborate with find it useful. This is to some extent similar to the findings by Yadav et al. (2016), that computer science teachers often work without peers in their content area. There is reason to believe that it would also be beneficial with a focus on developing communities of practice for programming teachers in Norwegian school, both to support current teachers of programming as well as providing easy access to content knowledge for new teachers of programming.

Two of the teachers in the main interview study also mentioned gender differences in some form when it came to teaching programming. This is not explored in this study, but it can be seen in relation to other studies on gender differences in technology education, as some researchers have argued that the introduction of technology into the educational system were followed by new stereotypes and gender inequalities (Markauskaite, 2006), and that men and women have been socialized to think differently about technologies and their various uses (Campbell & Varnhagen, 2002; Huffman, Whetten, & Huffman, 2013).

## 7.3 Teachers' self-efficacy in teaching programming

### 7.3.1 Teachers' instructional self-efficacy

From the reflection notes, it is indicated that many of the teachers feel more secure and better equipped to teach programming. There is in an indication that their general self-efficacy towards teaching programming is relatively high, and this is also indicated in the results from the main interview study. The results from the supplementary interview study show that the teachers do not yet feel completely comfortable with teaching programming, but that they also believe they will get more comfortable with time and experience. The teachers from the main interview study took the programming education one year ago, thus having more time and experience, and this might be the reason why these teachers reported a higher sense of self-efficacy in regard to their ability to teach programming. From the reflection notes and the supplementary interview study, the teachers also believe they will get better with time and experience. That the teachers with programming education indicate a rather high sense of self-efficacy in teaching programming can be seen as a positive trend in relation to other studies from Sweden and UK where teachers showed lower grades of self-efficacy in teaching programming (Hartell et al., 2019; Mannila et al., 2018; YouGov, 2015). It is important to note that the teachers in these studies were not exclusively teachers with education in programming like in this study, but this can still be an indication of that programming education of in-service teachers can support teachers in increasing their self-efficacy in teaching programming.

From the reflection notes written by the teachers from the 2018-19 programming education at the end of their programming education, it is indicated that the teachers find adapted teaching as challenging, due to pupils' varied prior knowledge. This is also reported by the teachers in the supplementary interview study, who were also in the end of the programming education at the time of the interviews. Similar to this, US K-12 computer science teachers also reported in a study that it was difficult to meet all pupils' needs on an individual level (Yadav et al., 2016). However, only one of the teachers in the main interview study indicated a lower sense of self-efficacy in adapting her teaching to her pupils, the other teachers reported a high sense of self-efficacy when it came to adapted teaching. Since these teachers were also part of those who wrote the reflection notes, it can be seen as a possible indication that adapted teaching in practice is not such a challenge as reported in the reflection notes, and that time and experience has a positive effect on the teachers' self-efficacy towards adapted teaching in programming, but this is still not sufficient data to draw this conclusion.

The teachers' self-efficacy towards assessment in programming is more varied. Some teachers report in the reflection notes that they found assessment in programming challenging, this was also the case for four of the teachers in the main interview study. Some of the teachers that did not find it very challenging expressed that oral presentation was one of the better way of assessing their pupils in programming. Yadav et al. (2016) found that there was a need for quality assessment tools in computer science and coding education, there is an indication that this is also the case in this study. In the main interview study, the teachers seemed to have lowest self-efficacy when it came to assessment in programming. This was not the case in the supplementary interview study, but these teachers also reported that they were not sure if assessment would be a challenge, due to lack of experience. This could indicate that assessment in programming is a bigger challenge than the teachers without experience perceive.

In the reflection notes some of the teachers express that they would have liked more focus on what and how to teach programming in the education, and that the didactical aspects are challenging. This is also found in the main interview study. In relation to this, most of the teachers express they can develop and adapt varied teaching material in programming and seem to have a relatively high sense of self-efficacy towards developing and adapting teaching material. Some express that it can be difficult to find the most suitable teaching resources in the abundance of teaching resources in programming found online, a similar result is also found in the interview study of US K-12 computer science teachers (Yadav et al., 2016). Some teachers also express that they would like more teaching resources in programming, as is also indicated by the teachers in the study by Kadirhan et al. (2018). The results also indicate that the teachers have a relatively high sense of self-efficacy towards motivating their pupils in programming, but that it can be difficult to create exercises that engage the lesser capable pupils in programming, while also giving them a sense of mastery. It is also indicated that the teachers have a moderately high sense of self-efficacy in explaining and conveying programming knowledge, due to the teachers not feeling able to answer difficult programming questions. At the same time, the teachers felt they could find answers to difficult and complicated questions either later, or along with the pupils. That a part of the teachers still express a need for more focus on how and what to teach in programming, points to a need of focusing on specifying what is most beneficial to focus on in pupils' learning of programming, rather than how to develop lesson plans and conveying the programming knowledge.

### 7.3.2 Teachers' programming skill and their self-efficacy in teaching programming

The results indicate that the majority of the teachers feel their programming skill is sufficient for their teaching but also that many teachers perceive their programming skill as relatively low. Some teachers report that they do not feel completely comfortable with teaching programming, due to low content knowledge or programming skill, but the majority of the teachers do not report that their perceived low programming skill significantly impact their self-efficacy in teaching programming. Korkmaz (2013) argues that it is important for prospective teachers of programming to attain skills in programming, this is also found in this study, but there is also an indication that even though the teachers perceive their own programming skills as not very high, they still feel capable of teaching programming in their respective grades and teaching subjects. Many of the teachers also report that they feel capable of increasing their programming skill on their own or with colleagues, and that they will get better with experience. Studies have shown that teachers' programming self-efficacy can be improved with training in programming (Mazman & Altun, 2013; Yukselturk & Altiok, 2017), this study indicate that teachers with programming education believe they can improve their self-efficacy towards their programming skill also on their own and with colleagues.

### 7.3.3 Pupils' digital competence and technical issues

The teachers in the study perceived pupils' varied or low digital competence as a challenge in their teaching of programming. This is a challenge that can also be related to the teachers' self-efficacy in teaching programming, as it can be perceived as a constraint for how capable the teachers feel towards their teaching. A part of the teachers report that pupils have varied computer skills, and that some struggle with using computers in a good manner. One of the purposes of including programming in the Norwegian curriculum is to improve the pupils' digital competence (Sevik, 2015). As pupils' digital competence is perceived by some teachers as a challenge to their teaching of programming, this seems to be a relevant area of research in the years to come.

The teachers also perceived and experienced technical issues with equipment or computers as challenging and time consuming in their teaching of programming. It is not clear whether this is due to the quality of the technological resources, or the teachers' ability to use these resources. There is an indication that there is a need for addressing this challenge, for example in providing more reliable technological equipment for the teachers or finding out whether these teachers have sufficient computer skills to use the technological resources. A few of the teachers reported that they did not have access to suitable technological resources at their schools, but this was not a problem for most of the teachers.

## 7.4 Impact of programming education on teachers self-efficacy in teaching programming

The programming education seem to have had a positive impact on the teacher's self-efficacy in teaching programming. As seen in other studies, training in programming can increase teachers programming self-efficacy (Mazman & Altun, 2013; Yukselturk & Altiok, 2017), this study also indicate that teachers self-efficacy towards teaching programming can be increased with education in programming for teachers with a focus on teaching basic programming, as well as how programming can be applied in school. Most of the teachers in the interview study report that they did not feel capable of teaching programming before taking the programming education, but that they feel they are able to teach programming in their grades and teaching subjects after the programming education. This is seen as an indication that the education has impacted their self-efficacy towards teaching programming in a positive way. Teachers also report that their attitudes towards programming in school has been impacted positively by the programming education, as they see more possibilities and why programming is relevant in school.

An observation from the results is that many of the teachers reported that the introductory programming course had a very steep learning curve, and that many worked hard in this course. That programming can be hard to learn is no secret (Guzdial, 2015; Saeli, Perrenet, Jochems, & Zwaneveld, 2011), and seems to be the case for many of the teachers in this study as well. One of the teachers in the main interview study stated that her confidence in teaching programming went down during the introductory programming course. Most of the teachers were more content with the workload and relevance in the second course, applied programming, but two of the teachers from the main interview study expressed that they thought the introductory programming course was very good, and that the applied programming course was too easy and not very good. It seems to be difficult to have a programming education for teachers from different grades and subjects where one approach fits all. But as the programming education in this study has already undergone some changes, and it will probably also see more changes over the next years (Rouhani et al., 2019), it will be interesting to follow the development and results over the years.

## 7.5 Impact of time and experience on teachers self-efficacy in teaching programming

This study also explores how the time after the teachers took the programming education have affected their self-efficacy in teaching programming. The results indicate that the teachers have either increased sense of self-efficacy in teaching programming due to experience in teaching programming, or little to no change in their self-efficacy towards teaching programming. Several of the teachers report that their perceived programming skills lowers with time, due to not being used or maintained, but also state that they believe they can refresh their skills and knowledge in programming. Some of the teachers state that they would have liked more follow up exercises or a community of teachers where one could maintain one's programming skill after the programming education. This can also be seen in relation to that computer science teachers often work in isolation from peers in their content area (Yadav et al., 2016), and again argue for developing communities of practice for teachers of programming.

## 7.6 Implications of the research

There is an indication that teachers in this study and other studies are worried that there is a lack of competent teachers of programming. Some teachers are also worried towards how programming is included into the curriculum in Norway. The way programming is being included in the curriculum in Norway does not seem to follow the recommendations given by Sanne et al. (2016) as programming is being included into existing subjects, and not into a new technology subject. It does not follow the recommendations by Vahrenhold et al. (2017) as many teachers in Norway without programming competence will have to teach and use programming in their subjects. This thesis does not look into why programming is included into the curriculum in Norway the way it is, but this seems to be a relevant question to ask. There is reason to believe that it will be important to increase teachers programming competence in the years to come, and that there should be a significant focus from the Norwegian government and educational institutions with teachers education towards providing and giving teachers the necessary competence for teaching programming, if the inclusion of programming in school is to be successful. In relation to this, this study indicates that continuing education in programming can have a positive impact on teachers' self-efficacy in teaching programming as well as on their attitudes towards programming in school, and therefore that continuing education in programming can be a suitable way of preparing new teachers of programming for the challenges that lay ahead.

The results from the reflection notes and the main interview study indicate that the teachers find assessment in programming challenging. Many teachers also find learning programming hard. The researcher perceives these two aspects as relevant to focus on in further development of programming studies for teachers by educational institutions. There also seems to be a need for specifying what the content and learning objectives in teaching of programming should be for the programming education of pupils to be most beneficial, which is also relevant for educational institutions in the development of programming courses for teachers.

Some teachers express that they find collaboration with other programming teachers useful, and some express they would like teachers to work with in programming, both to increase and maintain their programming skill as well as share and discuss teaching of programming. Some teachers also experience that their colleagues are negative towards programming in school. There is an indication that there is a need for the development of communities of practice in the domain of teaching programming, especially locally at the teachers' own school, to support teachers of programming in "isolation" as well as providing easy access to content knowledge for new and inexperienced teachers of programming. Such communities of practice could potentially also help ease the negativity towards programming at some schools, as teachers might get more insight to the definition of programming and the possibilities that follows.

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

# 8 Conclusion

## 8.1 Chapter overview

This chapter first summarizes the research and presents the conclusions of the research in light of the research questions, as well as what further research the researcher perceives as relevant. Then the quality of the research is discussed.

## 8.2 Contribution

This master's thesis has explored the attitudes and self-efficacy of in-service teachers with programming education, with a focus on the inclusion of programming in the Norwegian educational system and teaching of programming. The research is based in the research question:

*How do in-service teachers with programming education perceive their attitudes and self-efficacy towards teaching programming, and the impact of programming education on their self-efficacy towards teaching programming?*

And explore the research question through the sub-questions:

**RQ1:** *What attitudes do in-service teachers with programming education have towards programming in school?*

**RQ2:** *How do in-service teachers with programming education perceive their self-efficacy in teaching programming?*

**RQ3:** *How do in-service teachers perceive that programming education has affected their self-efficacy in teaching programming?*

**RQ4:** *How do in-service teachers perceive the lasting effect of programming education in regard to their self-efficacy in teaching programming?*

The researcher has explored the research questions through the case of a continuing education program in programming for in-service teachers. Reflection notes on the students' (in-service teachers) reflections towards their learning and results from the programming education in 2018-19 have been analyzed. Interviews with ten of the teachers from the 2018-19 programming education have been conducted, transcribed and analyzed. Interviews with three of the teachers from the 2019-20 programming education have also been conducted, transcribed and analyzed.

The result and conclusion of the research sub-questions are presented below.

**RQ1:** *What attitudes do in-service teachers with programming education have towards programming in school?*

The teachers are very positive towards programming in school and perceive it as relevant for the pupils to learn, and to use interdisciplinary. Some teachers report that their attitudes have been positively impacted by the programming studies as they have received more insight into the possibilities of programming, as well as a broader view on the definition of programming. Some of the teachers are worried on how programming is included in the curriculum in Norway, and express it will be challenging for teachers without competence in programming to teach and use programming. A part of the teachers express that their colleagues are not as positive towards programming as themselves, and some also experience that colleagues are negative towards programming. The teachers are positive towards collaboration with other teachers of programming, and they want colleagues to work with in programming. There is an indication that the teachers perceive it as relevant to raise the programming competence of teachers in school.

**RQ2:** *How do in-service teachers with programming education perceive their self-efficacy in teaching programming?*

The teachers perceive their general self-efficacy towards teaching programming in their respective grades and teaching subjects as high. There is a small indication that the teachers do not have a high sense of self-efficacy towards adapted teaching in programming right after the end of the programming education, but that they have a higher sense of self-efficacy in adapted teaching one year after the programming education. This study indicate that the teachers have lowest self-efficacy towards using different methods of assessment in programming. It is indicated that assessment, as well as providing quality tools for assessment in programming, should be a focus in future education of programming teachers. The teachers also find what to teach in programming and how to teach programming challenging to some extent, but they also report a relatively high sense of self-efficacy towards conveying programming knowledge, motivating their pupils, and developing and adapting teaching material. It is indicated that the teachers have a lower sense of self-efficacy when it comes to specifically what to teach in programming. The results indicate that the teachers perceive their programming skill as relatively low, but that in most cases this does not significantly impact their self-efficacy in teaching programming.

**RQ3:** *How do in-service teachers perceive that programming education has affected their self-efficacy in teaching programming?*

The results indicate that the teachers in this study perceive that programming education has made a positive impact on their self-efficacy towards teaching programming, and that they feel more secure and comfortable in their teaching of programming. Some report that they are more positive towards programming in school after the programming education. Many of the teacher expressed that programming was hard to learn, and that the introductory programming course had a steep learning curve, but also that they learned much from the course. One teacher perceived that the introductory programming course had a negative impact on her self-efficacy in teaching programming due to it being hard and having a big workload. Most teachers perceived that the applied programming course had a positive impact on their self-efficacy in teaching programming, and that it was relevant to their teaching. Two teachers perceived that the applied programming course had very little impact on their self-efficacy in teaching programming due to it not being interesting and relevant enough or that it had a small workload, but that the introductory programming course had a very positive impact on their self-efficacy in teaching programming.

**RQ4:** *How do in-service teachers perceive the lasting effect of programming education in regard to their self-efficacy in teaching programming?*

The results indicate that the teachers have either an increased sense of self-efficacy in teaching programming due to experience in teaching programming, or little to no change in their self-efficacy towards teaching programming due to little experience in teaching programming in the past year. Several of the teachers report that their perceived programming skills lowers with time, due to not being used or maintained, but also state that they believe they can refresh their skills and knowledge in programming. Some of the teachers express that they would have liked more follow up exercises or a community of teachers where one could maintain and increase one's programming skill after the programming education.

### 8.2.1 Further work and research

This study lurks in the surface of the research area of programming in school and teacher self-efficacy in teaching programming, which is also a rather uncharted area. The results and discussion give some pinpoints to what can be relevant to explore in further work and research.

The results indicate that the teachers find assessment in programming challenging, but that oral presentation is one of the better methods of assessment in programming. Assessment in programming should be explored in further research, as well as the development of quality assessment tools in programming education.

Many of the teachers experience technical issues as a challenge towards their teaching of programming. It is not clear whether this is due to the teachers' digital competence, or if there is a need for better quality technological resources. This could also be an interesting area of research, in addition to how to support teachers in developing their digital competence. Some of the teachers also express that pupils have low skills when it comes to using computers in a good and appropriate way. The researcher also perceives it as relevant to explore how the inclusion of programming affect the pupils' computer skills and digital competence.

As the inclusion of programming in the curriculum in Norway and other countries is a relatively new addition in the curriculum, there is also a need for more research on the phenomenon of programming in school in general. That some teachers in the study are worried on how programming is included, and that some experience negativity from colleagues towards programming is something that can also be relevant in further work. The researcher also perceives it as relevant to use self-efficacy theory in further research on teaching programming in school. The researcher also perceives it as relevant with research on developing a quality self-efficacy measurement instrument in the domain of teaching programming.

## 8.3 Quality of the research

This study looks into a relatively new area in the Norwegian educational system with the recent inclusion of programming into the curriculum, and the continuing education in programming of in-service teachers.

It is desirable to achieve good validity and reliability when working with research. Reliability refers to how reliable the results of the study are (Kvale et al., 2015), for example, two different transcriptions of the same interview can give different meanings and interpretations of statements depending on how it is transcribed. A way of getting better reliability in transcriptions is by having the same person do both the interviews and the transcriptions (Easton, McComish, & Greenberg, 2000), as done in this study. It is also impossible for the researcher to completely put aside all his opinions and attitudes while working on the research. Especially towards the analysis of the reflection notes and transcription is this relevant, as another researcher would maybe interpret the data in other ways than the researcher has in this study, and it is therefore difficult to say that these results can be generalized. This has been addressed by trying to thoroughly describe the method used in the analysis, and how the results are found. The results have also been compared and discussed in light relevant and similar research, which can enhance the rigor of the research if perceived as theory triangulation (Robson & McCartan, 2016).

Validity of research refers to whether the research is accurate, or in what grade we can say that the results are valid for what is being researched (Robson & McCartan, 2016). This research has a strength in that it is using multiple data sources, both the interviews and the reflection notes. The reflection notes give an indication of the opinions and attitudes of nearly all the teachers from the programming education, and the interviews provide a deeper insight into the teachers views and experiences. There is however an important issue with the reflection notes, as they are part of the final delivery and assessment in the programming education. The teachers might have been affected by the fact that the reflection notes were part of their assessment and final delivery in the programming education, and therefore they may not have given their exact or true opinions in the reflection notes. This was addressed by analyzing all the reflection notes available, and not only a sample of them, to be able to find the most common indications from the data. The interviews do not suffer from the same weakness, and that the results from the interviews point in the same direction as the results from the reflection notes can be argued to also strengthen the validity of the research.

The supplementary interview study gives an indication of how the results transfer from the 2018-19 students, to the 2019-20 students in the continuing education program. A weakness in relation to this, is that there have been made some changes in the continuing education program which are not taken into account in this research. The interviews with the 2019-20 students can therefore only be seen as a small strength towards the validity of the result. But the results from these interviews are also quite consistent with both the results from the reflection notes and the results from the main interviews.

As the researcher is a future informatics teacher, as well as having worked as a teaching assistant in the 2019 applied programming course, the aspect of bias has also been reflected upon. For example, some of the teachers in the main interview study might have received help or guidance from the researcher during his work as a teaching assistant. And the teachers may would have answered differently if the researcher was someone they did not have any relation to. The researcher may also have some opinion on the programming education himself after working as a learning assistant. As a future informatics teacher himself, the researcher also has his own opinions towards programming in school. Therefore, it has been important for the researcher to be aware of these opinions in the research process, to be able to conduct the research as objective as possible.

As the new curriculum in Norway is currently being implemented, as well as the continuing education program only being around two years old, the results from this study would most likely have been different if the study was done 5-10 years from now. The results from this study must be seen in context of the time it was done in, and this is also part of the purpose of the research; to give a pin-point of how well prepared we are towards the inclusion of programming in the Norwegian school system, and suggest what we can do to support new teachers of programming through continuing education in the years to come.

### 8.3.1 The implications of COVID-19 on the quality of the research

In context of the time the research was done (spring 2020), the situation with the pandemic and COVID-19 was also taken into account in the research process. The interviews with the teachers were conducted in a special time for both students and teachers in Norway, since schools and universities were closed at the time of the interviews. This was addressed in the interviews by asking whether the teachers believed their answers could have been impacted by the situation with COVID-19 in any way, and if relevant, in what way.

Five of the teachers in the main interview study believed that their answered could have been impacted by the COVID-19 situation. Two of the teachers reported that they had found new and different methods in their teaching due to the situation, and that this may had affected their answers somehow. For example, one stated:

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

One of the teachers reported that her answers might had been affected due to being insecure in how to continue her teaching of programming in the situation:

> Nothing else than what I have said around it that I am actually somewhat in doubt about how to continue teaching the subject right now. I'm a little unsure how to solve it (Teacher 5-Female-English-SS)

One teacher expressed that her answers might have been affected because it was a long time since she had seen her colleagues and discussed subject content with them, and that it was a long time since she had been teaching elective programming:

> Yes, it has been a long time since I have seen my colleagues, long since I have discussed subjects with them, so there are such things. It's been a long time since I've had elective subjects, since we haven't had electives yet, we are waiting a bit before doing it. After all, it can quickly become a full day of programming if we are to recover the lost. So yes, yes, such things may have affected it after all. (Teacher 8-Female-Math-SS)

One teacher worried that other teachers will be less prepared for the inclusion of programming into the curriculum when the next school semester comes, because some courses in programming arranged by the county council have been not prioritized due to COVID-19:

> The only thing is that I think it makes those of my colleagues who – there has been a little attempt by the county council at preparing all teachers for the programming that is coming through training, and so the only thing is that I think that there have probably been spent less time on this, and they are perhaps a little less prepared for the fall semester than they could have been. And that's a Corona thing. Or maybe it’s not, I’ve kind of been thinking about this all year. (Teacher 9-Female-Math-HS)

Of the three teachers in the supplementary interview study, one of them report that his answers may have been affected by the situation. He explains that he maybe feels less confident as a teacher due to not being able to give his pupils optimal teaching:

> Yes, I think that may have influenced the answers. I think if we had done this interview in a parallel universe without corona I would probably have responded differently … I am a little uncertain, but I think it has influenced, because I am sitting with so many fresh impressions of the teaching that I have been doing for 2-3 weeks now, which at least for me and certainly thousands of others, feels very suboptimal, and you feel like you have given lots of young pupils lots of not top teaching for 2-3 weeks now, so you might have a little lower self-esteem as a teacher than one would normally have. So I think I am a little affected by that. (Teacher 2-Male-Math-HS)

These answers indicate that the teachers may would have answered differently if the situation was normal, and this can therefore also somehow have affected the validity of the research. How it is affected is difficult to say anything about, but it can be an important aspect to have in mind either way when addressing the quality of the research. There was also planned to conduct a survey with the students from the 2018-19 continuing education program in programming, but this was not done due to time-issues that partly or indirectly was caused by COVID-19. In retrospective, the researcher believes that a survey that uses a teacher self-efficacy scale would have been a strength to the quality of the research, as it would have also given some quantitative data directly connected to the teachers' self-efficacy. The researcher believes this would have been more suitable than for example the supplementary interview study. The researcher would recommend taking this into account in similar research in the future.

# 9 References

Armor, D. (1976). Analysis of the school preferred reading program in selected Los Angeles minority schools.

Askar, P., & Umay, A. (2001). *Preservice elementary mathematics teachers' computer self-efficacy, attitudes towards computers, and their perceptions of computer-enriched learning environments.* Paper presented at the Society for Information Technology & Teacher Education International Conference.

Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioral change. *Psychological review, 84*(2), 191.

Bandura, A. (1986). *Social foundations of thought and action : a social cognitive theory*. Englewood Cliffs, N.J: Prentice-Hall.

Bandura, A. (1997). *Self-efficacy : the exercise of control*. New York: Freeman.

Bandura, A. (2006). Guide for constructing self-efficacy scales. *Self-efficacy beliefs of adolescents, 5*(1), 307-337.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology, 3*(2), 77-101.

Brouwers, A., Evers, W. J., & Tomic, W. (2001). Self-efficacy in eliciting social support and burnout among secondary-school teachers. *Journal of applied social psychology, 31*(7), 1474-1491.

Campbell, K., & Varnhagen, S. (2002). When Faculty Use Instructional Technologies: Using Clark's Delivery Model To Understand Gender Differences. *Canadian Journal of Higher Education, 32*(1), 31-56.

Easton, K. L., McComish, J. F., & Greenberg, R. (2000). Avoiding Common Pitfalls in Qualitative Data Collection and Transcription. *Qualitative Health Research, 10*(5), 703-707. doi:10.1177/104973200129118651

Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2011). *How to design and evaluate research in education*: New York: McGraw-Hill Humanities/Social Sciences/Languages.

Gibson, S., & Dembo, M. H. (1984). Teacher efficacy: A construct validation. *Journal of educational psychology, 76*(4), 569.

Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics, 8*(6), 1-165.

Hartell, E., Doyle, A., & Gumaelius, L. (2019). *Teachers' attitudes towards teaching programming in Swedish Technology education*.

Huffman, A. H., Whetten, J., & Huffman, W. H. (2013). Using technology in higher education: The influence of gender roles on technology self-efficacy. *Computers in Human Behavior, 29*(4), 1779-1786.

Kadirhan, Z., Gül, A., & Battal, A. (2018). Self-Efficacy to Teach Coding in K-12 Education. In *Self-Efficacy in Instructional Technology Contexts* (pp. 205-226): Springer.

Klassen, R. M., Tze, V. M., Betts, S. M., & Gordon, K. A. (2011). Teacher efficacy research 1998–2009: Signs of progress or unfulfilled promise? *Educational psychology review, 23*(1), 21-43.

Korkmaz, O. (2013). Prospective CITE Teachers' Self-efficacy Perceptions on Programming. *Procedia-Social and Behavioral Sciences, 83*, 639-643.

Kvale, S., Brinkmann, S., Anderssen, T. M., & Rygge, J. (2015). *Det kvalitative forskningsintervju* (3. utg., 2. oppl. ed.). Oslo: Gyldendal akademisk.

Lær Kidsa Koding. (2020). Om Lær Kidsa Koding. Retrieved from https://www.kidsakoder.no/om-lkk/

Malinen, O.-P., Savolainen, H., Engelbrecht, P., Xu, J., Nel, M., Nel, N., & Tlale, D. (2013). Exploring teacher self-efficacy for inclusive practices in three diverse countries. *Teaching and teacher education, 33*, 34-44.

Mannila, L., Nordén, L.-Å., & Pears, A. (2018). *Digital competence, teacher self-efficacy and training needs.* Paper presented at the Proceedings of the 2018 ACM Conference on International Computing Education Research.

Markauskaite, L. (2006). Gender issues in preservice teachers' training: ICT literacy and online learning. *Australasian Journal of Educational Technology, 22*(1).

Mazman, S., & Altun, A. (2013). The effect of introductory to programming course on programming self-efficacy of CEIT students. *Journal of Instructional Technologies & Teacher Education, 2*(3), 24-29.

NESH. (2016). Forskningsetiske retningslinjer for samfunnsvitenskap, humaniora, juss og teologi. Retrieved from https://www.etikkom.no/globalassets/documents/publikasjoner-som-pdf/60125_fek_retningslinjer_nesh_digital.pdf

NSD. (2020). Om NSD. Retrieved from https://nsd.no/nsd/english/index.html

NTNU. (2020a). IT6203 - Grunnleggende programmering for lærere. Retrieved from https://www.ntnu.no/studier/emner/IT6203/#tab=omEmnet

NTNU. (2020b). IT6204 - Anvendt programmering for lærere. Retrieved from https://www.ntnu.no/studier/emner/IT6204#tab=omEmnet

NTNU. (2020c). Natural Science with Teacher Education. Retrieved from https://www.ntnu.edu/studies/mlreal

NTNU. (2020d). Programmering 8. - 13. trinn. Retrieved from https://www.ntnu.no/videre/programmering

Olstad, H. A., & Rouhani, M. (2019). *Reflection on How to Write the Learning Outcomes for an Online Programming Course for Teachers*, Cham.

Papert, S. (1980). Mindstorms. *New York: Basic Books, 607*.

Postholm, M. (2005). Kvalitativ metode [Qualitative methodology]. *Oslo, Norway: Universitetsforlaget*.

QSR International. (2018). NVivo 12.

Ramalingam, V., & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research, 19*(4), 367-381.

Riggs, I. M., & Enochs, L. G. (1990). Toward the development of an elementary teacher's science teaching efficacy belief instrument. *Science Education, 74*(6), 625-637.

Robson, C., & McCartan, K. (2016). *Real world research*: John Wiley & Sons.

Rotter, J. B. (1966). Generalized expectancies for internal versus external control of reinforcement. *Psychological monographs: General and applied, 80*(1), 1.

Rouhani, M., Divitini, M., Vujosevic, V., Stai, S., & Olstad, H. A. (2019). *Design of a programming course for teachers supporting flexible learning trajectories*. Paper presented at the Proceedings of the 8th Computer Science Education Research Conference, Larnaca, Cyprus. https://doi.org/10.1145/3375258.3375263

Royal Society. (2017). After the reboot: Computing education in UK schools. *Policy Report*.

Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in Secondary school: A pedagogical content knowledge perspective. *Informatics in Education, 10*(1), 73-88.

Sang, G., Valcke, M., Van Braak, J., & Tondeur, J. (2010). Student teachers' thinking processes and ICT integration: Predictors of prospective teaching behaviors with educational technology. *Computers & Education, 54*(1), 103-112.

Sanne, A., Berge, O., Bungum, B., Jørgensen, E. C., Kluge, A., Kristensen, T. E., . . . Voll, L. O. (2016). Teknologi og programmering for alle-En faggjennomgang med forslag til endringer i grunnopplæringen-august 2016.

Sevik, K. (2015). Programmering i skolen. *Notat fra Senter for IKT i utdanningen*. Retrieved from https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf

Skaalvik, E. M., & Skaalvik, S. (2007). Dimensions of teacher self-efficacy and relations with strain factors, perceived collective teacher efficacy, and teacher burnout. *Journal of educational psychology, 99*(3), 611.

Slack Technologies. (2020). Slack. Retrieved from https://slack.com/intl/en-no/

Strauss, A. L., & Corbin, J. M. (1998). *Basics of qualitative research : techniques and procedures for developing grounded theory* (2nd ed. ed.). Thousand Oaks, Calif: Sage.

Tschannen-Moran, M., & Hoy, A. W. (2001). Teacher efficacy: Capturing an elusive construct. *Teaching and teacher education, 17*(7), 783-805.

Tschannen-Moran, M., & Johnson, D. (2011). Exploring literacy teachers' self-efficacy beliefs: Potential sources at play. *Teaching and teacher education, 27*(4), 751-761.

Uninett. (2020). Zoom og personvern. Retrieved from https://www.uninett.no/en/node/2084#gdpr-og-personvern

Utdanningsdirektoratet. (2017). Fagfornyelsen - matematikk. Retrieved from https://udirbloggen.no/matematikk/

Utdanningsdirektoratet. (2019). Algoritmisk tenkning. Retrieved from https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/

Utdanningsdirektoratet. (2020). Nye læreplaner - grunnskolen og gjennomgående fag vgo. Retrieved from https://www.udir.no/laring-og-trivsel/lareplanverket/nye-lareplaner-i-grunnskolen-og-gjennomgaende-fag-vgo

Vahrenhold, J., Nardelli, E., Pereira, C., Berry, G., Caspersen, M., Gal-Ezer, J., . . . Westermeier, M. (2017). Informatics education in Europe: are we all in the same boat. In: ACM/Informatics Europe, NY.

Wing, J. M. (2011). Research notebook: Computational thinking—What and why. *The Link Magazine*, 20-23.

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education, 26*(4), 235-254.

YouGov. (2015). TES & Nesta Computing Curriculum. Fieldwork 06/05/2014 - 16/05/2014 (2015).

Yukselturk, E., & Altiok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology, 48*(3), 789-801.

Zee, M., & Koomen, H. M. Y. (2016). Teacher Self-Efficacy and Its Effects on Classroom Processes, Student Academic Adjustment, and Teacher Well-Being:A Synthesis of 40 Years of Research. *Review of Educational Research, 86*(4), 981-1015. doi:10.3102/0034654315626801

Zoom Video Communications Inc. (2020). Zoom (Version 4.6.10). Retrieved from https://zoom.us/

# Appendices

# Appendix A: Interview guide used in the interviews with the 2018-19 students

1. Oppvarmingsspørsmål og informasjon om intervjuet
   (Hvordan går det?)

### Bakgrunnsinformasjon

2. Spørsmål om undervisningsfag og trinn
3. Hvilke fag underviser du i?
4. Hvilke trinn underviser du på?
5. Hvilke fag forventer du å undervise/bruke programmering i?

### Holdninger til programmering i skolen

6. Hva tenker du om at programmering nå er inkludert i læreplan?
7. Hva med (evt) i ditt/dine fag?
8. Har din holdning endret seg siden studiene i programmering - hvordan?
9. (Føler du at kollegene/ledelsen din deler denne holdningen?)
10. Hvordan har din interesse for programmering endret/utviklet seg siden studiene?

### Instructional self-efficacy

11. Følte du at du kunne undervise programmering før studiene?
12. Følte du at du kunne undervise programmering etter studiene?
    a. Hva var utfordrende?
13. Føler du deg mer/mindre kapabel/rustet nå enn etter studiene?
    a. Hva er utfordringene nå?
14. Føler du at du kan undervise i programmering?
    a. Føler du at du kan bruke ulike metoder for å vurdere elevene i programmering? (Hensiktsmessige?)
    b. Føler du at du kan vurdere hvor mye elevene har forstått av din programmeringsundervisning?
    c. Føler du at du kan tilpasse programmeringsundervisningen til dine elever?
    d. Føler du at du kan motivere elever med ulike behov?
    e. Føler du at du kan komme med alternative forklaringer om programmeringskonsept/kunnskap dersom elevene dine ikke forstår?
    f. Føler du at du kan lage gode undervisningsopplegg og oppgaver til dine elever i programmering?
    g. Føler du at du kan lage varierte undervisningsopplegg i programmering?
    h. Føler du at du kan svare på vanskelige eller mer kompliserte programmeringsspørsmål?
    i. Føler du at du kan tilby tilpassede utfordringer til elever som presterer høyt i programmering?
15. Kan du sette ord på hvordan studiene i programmering påvirket din følelse av mestringsevne i programmeringsundervisning?
16. Hvordan har tiden etter studiene påvirket dine evner i programmeringsundervisning?
    a. Føler du at din evne til å undervise programmering har endret seg siden du ble ferdig med studiene?

   b. På hvilken måte – til det bedre/dårligere?
   c. Utfordrer du deg selv mer/mindre nå?
17. Føler du deg komfortabel med å undervise programmering?
18. Hva ser du på som de største utfordringene i programmeringsundervisning?
   a. Føler du at kan du løse de utfordringene (evt hvorfor ikke)?

**Programming skill**

19. Føler du at du har god nok programmeringskompetanse for å undervise i programmering / i din undervisning av programmering?
20. Kan du utvikle din egen programmeringskompetanse?
   a. Evt hvordan gjør du dette?

**Course specific**

21. Er du fornøyd med videreutdanningen i programmering generelt?
22. Anvendt programmering faget
   a. måten det er lagt opp?
   b. Innholdet?
   c. Egenlæring?
   d. Fleksibiliteten?
   e. Oppfølging?
   f. Forbedringspotensiale?
23. Grunnleggende programmering faget
   a. (samme spørsmål som over utenom d.)

**Korona**

24. Kan denne situasjonen ha påvirket svarene du har gitt i intervjuet i dag?
   a. På hvilken måte?

*Oppsummeringsspørsmål og avslutning av intervju*

# Appendix B: Interview guide used in the interviews with the 2019-20 students

1. Oppvarmingsspørsmål og informasjon om intervjuet
   (Hvordan går det?)

**Bakgrunnsinformasjon**

2. Spørsmål om undervisningsfag og trinn
3. Hvilke fag underviser du i?
4. Hvilke trinn underviser du på?
5. Hvilke fag forventer du å undervise/bruke programmering i?

**Holdninger til programmering i skolen**

6. Hva tenker du om at programmering nå er inkludert i læreplan?
7. Hva med (evt) i ditt/dine fag?
8. (Føler du at kollegene/ledelsen din deler denne holdningen?)
9. Har dine holdnigner blitt påvirket av studiene?

**Instructional self-efficacy**

10. Følte du at du kunne undervise programmering før studiene?
11. Føler du at du kan undervise programmering nå?
    a. Hva er utfordringene nå?
12. Føler du at du kan undervise i programmering?
    a. Føler du at du kan bruke ulike metoder for å vurdere elevene i programmering? (Hensiktsmessige?)
    b. Føler du at du kan vurdere hvor mye elevene har forstått av din programmeringsundervisning?
    c. Føler du at du kan tilpasse programmeringsundervisningen til dine elever?
    d. Føler du at du kan motivere elever med ulike behov?
    e. Føler du at du kan komme med alternative forklaringer om programmeringskonsept/kunnskap dersom elevene dine ikke forstår?
    f. Føler du at du kan lage gode undervisningsopplegg og oppgaver til dine elever i programmering?
    g. Føler du at du kan lage varierte undervisningsopplegg i programmering?
    h. Føler du at du kan svare på vanskelige eller mer kompliserte programmeringsspørsmål?
    i. Føler du at du kan tilby tilpassede utfordringer til elever som presterer høyt i programmering?
13. Kan du sette ord på hvordan studiene i programmering påvirket din følelse av mestringsevne i programmeringsundervisning?
14. Føler du deg komfortabel med å skulle undervise programmering?
15. Hva ser du på som de største utfordringene i programmeringsundervisning?
    a. Føler du at kan du løse de utfordringene (evt hvorfor ikke)?

**Programming skill**

16. Føler du at du har god nok programmeringskompetanse for å undervise i programmering / i din undervisning av programmering?
17. Kan du utvikle din egen programmeringskompetanse?
    a. Evt hvordan gjør du dette?

**Course specific**

18. Er du fornøyd med videreutdanningen i programmering generelt?
19. Anvendt programmering faget
    a. måten det er lagt opp?
    b. Innholdet?
    c. Egenlæring?
    d. Fleksibiliteten?
    e. Oppfølging?
    f. Forbedringspotensiale?
20. Grunnleggende programmering faget
    a. (samme spørsmål som over utenom d.)

**Korona**

21. Kan denne situasjonen ha påvirket svarene du har gitt i intervjuet i dag?
    a. På hvilken måte?

*Oppsummeringsspørsmål og avslutning av intervju*

## Appendix C: Consent form

*Some elements have been anonymized due to privacy*

# *Vil du delta i forskningsprosjektet «Teachers' self-efficacy beliefs in programming»?*

**Introduksjon**
Jeg er student på lektorprogrammet i realfag ved NTNU og arbeider med min masteroppgave om programmeringslæreres følelse av mestringsevne i programmeringsundervisning. I dette skrivet gir jeg deg informasjon om målene for prosjektet og hva deltakelse vil innebære.

**Formål**
Formålet med prosjektet er å kartlegge programmeringslæreres følelse av mestringsevne i programmeringsundervisning for å kunne gi en indikasjon på hva utdanningen av programmeringslærere bør fokusere på. Prosjektet skal samle informasjon fra lærere i programmering og analysere opp mot relevant litteratur for å kunne lage et støtteverk for hva som er hensiktsmessig å fokusere videre på i utdanningen av programmeringslærere.

Programmering har fått en større plass i skolen, og lærere i programmering står overfor en rekke utfordringer: Læreplanen er i utvikling, faget endrer seg raskt, og det trengs mer kompetanse i programmeringsundervisning. Flere lærere vil trenge utdanning og kursing i programmering for å inkludere dette på en god måte i sin undervisning. Problemstillingen som blir undersøkt er:

Hvilke aspekter ved programmeringsundervisning mener videreutdannede lærere i programmering det er viktig å fokusere på i videreutviklingen av læreres programmeringsutdanning?

Dette undersøkes ved å se på læreres følelse av mestringsevne i programmering - hva føler de at de kan mestre i programmeringsundervisning etter utdanning og hva føler de at de ikke har nok kompetanse/evner til å klare.

Resultatene vil bli brukt i masteroppgaven, og ingen andre formål.

**Hvem er ansvarlig for forskningsprosjektet?**
NTNU, Institutt for datateknologi og informatikk [anonymisert] er ansvarlig for prosjektet.

**Hvorfor får du spørsmål om å delta?**
I prosjektet ønsker jeg å samle informasjon fra lærere som har tatt og tar videreutdanning i programmering. Dette er også grunnen til at du blir forespurt om å delta i dette prosjektet. Antallet personer som får denne henvendelsen er mellom 150-200 personer.

**Hva innebærer det for deg å delta?**
Hvis du velger å delta i prosjektet innebærer det at du deltar i et intervju. Det vil ta deg ca 40 minutter. Intervjuet vil omhandle din følelse av mestringsevne i ulike aspekter ved

programmeringsundervisning, samt dine tanker om hvordan utdanningen av programmeringslærere kan forbedres. I intervjuet vil det bli tatt lydopptak som vil bli transkribert og senere slettet.

**Det er frivillig å delta**

Det er frivillig å delta i prosjektet. Hvis du velger å delta, kan du når som helst trekke samtykket tilbake uten å oppgi noen grunn. Alle dine personopplysninger vil da bli slettet. Det vil ikke ha noen negative konsekvenser for deg hvis du ikke vil delta eller senere velger å trekke deg.

**Ditt personvern – hvordan vi oppbevarer og bruker dine opplysninger**

Vi vil bare bruke opplysningene om deg til formålene vi har fortalt om i dette skrivet. Vi behandler opplysningene konfidensielt og i samsvar med personvernregelverket.

- Kun jeg [anonymisert] og prosjektansvarlig vil ha tilgang til opplysningene.
- Lydopptak vil lagres på en kryptert minnepinne og låses inne i et skap når det ikke transkriberes. Navn vil bli anonymisert ved at de erstattes med en kode som lagres på en egen navneliste adskilt fra øvrige data.
- Deltakere vil ikke kunne gjenkjennes i publikasjon.

**Hva skjer med opplysningene dine når vi avslutter forskningsprosjektet?**

Prosjektet skal etter planen avsluttes 02.06.2020. Personopplysninger og opptak blir permanent slettet ved prosjektslutt.

**Dine rettigheter**

Så lenge du kan identifiseres i datamaterialet, har du rett til:

- innsyn i hvilke personopplysninger som er registrert om deg, og å få utlevert en kopi av opplysningene,
- å få rettet personopplysninger om deg,
- å få slettet personopplysninger om deg, og
- å sende klage til Datatilsynet om behandlingen av dine personopplysninger.

**Hva gir oss rett til å behandle personopplysninger om deg?**

Vi behandler opplysninger om deg basert på ditt samtykke.

På oppdrag fra NTNU har NSD – Norsk senter for forskningsdata AS vurdert at behandlingen av personopplysninger i dette prosjektet er i samsvar med personvernregelverket.

**Hvor kan jeg finne ut mer?**

Hvis du har spørsmål til studien, eller ønsker å benytte deg av dine rettigheter, ta kontakt med:

- Institutt for datateknologi og informatikk - NTNU, ved [anonymisert] Vårt personvernombud: [anonymisert]

Hvis du har spørsmål knyttet til NSD sin vurdering av prosjektet, kan du ta kontakt med:

- NSD – Norsk senter for forskningsdata AS på epost (personverntjenester@nsd.no) eller på telefon: 55 58 21 17.

Med vennlig hilsen

| | |
|---|---|
| Prosjektansvarlig | Student |
| [anonymisert] | [anonymisert] |

-------------------------------------------------------------------------------------------------------------

## Samtykkeerklæring

Jeg har mottatt og forstått informasjon om prosjektet *Teachers' self-efficacy beliefs in programming*, og har fått anledning til å stille spørsmål. Jeg samtykker til:

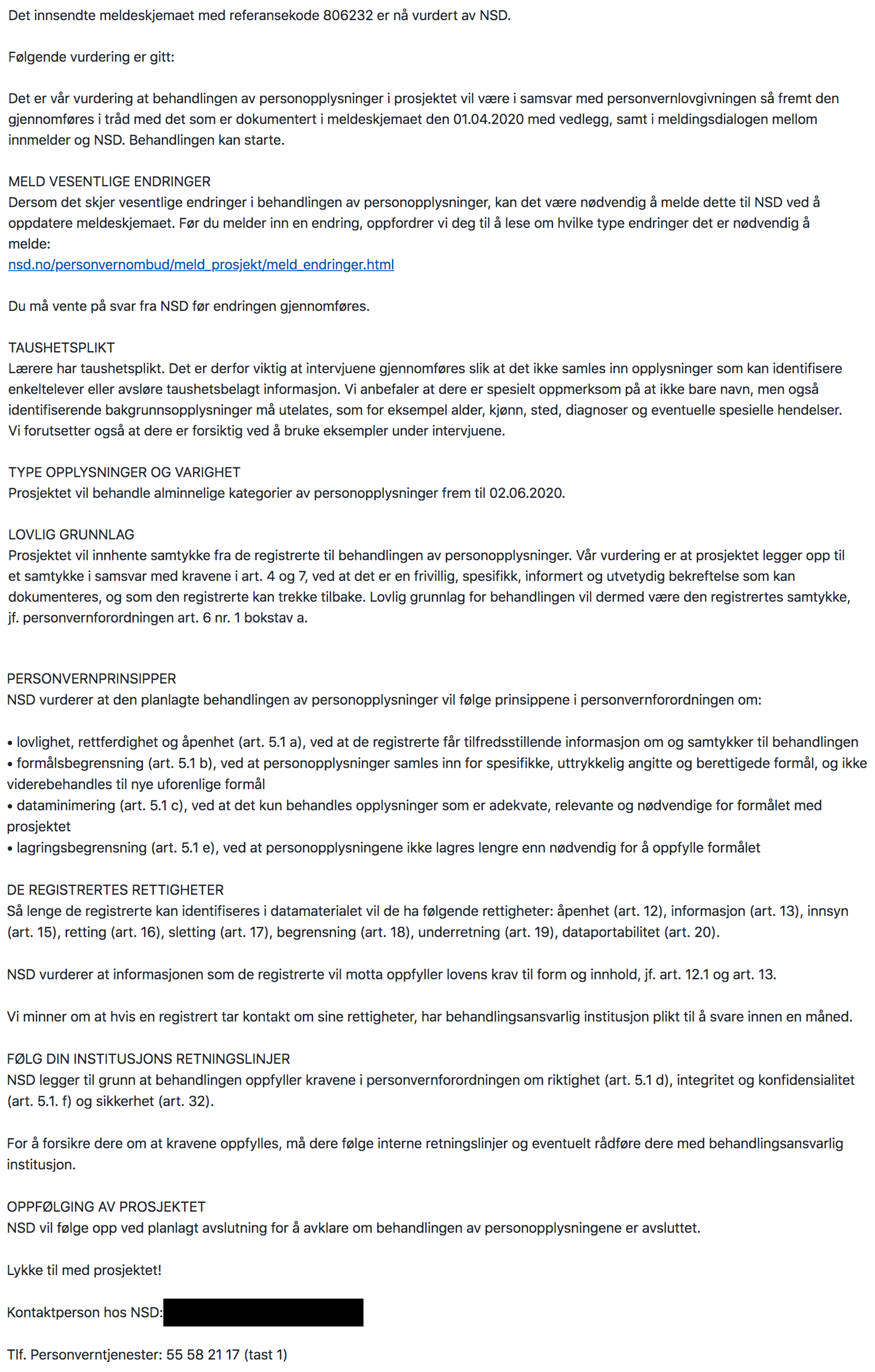
□ å delta i intervju

Jeg samtykker til at mine opplysninger behandles frem til prosjektet er avsluttet, *02.06.2020*

------------------------------------------------------------------------------------------------------------

(Signert av prosjektdeltaker, dato)

# Appendix D: Approval from NSD



Det innsendte meldeskjemaet med referansekode 806232 er nå vurdert av NSD.

Følgende vurdering er gitt:

Det er vår vurdering at behandlingen av personopplysninger i prosjektet vil være i samsvar med personvernlovgivningen så fremt den gjennomføres i tråd med det som er dokumentert i meldeskjemaet den 01.04.2020 med vedlegg, samt i meldingsdialogen mellom innmelder og NSD. Behandlingen kan starte.

MELD VESENTLIGE ENDRINGER
Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til NSD ved å oppdatere meldeskjemaet. Før du melder inn en endring, oppfordrer vi deg til å lese om hvilke type endringer det er nødvendig å melde:
nsd.no/personvernombud/meld_prosjekt/meld_endringer.html

Du må vente på svar fra NSD før endringen gjennomføres.

TAUSHETSPLIKT
Lærere har taushetsplikt. Det er derfor viktig at intervjuene gjennomføres slik at det ikke samles inn opplysninger som kan identifisere enkeltelever eller avsløre taushetsbelagt informasjon. Vi anbefaler at dere er spesielt oppmerksom på at ikke bare navn, men også identifiserende bakgrunnsopplysninger må utelates, som for eksempel alder, kjønn, sted, diagnoser og eventuelle spesielle hendelser. Vi forutsetter også at dere er forsiktig ved å bruke eksempler under intervjuene.

TYPE OPPLYSNINGER OG VARIGHET
Prosjektet vil behandle alminnelige kategorier av personopplysninger frem til 02.06.2020.

LOVLIG GRUNNLAG
Prosjektet vil innhente samtykke fra de registrerte til behandlingen av personopplysninger. Vår vurdering er at prosjektet legger opp til et samtykke i samsvar med kravene i art. 4 og 7, ved at det er en frivillig, spesifikk, informert og utvetydig bekreftelse som kan dokumenteres, og som den registrerte kan trekke tilbake. Lovlig grunnlag for behandlingen vil dermed være den registrertes samtykke, jf. personvernforordningen art. 6 nr. 1 bokstav a.

PERSONVERNPRINSIPPER
NSD vurderer at den planlagte behandlingen av personopplysninger vil følge prinsippene i personvernforordningen om:

- lovlighet, rettferdighet og åpenhet (art. 5.1 a), ved at de registrerte får tilfredsstillende informasjon om og samtykker til behandlingen
- formålsbegrensning (art. 5.1 b), ved at personopplysninger samles inn for spesifikke, uttrykkelig angitte og berettigede formål, og ikke viderebehandles til nye uforenlige formål
- dataminimering (art. 5.1 c), ved at det kun behandles opplysninger som er adekvate, relevante og nødvendige for formålet med prosjektet
- lagringsbegrensning (art. 5.1 e), ved at personopplysningene ikke lagres lengre enn nødvendig for å oppfylle formålet

DE REGISTRERTES RETTIGHETER
Så lenge de registrerte kan identifiseres i datamaterialet vil de ha følgende rettigheter: åpenhet (art. 12), informasjon (art. 13), innsyn (art. 15), retting (art. 16), sletting (art. 17), begrensning (art. 18), underretning (art. 19), dataportabilitet (art. 20).

NSD vurderer at informasjonen som de registrerte vil motta oppfyller lovens krav til form og innhold, jf. art. 12.1 og art. 13.

Vi minner om at hvis en registrert tar kontakt om sine rettigheter, har behandlingsansvarlig institusjon plikt til å svare innen en måned.

FØLG DIN INSTITUSJONS RETNINGSLINJER
NSD legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

For å forsikre dere om at kravene oppfylles, må dere følge interne retningslinjer og eventuelt rådføre dere med behandlingsansvarlig institusjon.

OPPFØLGING AV PROSJEKTET
NSD vil følge opp ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet.

Lykke til med prosjektet!

Kontaktperson hos NSD: [redacted]

Tlf. Personverntjenester: 55 58 21 17 (tast 1)

NTNU

Kunnskap for en bedre verden