LUMAT
INTERNATIONAL JOURNAL ON MATH, SCIENCE AND TECHNOLOGY EDUCATION

Research article

Norwegian mathematics teachers' conceptions of programming in mathematics education

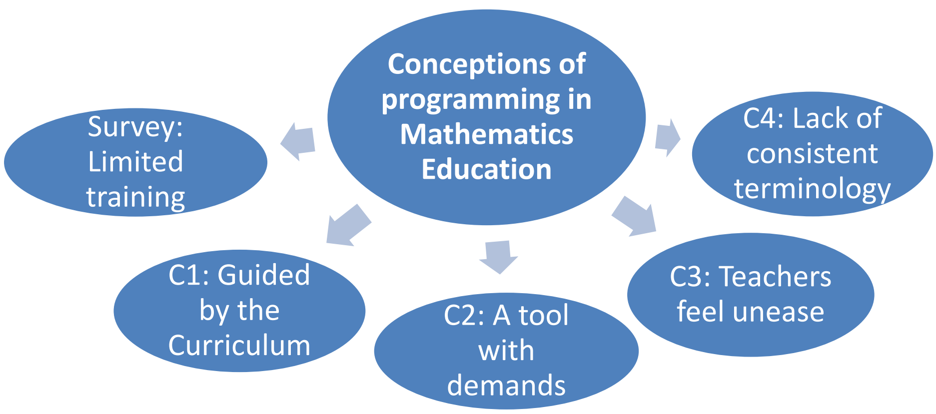
Martyna Katarzyna Fojcik [1, 2]

[1] Volda University College, Norway
[2] University of Agder, Norway

Abstract: As programming is being integrated into mathematics education in Norway, it is increasingly important to understand how teachers perceive and implement programming. This study investigates Norwegian mathematics teachers' conceptions, that is their situated understandings and rationales, regarding programming in mathematics education at the primary and lower secondary school levels. Data were collected through semi-structured interviews with four teachers and a nationwide questionnaire answered by 215 teachers, providing both qualitative and quantitative insights. The interviews were analysed using thematic analysis, while the questionnaire provided contextual quantitative insights. The results show that (1) the majority of Norwegian mathematics teachers have no formal education in programming, and many have not received training in programming, yet they are largely positive about introducing programming to their pupils, (2) teachers align their teaching of programming with and are guided by the mathematics curriculum and the exam, (3) teachers view programming as a tool in the subject, however, they are often hindered from realizing programming's potential by its challenges and limitations, (4) teachers feel unprepared and sometimes uncomfortable teaching programming in mathematics, and (5) there appears to be a lack of appropriate collaborative discourse with consistent terminology on programming in mathematics education among primary and lower secondary school teachers in Norway, which may be contributing to the fragmentation of practice. By shedding light on how teachers conceptualize programming, this study contributes to understanding the pedagogical and structural challenges and opportunitiesS of integrating programming into mathematics education, and highlights areas for future support.

Keywords: programming, mathematics education, teacher conceptions, curriculum,

Correspondence: martyna.fojcik@hivolda.no



Received 23 June 2025 / Accepted 4 December 2025 / Published 18 December 2025
Published by the University of Helsinki, Finland and LUMA Centre Finland.

1 Introduction

Different approaches have been taken to introduce programming into primary and secondary education (Bocconi et al., 2022; Misfeldt et al., 2019). In some countries, such as the United Kingdom and Poland, programming has been introduced through new technology- or informatics-driven subjects. In others, including Sweden and France, programming has been embedded into the mathematics curriculum.

In Norway, programming was formally introduced into the national curriculum (LK20) in autumn 2020, It appears in four subjects:

- Mathematics[1] (six Competency aims, one each year from Year 5 to Year 10)
- Science (two Competency aims, one after Year 7 and one after Year 10)
- Art & Craft (one Competency aim after Year 7)
- Music (one Competency aim after Year 10)

Although programming is presented as an interdisciplinary skill, the mathematics curriculum contains the most detailed and frequent references to programming. It also includes related concepts such as computational thinking (CT), algorithms, instructions, or programmes. However, these terms are not explicitly defined. For instance, Denning and Tedre (2019) define CT as the mental skills and practices for (1) “designing computations that get computers to do jobs for us, and (2) explaining and interpreting the world as a complex of information processes” (p. 4). No such definition is provided in the Norwegian mathematics curriculum, as programming is listed in a broader set of digital tools:

> “Digital skills in mathematics refers to the ability to use graphing tools, spreadsheets, CAS [computer algebra systems], dynamic geometry software [DGS], and programming to explore and solve mathematical problems” (Ministry of Education and Research, 2019, p. 5, abbreviations explained or added in square brackets).

This formulation positions programming as one of several digital tools for mathematical exploration, yet it does not specify what the programming process entails or how it should develop across year levels.

While the curriculum outlines programming as a digital skill, how this is interpreted and enacted by teachers remains underexplored. Although Nordby et al. (2022) examined how programming and CT are incorporated into mathematics education, there remains a notable gap in understanding how teachers make sense of and respond to curriculum and how it shapes their conceptions. Turgut et al. (2024) argue that more research is needed to explore teachers’ voices and rationales regarding programming in mathematics education, as such insights are crucial for informing classroom implementation.

---

[1] Mathematics has competency aims each year, whereas other subjects have joint competency aims Year 1–4, 5–7 and 8–10.

In their study Broley et al. (2018) investigated mathematicians' views on teaching and learning with programming, concluding with 6 levels of individual interaction with programming: (L0) observe results of a computer program, (L1) manipulate the interface of a program, (L2) analyse the code of a program, (L3) Modify existing code to accomplish something new, (L4) construct code of a program, (L5) code, test and verify a program. These levels refer to how a mathematician might use programming for teaching and learning mathematics at university level, yet they offer a useful lens for considering how programming might be conceptualized across educational levels, including primary and lower secondary education.

While previous studies have explored how programming is implemented in mathematics classrooms, few have examined how teachers themselves conceptualize programming within the framework of the Norwegian curriculum. This study addresses that gap by investigating mathematics teachers' conceptions, offering insights into how programming is understood and potentially enacted in primary and lower secondary education. In this context, *mathematics education* refers to the teaching and learning mathematics in compulsory education, and *mathematics* refers specifically to the subject taught in primary and lower secondary schools.

The article begins with a brief overview of previous research on programming in Norwegian mathematics education. It then presents the conceptual framework and research question, followed by a detailed account of the methodology. The results section highlights key findings from the thematic analysis, which are discussed in relation to theoretical and empirical perspectives. The article concludes with reflections on limitations and implications for practice.

2 Previous research

Research in Norway has shown that in-service primary school mathematics teachers vary in the extent to which they integrate CT and programming into their teaching. Many have limited knowledge and experience with these concepts, and few have had opportunities to develop their competence further (Nordby et al., 2022). Similar challenges are evident among pre-service teachers. Kaufmann et al. (2022), for example, found that those with no prior programming experience were less likely to associate programming with problem solving, compared to their more experienced peers.

At the pupil level, Refvik and Opsal (2023) observed that only some pupils were able to recognize and utilize the relationship between programming, problem solving, and mathematics. They attributed this to two main reasons: (1) a weak conceptual link between problem solving in CT/programming and mathematical problem solving, and (2) the lack of pedagogical instruction that teachers could use to support this connection in the classroom.

Despite these challenges, both pre- and in-service teachers tend to express positive attitudes towards programming in mathematics. Kaufmann et al. (2022) reported that pre-service teachers valued the pedagogical potential of programming, such as increased motivation and opportunities for differentiation, sometimes even more than its connection to specific mathematical topics like geometry or algebra. Turgut et al. (2024) synthesised six common teachers' perceptions from the literature on programming in mathematics education: (1) a perceived link between programming practices and mathematics; (2) inconsistent use and understanding of the terms 'programming' and 'CT'; (3) generally positive attitudes towards programming; (4) feelings of unpreparedness; (5) practical challenges in implementation; and (6) a need for professional support. Similarly, Kilhamn et al. (2021a) identified four reasons Swedish in-service teachers gave for including programming in mathematics: (a) its potential as a powerful tool, (b) its ability to increase engagement, (c) its role in developing CT, and (d) its value as a way of learning mathematics. Also another Swedish study, Johansson et al. (2023) pointed to teachers experiencing inadequacy to teach programming in mathematics only after short training.

These findings highlight both the perceived value and the practical challenges of integrating programming into mathematics education. However, they also point to a deeper issue: while teachers recognize the potential of programming, how they interpret and enact curriculum expectations remains unclear.

While national strategies and curriculum guidance aim to support programming integration, their impact on how mathematics teachers interpret and respond to these initiatives in practice remains unexplored. Previous research suggests that teachers' interpretations of curriculum documents play a significant role in shaping how programming is understood and enacted in mathematics education and that mathematics teachers often feel unprepared and uncomfortable teaching programming (Johansson et al., 2023; Nordby et al., 2022). However, the ways in which teachers navigate and make sense of curriculum expectations remain underexamined. This study addresses that gap by investigating Norwegian mathematics teachers' conceptions of programming in mathematics education.

Finally, the lack of clear definitions of programming and the varied interpretations of CT in educational research (e.g., Bergqvist, 2021; Broley et al., 2018; Fojcik, 2022; Nordby et al., 2022; Turgut et al., 2024) contribute to conceptual ambiguity. This ambiguity may shape teachers understand and use programming in mathematics education. These findings, along with the conceptual ambiguity surrounding programming and CT, motivate a closer examination of how teachers conceptualize programming in mathematics education. This is further developed in the next section.

3 Conceptual framework

This section presents the definitions of key terms guiding the study. Including the conceptual framework that draws on Phillipp's (2007) definition of conceptions as dynamic structures, Vergnaud's (2009) contextual view of conceptual development and Misfeldt et al.'s (2019) work on interdisciplinary meaning-making between mathematics and programming. Then the following sections address the relationship between programming and mathematics in school, and the presents the research question and its rationale.

3.1 Concepts and conceptions

The term *concept* typically refers to a formally defined notion or construct within a domain. Tall and Vinner (1981) distinguished between the concept image, the collection of mental pictures and processes an individual holds, and the concept definition, which refers to the formal, shared understanding within a discipline. This distinction is especially relevant, as teachers may express beliefs and perceptions that diverge from formal definitions, particularly when engaging with new or interdisciplinary content.

Vergnaud (2009) emphasises that a concept's meaning emerges from a variety of situations and that no single concept suffices to analyse a situation. This view highlights the contextual and interconnected nature of conceptual development, which is particularly relevant in interdisciplinary contexts such as programming in mathematics education.

Philipp (2007) defines *conception* as "a general notion or mental structure encompassing beliefs, meanings, concepts, propositions, rules, mental images, and preferences"(p. 259). This broad definition supports a holistic view of teachers' mental representations, conscious beliefs, encompassing both cognitive and affective dimensions that influence teaching practices and decision-making. In this study, the term *conceptions* is used to represent a dynamic cognitive structure through which individuals organize and relate to known and experienced concepts.

In this study, *teachers' conceptions* refer to their views, understandings, and rationales regarding programming in mathematics education. These conceptions reflect both personal (teachers as learners) and professional (teachers as educators) dimensions. They include how teachers relate concepts, structure assumptions, and express tendencies in their teaching. This view aligns with Furinghetti and Pehkonen's (2002) stance that teachers' conceptions are shaped by both individual experiences and institutional contexts.

While the term *conception* has been explored in mathematics education, its application to programming remains relatively underdeveloped. This study builds on emerging work (e.g., Misfeldt et al., 2019) to extend the notion of teacher conceptions into the interdisciplinary space between mathematics and programming. Misfeldt et al. (2019), drawing on Papert (1980) and Vergnaud (1998), argue that "active articulation and representation of the relation between the mathematics and programming plays a critical role in shaping individual conceptions" (p. 5).

3.2 Relation between programming and mathematics in schools

Papert (1980) introduced programming as a "powerful idea" for learning, emphasising creativity, exploration, and the development of conceptual understanding through practical engagement and shared cultural practices. However, Papert's vision was not primarily situated in the realities of classroom teaching or curriculum constraints.

In more recent research, Misfeldt et al. (2020) identified four ways teachers could relate programming to mathematics: (1) specific relation, (2) explicit relation, (3) implicit relation and (4) no or weak relation. These categories reflect varying degrees of integration, with many teachers in their study tending toward implicit or weak connections. Similarly, Kilhamn et al. (2021b) proposed four categories: (a) only programming, (b) mathematics as context for programming, (c) programming as a tool for efficient calculation, and (d) programming as a tool to inquire into mathematics. Again, they reported that most common practices involved minimal or surface-level integration, though this may vary by context.

Bergqvist (2021) examined different interpretations of programming in mathematics teaching, distinguishing between a narrow view (focused on coding) and a broader view (programming as more than coding), and found that narrow interpretation was more commonly adopted. In contrast, Tossavainen et al. (2024) investigated how programming can serve as a mediator of mathematical thinking. Their study not only demonstrates how programming can mediate mathematical thinking but also challenges existing categorizations by suggesting that programming itself may constitute mathematical activity, thereby contributing to the broader discourse on the role of programming in mathematics education.

Rolandsson (2013) and Dilling et al. (2024) highlight the challenges pre- and in-service teachers encounter when programming is introduced through curriculum reform. These include uncertainty about what content to teach, a lack of pedagogical models, and limited access to professional development.

Taken together, these studies highlight both conceptual debates about the role of programming in mathematics and the practical challenges teachers face. Such findings emphasise the need to understand how teachers conceptualise programming, not only as a technical skill, but as a pedagogical and disciplinary tool within mathematics education.

3.3 Research question

The research question guiding this study is: *What conceptions do Norwegian mathematics teachers reveal regarding the use of programming in teaching and learning mathematics?*

The question is grounded in the connection between teachers' conceptions and pupils learning. As Philipp (2007) notes, mathematics teachers' conceptions are inseparable from their knowledge and experiences. Opre (2015) similarly argues that teachers' conceptions are "key factors, being regarded as essential determinants of the instructional activity and of the students' learning process" (p. 230).

Understanding teachers' conceptions offers valuable insight into how they interpret, adapt, and implement curriculum content. This is particularly relevant in areas like programming, where formal training may be limited and interpretations vary widely. This study seeks to capture the nuanced ways in which programming is understood and positioned within mathematics teaching, including how teachers make sense of curriculum expectations, navigate pedagogical challenges, and articulate programming's role in their subject.

In this context, conceptions refer not only to teachers' knowledge and experience with programming, but also their professional approaches, disciplinary perspectives and the institutional contexts. As Opre notes, that teachers hold "a variety of beliefs and this influences the quality of their performance" (2015, p. 230). Turgut et al. (2024, p. 4) describe these as teachers' "personal orientations" encompassing beliefs, values, preferences, concerns and personal choices. This study adopts the broader term *conceptions* to reflect the integrated nature of these influences.

Moreover, teaching is shaped by context. As Bingolbali and Monaghan (2008) observe, "a teacher is not a *lone agent* but, knowingly or unknowingly, adapts his/her teaching to the class and/or institution s/he is teaching in" (pp. 21–22, italics in original). Investigating teachers' conceptions is therefore essential, as these shape both instructional practices and pupils' learning outcomes.

This study adopts a conception-based lens, viewing teachers' statements not merely as thematic patterns but as expressions of their underlying mental structures—how they make sense of programming in relation to mathematics education. This approach aligns with prior work in mathematics education (e.g., Furinghetti & Pehkonen, 2002; Misfeldt et al., 2019), where teachers' conceptions–though not always in context of programming–are reconstructed from qualitative data to understand how they interpret and enact disciplinary practices.

These theoretical perspectives inform the methodological choices in this study. The next section outlines the research design, data collection, and analytical approach used to investigate teachers' conceptions in depth.

4 Method

Guided by this conception-based framework, the study employed a mixed-methods design to explore how teachers understand and experience programming in mathematics education. This study combines qualitative semi-structured interviews with a broader quantitative contextualizing questionnaire to investigate Norwegian mathematics teachers' conceptions of programming. The interviews provided in-depth insight into individual teachers' conceptions, while the questionnaire offered contextual information about a wider group of teachers' self-reported experiences, beliefs, and challenges (Clark et al., 2021). Developed in parallel and addressed overlapping themes, the two methods allowed both breadth and depth in the analysis.

While the number of interview participants was limited, four teachers, the aim was to gain in-depth insight into individual teachers' conceptions rather than to generalise across the population. Ethical approval for the study was obtained from the Norwegian Council of Ethics (SIKT), and all participants took part voluntarily.

The questionnaire was designed to offer a broader view of how programming is perceived and experienced across a larger sample of teachers. The 215 responses were analysed descriptively to identify patterns in teachers' reported experiences and to contextualize the themes emerging from the interviews. It also informed the interpretation of interview findings by highlighting patterns and variations in teachers' reported confidence, challenges, and perceived benefits.

This study employs a thematic analysis approach (Braun & Clarke, 2006) to explore and organize the conceptions Norwegian mathematics teachers hold regarding the use of programming in teaching and learning mathematics. Thematic analysis was chosen for its flexibility and suitability in identifying, analysing, and reporting patterns (themes) within qualitative data, and connecting them to items from the questionnaire. It allows for a rich, detailed, yet complex account of data, which aligns with this study's research question. While thematic analysis is the primary analytical method, the resulting themes are interpreted as representations of teachers' conceptions—structured understandings shaped by experience, context, and pedagogical reasoning. Thematic analysis is used not only to identify recurring patterns in the data but to reconstruct teachers' conceptions—that is, their situated, structured understandings of programming in mathematics education, which reflect more than surface-level opinions. In this study, thematic analysis is used not only to identify recurring patterns in the data but to reconstruct teachers' conceptions—that is, their situated, structured understandings of programming in mathematics education. These conceptions encompass both cognitive and affective dimensions, and are shaped by personal experience, institutional context, and pedagogical reasoning. Following sections describe the context of the study, each data collection method, and the analytical approach.

4.1 Context for the study

In Norway, the revised national curriculum (LK20) was published in 2019 and implemented in the 2020–2021 school year. One of the key changes was the introduction of programming across several subjects, including mathematics. However, the onset of the COVID-19 pandemic in spring 2020 shifted many teachers' priorities, which may have delayed the integration of programming into classroom practice.

The mathematics curriculum in LK20 consists of several components. This study focuses on four: (1) the six *core elements* of the subject (e.g., "Abstraction and generalisation", "Exploration and problem solving"); (2) the description of *basic skills*, including digital skills, as they apply to mathematics; (3) the *competence aims* and assessment criteria for each year from Year 2 to Year 10 (e.g., after Year 5, pupils should "create and programme algorithms with the use of variables, conditions and loops"; (Ministry of Education and Research, 2019, p. 9)); and (4) the structure of assessment, including the national exam at the end of Year 10.

The concept of CT is translated in the Norwegian mathematics curriculum as *algoritmisk tenkning* (AT), or *algorithmic thinking*. Since there is no other word in Norwegian for "computational", and the official translation uses "algorithmic", it is unclear what is meant by this notion. In this study, the term is used as *CT/AT* to reflect this translation and to highlight its potential ambiguity, as no explanation is given in the curriculum for the difference between "computational" and "algorithmic". The curriculum includes only one explicit reference to CT/AT, stating: "Computational thinking is important in the process of developing strategies and approaches to solve problems, and means breaking a problem down into sub-problems that can be solved systematically" (Ministry of Education and Research, 2019, p. 2). This potentially ambiguous definition raises questions about how teachers interpret and operationalize CT/AT in practice.

In Norway there are multiple pathways to becoming a teacher, and these have evolved over time. In this study, all participants are referred to as *Norwegian mathematics teachers*, based on their self-identification. The study does not distinguish between different levels of formal mathematics education or teaching certification.

4.2 Data gathering – interview

Data were collected through semi-structured interviews (Clark et al., 2021) with in-service mathematics teachers. The interview guide was designed to explore teachers' personal conceptions of programming, coding and CT/AT in mathematics education, as well as their experiences with introducing programming through mathematical activities in their classrooms and schools. The questions focused on broader perceptions, such as definitions, rationale or understanding, rather than specific classroom practice.

Although this study distinguishes between the terms *programming* and *coding*—with programming referring to the entire process (including planning, designing, writing, testing, and debugging), and coding referring specifically to the act of writing code in a programming language, some items in the questionnaire combined these terms (programming/coding). This was done to differentiate them from broader processes such as computational thinking (CT/AT), while maintaining accessibility for respondents.

Between January and October 2023, 50 schools were contacted via email through school principals or administrators. The email included information about the study and an invitation for the interview to forward it to mathematics teachers. Three schools agreed to participate, resulting in four volunteer participants.

All four teachers primarily taught Years 8–10 (lower secondary), though some had experience with primary levels due to two schools being a combined Year 1–10 institutions. The participants represented both urban and rural schools and had diverse backgrounds in mathematics education and programming. Some were newly qualified teachers, while others had many years of practice. All four teachers had some training or prior exposure to programming, ranging from a brief two-hour introduction to a formal teacher development course with ECTS credits. To ensure confidentiality, especially in smaller communities, individual profiles are not reported, instead, concise characteristics were provided. The names used in this study (Henry, Nora, Lisa, and Anne) are pseudonyms.

Before the interview, participants received an overview of the project and signed a consent form. The interviews were audio-recorded and transcribed. No personal data were collected, and all data were anonymized. While the number of interview participants was small, the study prioritised depth over breadth, and the inclusion of a broader questionnaire helped contextualize the findings.

Interviews were conducted in person at the participants' schools and lasted between 35-47 minutes. Selected excerpts were translated into English for this study. In the translation process, minor modifications were made to improve clarity for an English-speaking audience. These included: (1) replacing vague pronouns (e.g., "them," "it") with specific referents (e.g., "pupils," "curriculum") if they were repeated from the questions, and (2) adding brief clarifications in square brackets when participants referred to earlier parts of the conversation, or when smoothing conversational shifts or incomplete thoughts to enhance readability, while preserving the original meaning.

4.3 Data gathering – questionnaire

An online questionnaire was distributed to all primary, lower secondary and combined schools in Norway. The invitation for the questionnaire, sent via email to school principals or administrators, included a description of the study, and a request to forward the link to mathematics teachers. While some schools responded positively or declined, most did not respond, and some emails were returned due to invalid addresses.

Participation was voluntary, and informed consent was implied by completing and submitting the questionnaire. No personal data were collected, including IP addresses or metadata on how many recipients opened the link. The questionnaire was sent to 2 413 schools in Norway, and 215 teachers responded.

The questionnaire was developed in Norwegian and reviewed by both language experts and mathematics educators. Inspired by Swedish study by Misfeldt et al. (2019), which focused on teachers' conceptions of programming. The questionnaire addressed similar themes as the interviews, as both instruments were developed in parallel. It did not include open-ended items. Aside from a few background questions, most items used a 5-point Likert scale ranging from 1 ("not at all") to 5 ("to a great extent"), with 3 as a neutral midpoint. The items focused on teachers' self-reported pedagogical, mathematical, and programming knowledge and experience. Example items included:

- "I feel confident in my own competence in..." (e.g., mathematics, programming)
- "I think it is okay to..." (e.g., make a mistake in the classroom, say 'I don't know')
- "It was challenging for me to..." (e.g., find time to learn programming, collaborate with colleagues)

The final section included 12 adapted items from Misfeldt et al. (2019) addressing perceived benefits of programming for both teachers and their pupils. In contrast to the original 4-point Likert scale, this study used a 5-point scale to allow for a neutral midpoint.

4.4 Data analysis

The analysis began with the interview data and followed the six phases of thematic analysis as outlined by Braun and Clarke (2006) in an exploratory, inductive and iterative approach, see Table 1. The transcribed interviews were coded for explicit statements that reflected teachers' conceptions, including their views, attitudes, knowledge and experiences related to programming in mathematics education. With iterations that moved perspectives and tried to discover broader connections. A key assumption underpinning the analysis is that teachers expressed their thoughts and experiences truthfully and openly.

These conceptions are reconstructed from teachers' statements and interpreted as manifestations of their underlying mental structures, encompassing their knowledge, experiences, professional approaches, and disciplinary perspectives.

Following the interview analysis, the identified codes and topics were compared with relevant items from the questionnaire. This comparison aimed to broaden the understanding of teachers' conceptions by situating individual perspectives within a larger context. While the interviews served as the primary data source for identifying teachers' conceptions, the questionnaire data were analysed descriptively to identify broader trends and to support, contrast, or nuance the themes emerging from the interviews. All diagrams presented in the Results-sections are based on questionnaire data and represent responses from 215 participants, shown as percentages.

Table 1. Phases of the thematic analysis

Phase	What the author did
Familiarizing yourself with your data	Transcribed the interviews, read and re-read the data while making initial notes. Organised notes in various formats (e.g., spreadsheets, printed notes, tables, concept maps etc.) to gain a clearer overview.
Generating initial codes	Marked initial codes based on features of the data, including specific wording and discussed characteristics. Many codes initially aligned with the interview guide topics (e.g., curriculum, practice, definitions, challenges etc.), but later iterations moved beyond these to capture broader patterns and nuances.
Searching for themes	Combined and grouped relevant codes. Looked for underlying conceptions beyond surface-level expressions. Created drafts of each teacher's conceptions individually.
Reviewing themes	Shifted from individual participant-level codes to a cross-case perspective. Identified commonalities and divergences among participants, things the participants had in common, or what they disagreed about.
Defining and naming themes	Refined themes to represent mathematics teachers' conceptions of programming in mathematics education. Developed broader, holistic conceptions supported by both interview data and relevant questionnaire items.
Producing the report	Iteratively phrased and rephrased the themes. Related interview-derived conceptions to questionnaire findings to present a comprehensive view of Norwegian mathematics teachers' conceptions. Defined each theme and illustrated it with representative data excerpts.

A brief example of the analysis is presented below, based on a quote from Anne:

> **Interviewer** (14:49): "So, the new curriculum introduced a number of concepts that have not been used much before, one of these concepts is algorithmic thinking [CT/AT]. Do you see any connection between that term and programming? Or are they more like different things?"
> **Anne** (15:17): "Mm, what should I say? I tried a lot to familiarize myself with a bit of 'why we should have programming'... I understand it a bit, but I do not understand it completely. And then it is perhaps difficult for me to see the context of that, but it may be that I understand it ... I do not really know".

This excerpt was initially coded under 'curriculum' due to the framing of the interview question. However, Anne does not directly engage with the concept of CT/AT. Instead, her response reveals a struggle to understand the rationale behind programming in the curriculum and reflects broader challenges in making sense of its purpose. Her lack of engagement with the specific terminology may also indicate a disconnect between curriculum language and teachers' knowledge and experience.

Anne's uncertainty and hesitation suggest a view of programming as a curricular obligation rather than a meaningful pedagogical practice. Her response indicates that programming has not yet been internalized as part of her professional knowledge but is instead perceived as an externally imposed requirement. While this may point to a lack of clarity and support structures in the curriculum, it could also reflect a more personal struggle with unfamiliar terminology or a general discomfort with programming. These alternative readings highlight the complexity of teacher sense-making processes.

Although this is a single instance, it reflects a broader pattern observed across the dataset, where teachers expressed uncertainty about the rationale for programming in mathematics education. This contributed to the development of the theme C1 – "*Programming is introduced in mathematics education because it is required by the curriculum and assessed in the exam in mathematics*". This theme captures a shared conception of programming as a top-down curricular demand rather than a pedagogically understood and integrated practice. It highlights the difficulties teachers face in internalizing programming as part of their professional knowledge and practice.

5 Results

The results are presented in two parts. The first offers a brief overview of selected questionnaire findings, providing contextual insights into Norwegian mathematics teachers' educational qualifications, teaching practices, and self-assessed confidence in different areas. The second presents four overarching conceptions identified through thematic analysis. These conceptions reflect how Norwegian mathematics teachers perceive the role of programming in mathematics education. Each theme is illustrated with excerpts from interviews and contextualized with relevant questionnaire findings.

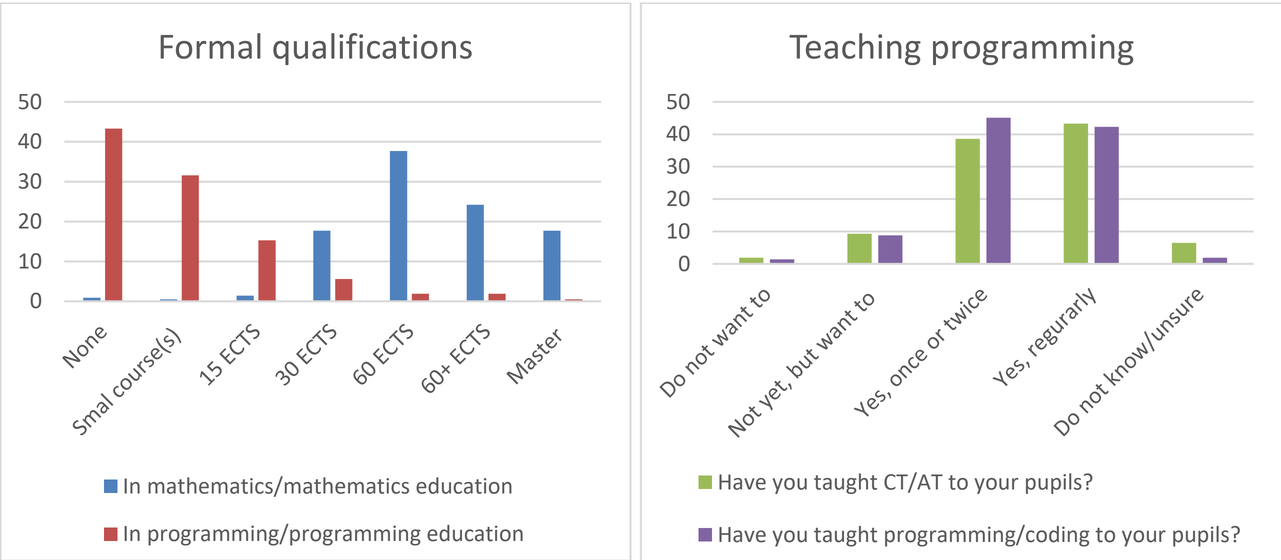
5.1 About Norwegian mathematics teachers

According to the questionnaire, Norwegian mathematics teachers have strong formal qualifications in mathematics and mathematics education but limited in programming. In Norway, a minimum of 30ECTS is required to teach the mathematics subject in primary school, and 60 ECTS in lower secondary school. Among the participants, approximately 97% reported having 30 or more ECTS in mathematics or mathematics education, while just below 10% reported equivalent coursework in programming or programming education, see Figure 1a. Notably, there are currently no national requirements for programming competence among mathematics teachers.

In the questionnaire, teachers self-assessed their competence levels in mathematics, programming and using digital tools in general. These responses mirrored the trends in formal qualifications. On a 5-point Likert scale, an average of around 87% of participants rated themselves as confident (to a little and great extent) in mathematic/mathematics education. Confidence in using digital tools was lower, at almost 60%, and confidence in programming/programming education was markedly lower, at just under 19%.

When discussing tools that can be used to teach programming, especially for children or teens, there are two distinguishable types: (1) visual- or block-based programming languages/environments, also called drag-and-drop environments, which are interfaces where the user gets all the code presented as "blocks" one can just put together like puzzles, or (2) text-based programming languages/environments where the user needs to enter text-code based on specific symbols and syntax that form certain programming expressions (Duncan et al., 2014; Weintrop & Wilensky, 2017). When asked about the total amount of training or education they had received in programming, 63% of the teachers reported having no training in visual-based programming, and 56.7% reported having no training in text-based programming.

Figure 1. a. and b. Teachers' responses about qualifications and their teaching practices



Despite limited training in programming (regardless of whether visual- or text-based type), many teachers reported engaging with programming-related content in their teaching, see Figure 1b. Approximately 39% reported having taught CT/AT once or twice, and 45% had done it the same for programming/coding. Over 40% indicated that they now teach such content regularly. This suggests that programming is being implemented in practice, often without formal preparation or strong confidence. These conditions may be relevant for understanding how teachers approach or conceptualize its role in mathematics education, and they provide important context for interpreting the more nuanced perspectives that emerged in the interviews.

While the questionnaire captures broader national trends, the interviews provide insight into how individual teachers conceptualize programming in mathematics education. All interview participants approached programming with a positive and motivated stance, each with their own rationale. Yet their shared conceptions tended to focus on the challenges and limitations encountered in practice, not as a sign of resistance, but as reflections from teachers committed to integrating programming into mathematics. The following sections present four overarching conceptions derived from thematic analysis of interview data, supported by patterns observed in the questionnaire responses.

5.2 Norwegian mathematics teachers' conceptions

Based on thematic analysis of interview data, supported by questionnaire responses, four overarching conceptions were identified. These reflect how teachers understand, experience, and position programming within mathematics education.

Conception 1 (C1): Programming is introduced in mathematics education because it is required by the curriculum and assessed in the exam in mathematics

Norwegian mathematics teachers in this study frequently referred to the curriculum as the primary driver for introducing programming into their teaching. The curriculum is seen as a guiding document that determines not only the content and progression of mathematics education but also the methods and approaches to be used. It determines topics (e.g., geometry or probability), sequencing (e.g., what is taught when), methods (e.g., what to do, explore, solve) and the overreaching approach to mathematics, as articulated through the competence aims.

Teachers described programming as something that has been added to the curriculum, often introduced without sufficient support or preparation. Lisa described the curriculum as a driving force behind her decision to engage with programming:

> "Yes, I have [chose to learn programming], because we have to … there are competency aims that say, pupils should be able to solve using programming" (L27:11).

Nora similarly emphasised the curriculum's authority:

> "It is the curriculum that governs – it should at least do that" (N13:14).

While some teachers valued the new curriculum's emphasis on reasoning and open-ended tasks, they also noted that programming adds to the workload. Henry reflected:

> "The curriculum is now much more focused on pupils understanding, arguing, reasoning … and in addition, programming comes on top, right? And that makes it demanding" (H04:35d).

Others like Anne, expressed uncertainty about programming's role in mathematics, indicating a lack of clarity about its purpose:

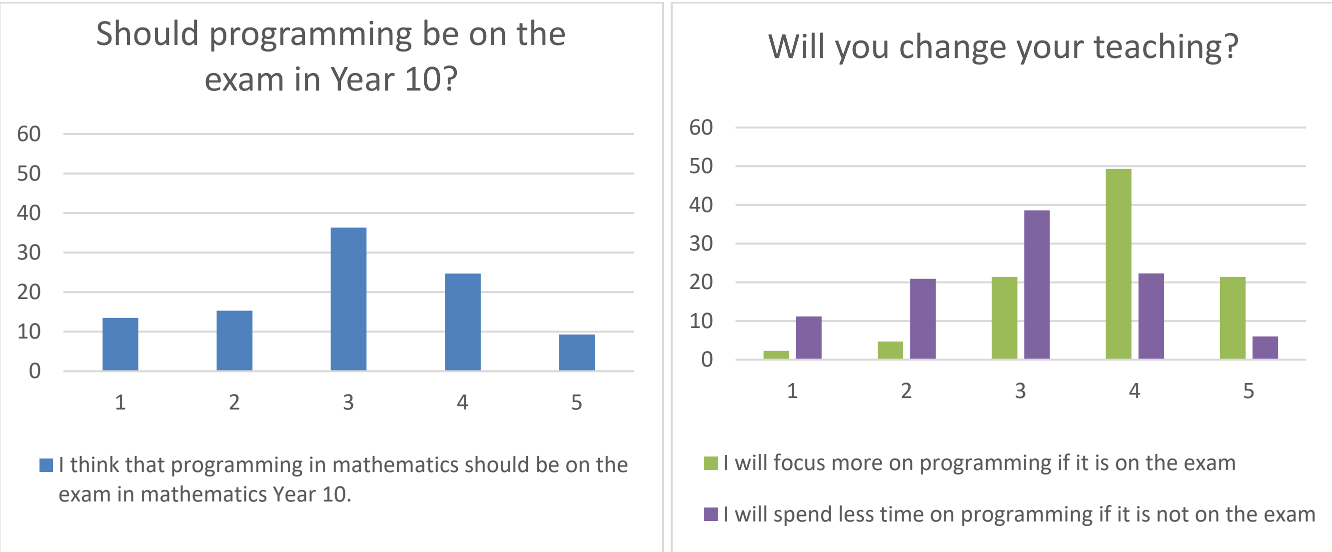
> "I tried a lot to familiarize myself with why we should have programming … I understand it a bit, but I do not understand it completely" (A15:17)

Lisa also noted that the curriculum's emphasis on reflection and assessment made it difficult to prioritise programming. She felt it was "imposed" on, then stating:

> "If we [my pupils] are to be assessed on programming, it must be assessed on the exam" (L33:18)
> "I am perfectly fine with programming not being on the exam, but it has to be the same everywhere, and we [teachers] have to know about it" (L34:16).

This connection between curriculum and assessment was also reflected in the questionnaire data. This perception is reinforced by questionnaire responses, which show that teachers' emphasis on programming is closely tied to whether it is assessed in the national exam. When asked whether programming should be assessed in the national Year 10 exam, 36.3% of teachers responded neutrally. The remaining responses were somewhat split, with approximately 29% expressing disagreement and 34% expressing agreement, see Figure 2a. However, when asked whether they would focus more on programming if it were included in the exam, 49.3% selected that they would to a little extent, and 21.4% to a great extent. In contrast, if programming were not assessed, 22.3% indicated they would focus on it less, and 6% to a much lesser extent, see Figure 2b.

Figure 2. a. and b. Teachers' views on exam and its effect on their teaching



C1: Summary and interpretation

Taken together, these findings suggest that teachers' conceptions of programming in mathematics education are strongly shaped by the curriculum and the exam. While some see potential in new approaches, programming is often perceived as an additional and demanding component in an already full curriculum. Several teachers expressed a desire to balance curriculum mandates with professional autonomy, while others voiced concern that programming may displace core mathematical content. The presence or absence of programming in national assessments significantly influences teacher practices. For some, programming remains a concept they are still trying to make sense of. Teachers' conceptions of programming as a curricular requirement reflect how institutional expectations shape their internal prioritisation and pedagogical decisions.

Conception 2 (C2): Programming is a useful tool in mathematics education; however, it is time consuming and challenging to teach

Norwegian mathematics teachers in this study frequently described programming as a tool in mathematics education – a digital aid comparable to calculators, spreadsheets, CAS, or DGS. At the same time, they expressed concerns about the time required for preparation, teaching, and learning; the resources needed throughout the process; and the level of programming knowledge pupils must attain to make autonomous and appropriate decisions about when and how to use programming effectively.

Henry illustrated this perspective of programming as a mathematical aid:

> "I think that programming is a tool you can use to solve a problem in the same way that spreadsheet is. In the same way that knowing a formula for calculating the area of a circle is a tool. So, you can understand this tool, how it works. It is an advantage" (H28:48).

However, he also noted that programming differs from other tools in its complexity and the demands it places on both teachers and pupils. Henry explained:

> "When pupils work in a DGS, they need quite a long time to learn this tool. … It is in a way, if not self-explanatory then you can search around a bit, right? … But it is probably a bit, maybe a bit stricter [with programming] if you know what I mean. There is less room for error" (H22:26).

Lisa echoed this concern, suggesting that programming can feel less integrated with mathematical thinking:

> "I think CAS – a robust calculator with extensive options – I would say it is more functional than the programming tool, to be honest" (L20:28).
> "With programming, it is like you sit in a booth/bubble, and then you just type, and then you test whether it works, and then either it works or it does not. So, it is not very mathematical. It becomes a separate goal" (L20:45).

Nora emphasised the importance of contextual integration, warning that programming risks becoming isolated if not meaningfully connected to mathematical content:

> "I feel that it is natural to include programming where it fits in the subject" (N05:07). "... I feel that the pupils should understand the principles with the mathematical topic. If not, the programming becomes just another thing they need to do" (N05:58).

These reflections highlight the ambivalence many teachers feel. While programming is seen as a potentially valuable tool, it raises different challenges than other digital tools.

Another recurring concern was time pressure. Programming, while potentially beneficial, was often described as difficult to fit into an already full curriculum. Anne noted:

> "To get the greatest possible outcome, one may have to work with it [programming] all the time ... then we have really spent so much time that we have to move on to a new topic" (A23:23).

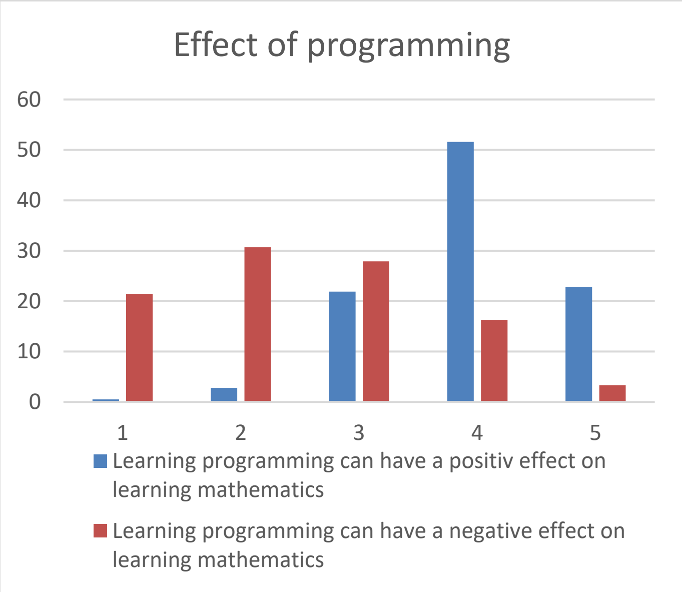
Lisa also expressed her view about this:

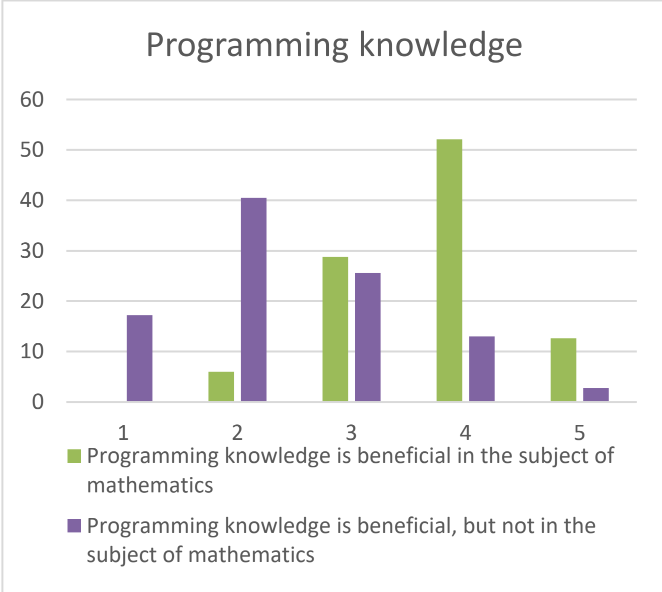
> "... there is quite a lot to learn in addition to the book, ... and then the timetable breaks down" (L24:04).

These qualitative insights are supported by the questionnaire data. A majority of teachers, 74.2% (to a little and great extent) believe programming has a positive impact on mathematics learning, while only 3.3% disagree or strongly disagree. However, 19.6% (to a little and great extent) support the opposite view, suggesting that programming may negatively affect mathematics learning, see Figure 3a.

A related item explored whether programming knowledge is beneficial within mathematics, or outside of it, see Figure 3b. While 64,7% of teachers (to a little and great extent) affirmed its value in mathematics, 15,8% (to a little and great extent) agreed that programming is beneficial, but not in the context of mathematics. Since both items were positively framed, this overlap suggests that many teachers see programming as a valuable competence, though not one that is inherently mathematical.

Figure 3. a. and b. Teachers' views on effects and benefits of programming

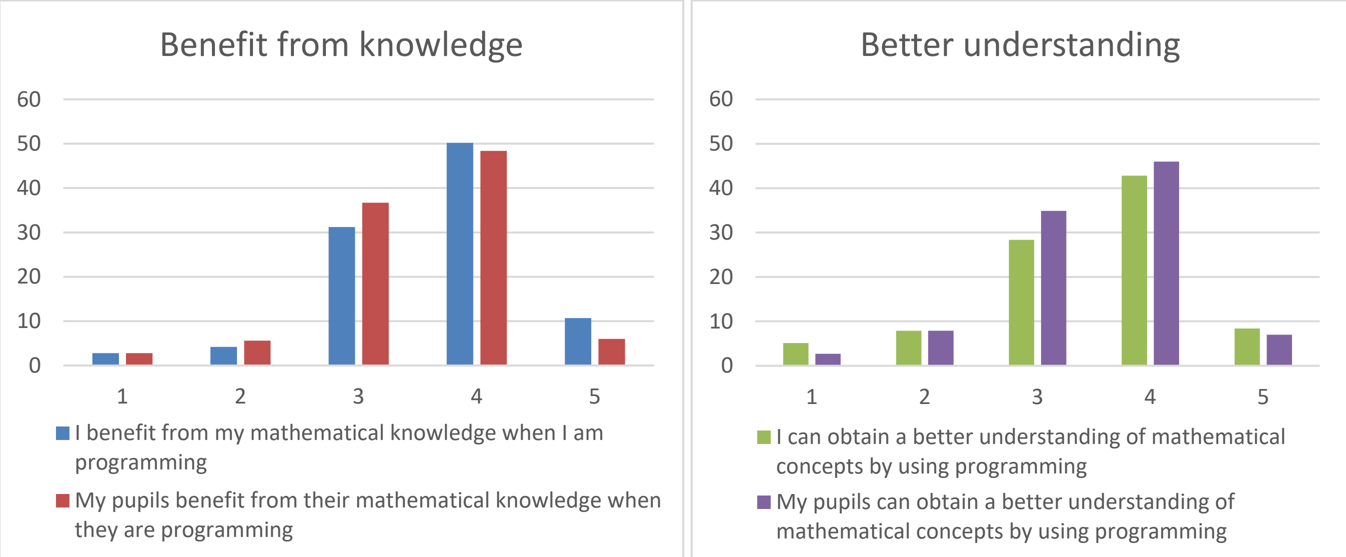




These nuanced views are further reflected in how teachers evaluate the relationship between programming and mathematical knowledge—both for themselves and for their pupils. In one questionnaire item, teachers were asked whether mathematical knowledge supports programming, see Figure 4a. Here, 60.9% of teachers agreed (to a little and great extent) that they personally benefit from their mathematical knowledge when programming, while 54.4% said the same about their pupils. Notably, more teachers selected the highest category "to a great extent" for themselves (10.7%) than for their pupils (6%), suggesting a perception that teachers draw more directly on mathematical knowledge when programming than pupils do.

A similar pattern appears when asked whether programming supports understanding of mathematical concepts, see Figure 4b. Again, teachers rated the benefit slightly higher for themselves than for their pupils: 51.2% of teachers agreed (to a little and great extent) that programming enhances their own understanding, compared to 53% for pupils. While the difference is small, the distribution still suggests that teachers may perceive programming as more beneficial for their own conceptual development than for their pupils' learning. Moreover, more teachers believed pupils benefit only "to a little extent" (46%) than believed so about themselves (42.8%).

Figure 4. a. and b. Teachers' views on the difference between them and their pupils



Together, these findings point to a nuanced and multifaceted perspective, where programming is recognized both as a support for mathematics teaching and a broader skill relevant across disciplines. This tension between potential and practicality is recurring in this theme. Lisa's reflection captures this duality particularly well:

> "We have got one more 'horse' to manage, but like this visual programming it is a really enjoyable activity, and there is mathematics in it as well. … I do not think it helps mathematics that we have got programming, but mathematics helps a lot, if you are to learn programming" (L22:41).

C2: Summary and interpretation

Taken together, these findings suggest that teachers' conceptions of programming in mathematics education are grounded in the view of programming is a potentially valuable tool. However, it requires considerable time, effort and pedagogical planning. Compared to familiar tools like calculators or DGS, programming is seen as less intuitive and more demanding. While teachers acknowledge programming's potential, some also question its mathematical relevance, describing it as a separate goal rather than an integrated aid. Teachers seem to find programming intellectually enriching for themselves but challenging to implement effectively for pupils – particularly given curriculum constraints and varying levels of pupil readiness. Its role in mathematics education remains contingent on context, integration, and purpose. Teachers' conceptions of programming are based on conceptualising it as a tool; however, its perceived complexity challenges their internal frameworks for integrating it meaningfully into mathematics.

Conception 3 (C3): Teachers feel unprepared, unsupported and insecure in teaching programming in mathematics education

Norwegian mathematics teachers in this study acknowledge that programming is now a part of the mathematics curriculum. Many expressed a genuine interest in programming, while also pointing to a persistent tension between their enthusiasm and the practical challenges of teaching it. This tension is shaped by limited institutional support, insufficient professional development, and a lack of clear pedagogical guidance.

During interviews, the teachers offered varied and sometimes contrasting perspectives, shaped by a range of factors influencing their ability to teach programming. Nora spoke of the excitement when programming works as intended, though also noting the time-consuming and unpredictable nature of classroom implementation. Anne expressed difficulties in translating her knowledge into teaching, especially when feeling uncertain about her understanding or instructional decisions. Lisa emphasised the importance of helping pupils connect programming with broader mathematical concepts. Henry reflected on the challenges of adapting to curriculum changes and called for greater collegial support during this transitional phase.

Lisa highlighted a common challenge: pupils may be able to write code, but often lack understanding of how programming connects to other areas of mathematics:

> “It is expected that one should be able to understand programming. The pupils can do it, but they do not understand why and what connections it has across the other mathematical topics” (L18:51).

Anne offered a more introspective view, describing how she enjoys experimenting with programming on her own but struggles to translate that into structured teaching:

> “I realise that I need a very A-B-C [approach]. I think it is a lot of fun to try and fail [with programming], so I spend a lot of time trying things out myself, but I have not yet been able to translate this into a lesson” (A10:10).

Henry framed the current situation as a transitional phase, calling for collegial support and patience:

> “This is a transitional phase where we have to be a little generous with each other ... Then we are talking about the teachers, because being in a classroom and teaching programming and then feeling like you are not in control is no fun” (H04:35b).

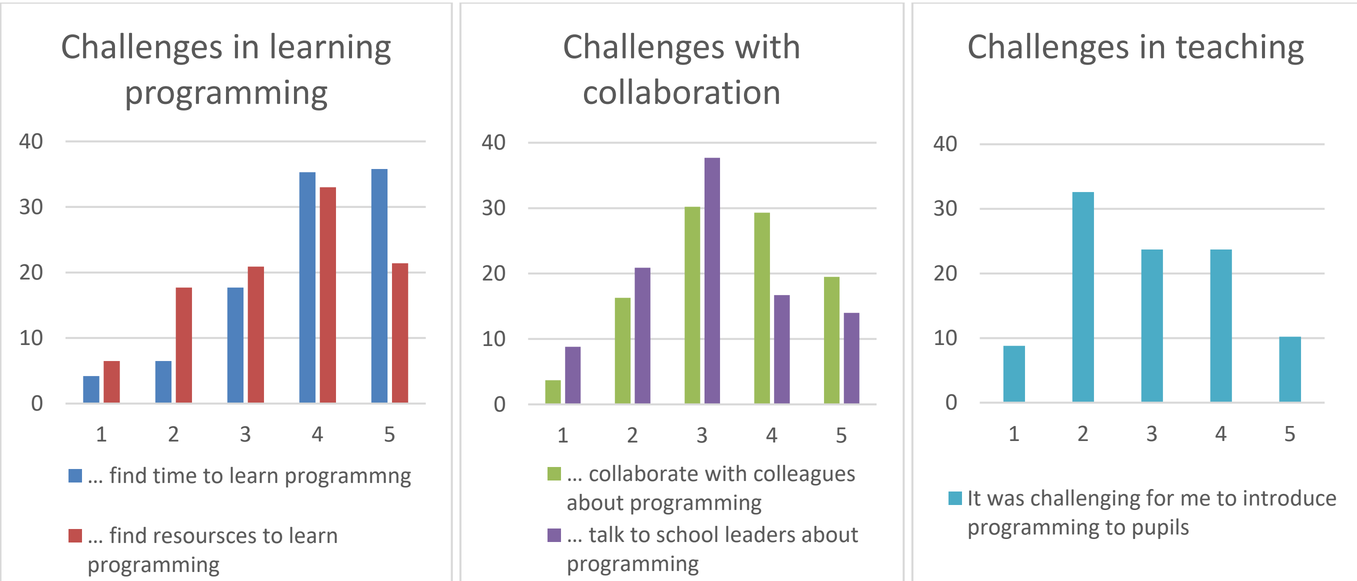
He also acknowledged that programming is perceived as both new and demanding, and that teachers respond differently to this ‘newness’, noting that change is not always easy or welcome.

These reflections point to a broader theme of vulnerability. Teachers are motivated to engage with programming but often lack the resources, time and confidence to do so effectively. The questionnaire mirrored this picture. When asked about specific challenges, the majority of teachers identified time as the most significant barrier. Specifically, 35,3% of participants indicated that finding time to learn was challenging to a little extent, and 35,8% selected “to a great extent”. In comparison, 33% and 21,4% of teachers reported similar levels of difficulty in finding resources, see Figure 5a.

Collaboration emerged as another area of concern. While some teachers, like Lisa, described strong collaboration within their schools, others – such as Nora and Henry – reported limited peer support and inconsistent practices across schools. Questionnaire responses reflected this ambivalence: the most common response to collaboration with colleagues and school leaders was neutral (30,2% and 37,7%, respectively), with moderate challenges reported slightly more often for colleagues than leaders, see Figure 5b.

Interestingly, introducing programming to pupils was perceived as less problematic. Most teachers selected lower challenge categories for this item, suggesting that while implementation is not without difficulties, it is generally manageable, see Figure 5c.

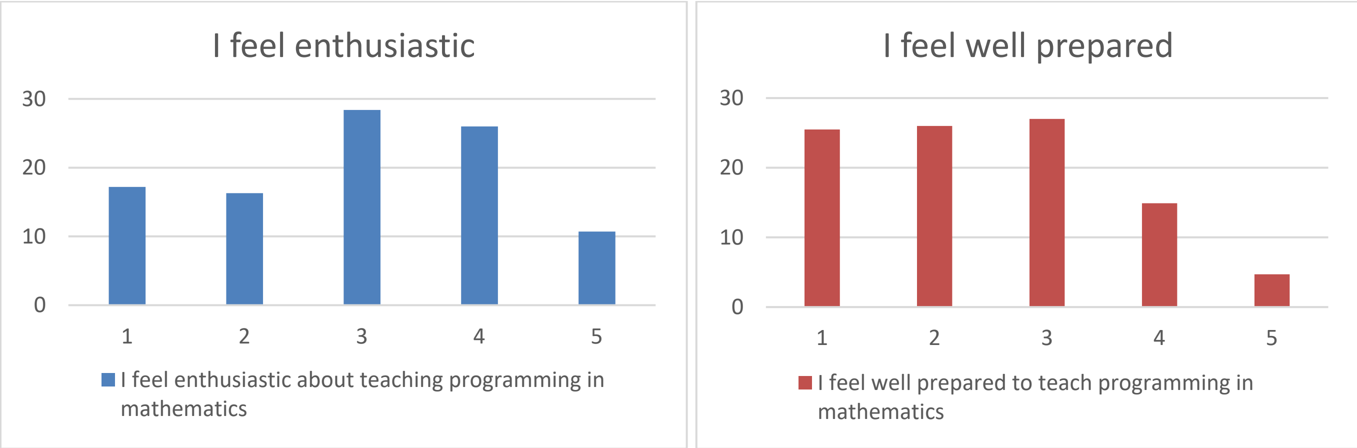
Figure 5. a., b. and c. Teachers’ views on “It was challenging for me to …”



A deeper tension emerged between teachers’ enthusiasm and their sense of preparedness. While 36,7% of teachers expressed enthusiasm (to a little and great extent), only 19,6% felt similarly prepared. In contrast, over half of the participants (51,5%) reported feeling either not at all or only slightly prepared, compared to 33,5% who expressed similarly low levels of enthusiasm, see Figures 6a and 6b. Nora also commented it rather clearly that it is about:

> “… lack of interest … interest and time, because you kind of have to familiarize yourself with programming too” (N31:42)

Figure 6. a. and b. Teachers' views on enthusiasm and preparedness



Henry described how he rationalized this tension about the professional and personal aspects of learning programming:

> "[What] is demanding when you do not quite have control yourself is when the pupils make mistakes, because they make mistakes all the time, … they did not use a colon, they did not use an indent. … And having worked so much with programming, having made all these mistakes so many times yourself, you know what to look for. That is really important to have programmed a lot yourself to have made these mistakes yourself" (H10:52).

C3: Summary and interpretation

Taken together, these findings suggest that teachers' conceptions of programming in mathematics education are formed by feelings of uncertainty and unpreparedness, often due to limited knowledge, time constraints, challenges in adaptability, and in some cases insufficient collegial support. Teachers feel expected to demonstrate control and expertise; however, they are still required to implement programming, even when they do not feel prepared to do so. While some remain motivated or enthusiastic, concerns about practical implementation of programming in mathematics education persist. This discrepancy suggests that, although teachers may be emotionally open to programming, they do not feel equipped to teach it confidently. This highlights a broader sense of professional uncertainty and underscores the need for targeted support and professional development. Teachers' conceptions of programming reflect their sense of vulnerability arising from a mismatch between external demands and their internal confidence and preparedness.

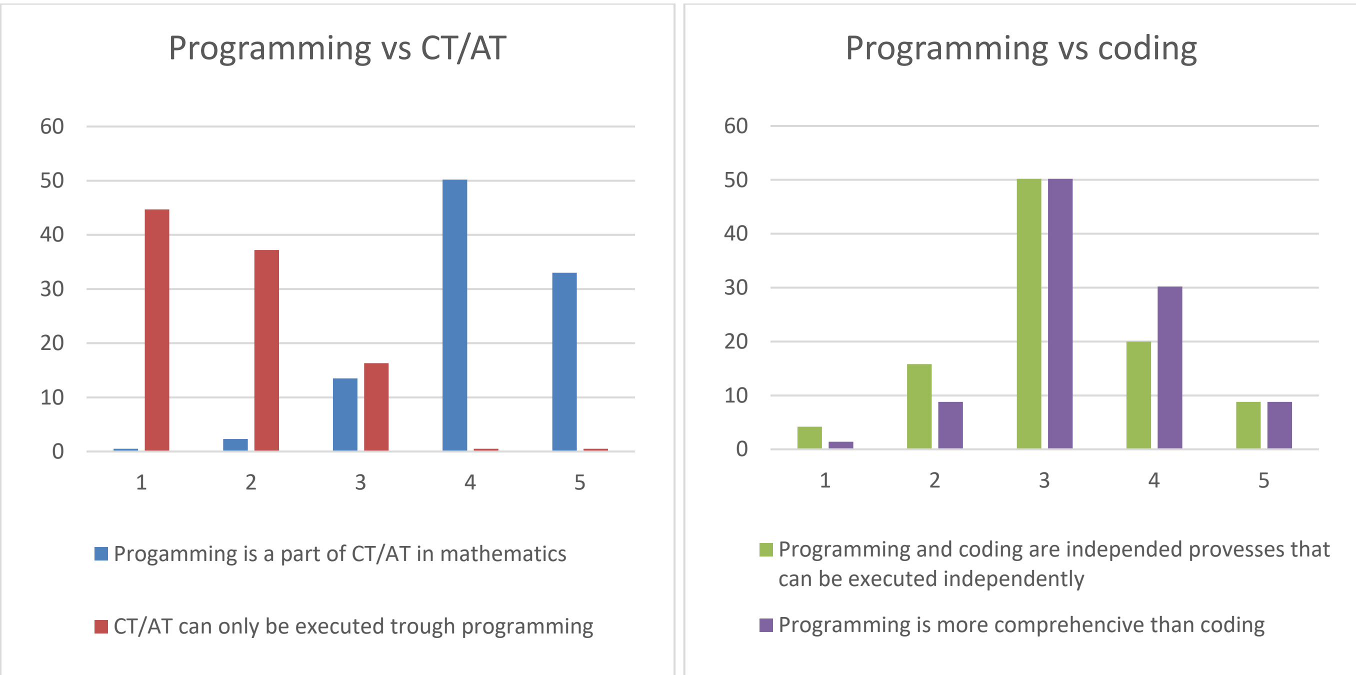
Conception 4 (C4): Inconsistent terminology reflects a lack of an appropriate professional discourse about programming in mathematics education

Norwegian mathematics teachers in this study revealed inconsistencies in how they use and interpret terminology related to programming in mathematics education. While there appears to be a general understanding of which words to use, teachers' approaches, explanations and vocabulary varies. Some teachers seem to use a more general or abstract terminology with relatively consistent interpretations, as seen through uniformity in the items in the questionnaire, whereas others take a more practical approach or elaborate with more details, offering individual explanations where examples and usage vary from one another, as seen in the interview data.

The questionnaire data indicate that teachers broadly agree on some conceptual relationships. For example, the majority affirmed that programming is part of CT/AT: 50,2% to a little extent and 33% to a great extent. Only 13,5% selected the neutral category, and disagreement was minimal (0,5% and 2,3%). However, when asked whether CT/AT must be implemented through programming, responses reversed: 44,7% selected "not at all" and 37,2% "not a little", with only 1% agreeing (0,5% and 0,5%), see Figure 7a. This pattern suggests that while programming is seen as related to CT/AT, it is not viewed as the sole method for engaging with these forms of thinking.

Further ambiguity emerged around the terms "programming" and "coding". When asked whether these are independent processes or whether programming is more comprehensive than coding: 50,2% of teachers selected the neutral category for both statements. While a slight majority leaned toward viewing programming as more comprehensive (30,2 % to a little extent and 8,8% to a great extent), the overall distribution suggests uncertainty or reluctance to take a firm stance, see Figure 7b.

Figure 7. a. and b. Teachers' views on terms and relations between them



Interview data further illustrate this conceptual ambiguity. Nora, when asked to relate algorithmic thinking in core elements and programming in competency aims in the mathematical curriculum, she explained:

> "It is really the same thing, because if you do not follow the algorithm, you will not get it right, ... You have to do this, in a very specific way (N18:32).

However, she later emphasised flexibility of for multiple valid approaches:

> "... there are different ways of arriving at the right answer and that one way is not necessarily more wrong than the other (N19:14).

Lisa echoed this view, describing how programming and CT/AT can be used together to verify solutions. Yet she also questioned the clarity of the term "algorithm", suggesting it may function more as educational jargon than as shared conceptual tool:

> "... algorithm is just the order of things, is it not? ... I think that algorithm is a bit of a difficult word both in mathematics and programming, ... a very commonly used word that not everyone understands. They just say it, it sounds good to say it, but do we understand what it is?" (L16:35a).

Henry also elaborated the importance of CT/AT in programming but expressed uncertainty about how to introduce the term to pupils:

> "We have not really talked much about algorithmic thinking [CT/AT] with that word in relation to pupils, ... In a way, you do not think very algorithmically, but it is when you have slightly more complicated things to figure out that you kind of have to start thinking algorithmically" (H27:00).

C4: Summary and interpretation

Taken together, these findings suggest that teachers' conceptions of programming in mathematics education are expressed in a variety of ways. Teachers seem to use terms programming, coding, CT/AT and algorithms inconsistently. This variation may imply a lack of a coherent professional terminology around programming in mathematics education or limited collaborations and discussions between teachers themselves. Some teachers associate programming more with mathematical algorithms (e.g., standard-algorithm for multiplication) or orderly processes, than with coding practices, indicating different interpretations of its role and purpose. Together, these findings suggest that while teachers recognize the relevance of programming and CT/AT in mathematics education, they lack a consistent vocabulary and conceptual framework for discussing and teaching these ideas. The absence of a shared professional discourse may contribute to uncertainty and inconsistency in implementation, particularly when it comes to aligning expectations across schools and grade levels. Teachers' conceptions of programming address the inconsistent terminology, framing it not only as curricular ambiguity but also as a reflection of divergent internal understandings of programming and CT/AT.

6. Discussion and concluding remarks

This discussion section interprets the findings in relation to the study's conceptual framework, highlighting how teachers' conceptions of programming in mathematics education reflect tensions between curricular demands, pedagogical practice, and conceptual clarity.

This study investigated the research question: *What conceptions do Norwegian mathematics teachers reveal regarding the use of programming in teaching and learning mathematics?* Drawing on interviews and a contextualising questionnaire, four key conceptions were identified:

- C1) Programming is shaped by the curriculum and the exam
- C2) Programming is a valuable but demanding tool
- C3) Teachers feel unprepared and uncertain
- C4) There appears to be a lack of clear terminology and professional discourse among teachers in primary and lower secondary education

These conceptions illustrate how teachers make sense of programming within their pedagogical practice, encompassing their motivations, perceived challenges, and alignment with curricular demands. The findings suggest that teachers' conceptions are not formed in isolation but emerge through a dynamic interplay between internal cognitive structures and external institutional factors. Recognizing this interaction is crucial for designing professional development and curricular support.

According to Papert (1980), programming enables pupils to construct their knowledge actively and creatively, supported by powerful machines. While Norwegian curriculum does not provide a detailed rationale for including programming, its emphasis on pupil exploration and digital skills reflects similar ideas. The curriculum and the exam play a central role in guiding the implementation and place expectations on teachers to adapt their practice accordingly (C1). However, as noted by teachers themselves and echoed in Nordby et al. (2022), the curriculum offers limited guidance on how programming should be taught in mathematics education.

Despite limited formal training, many teachers reported teaching programming regularly (see Section 5.1). This reflects a strong sense of professional responsibility, and, alongside ambivalence toward assessment, underscores programming's ambiguous position in the curriculum (C1). Teachers' willingness to prioritise programming appears closely tied to whether it is formally assessed.

In contrast to this institutional framing, many teachers also recognize programming's potential as a pedagogical tool (C2). Some questioned its mathematical relevance, describing it as a parallel activity rather than an integrated part of mathematics. This perception aligns with the lower levels of integration described by Broley et al. (2018), where programming is treated as a separate or technical task rather than a tool for mathematical reasoning. Such views may hinder its pedagogical potential unless explicitly connected to mathematical concepts.

Interestingly, teachers in this study perceived programming as slightly more beneficial for their own conceptual development than for their pupils' (C2). This may reflect a general positivity towards programming, but also a lack of clarity about how it connects to specific mathematical topics. Similar patterns are observed in other studies: Kaufmann et al. (2022) highlight the challenge of aligning programming with subject-specific learning goals, while Refvik and Opsal (2023) show that even when programming is introduced in mathematics context, its mathematical relevance is not always evident to pupils or teachers. Turgut et al. (2024) further emphasise that without explicit links to mathematical content, programming risks being perceived as an add-on rather than an integral part of mathematics education. A similar tendency was reported by Misfeldt et al. (2019), where Swedish teachers also viewed programming as more beneficial for their own understanding than for pupils' learning. These findings resonate with the present study, where teachers' conceptions often reflect a general enthusiasm for programming, but limited articulation of its mathematical purpose.

This duality between top-down expectations of the curriculum (C1) and bottom-up pedagogical practice (C2) reflects broader tensions between compliance and meaningful integration. It aligns with the four levels of integration described by Misfeldt et al. (2020) and (Kilhamn et al., 2021b), ranging from programming as a separate activity to a mediating instrument in mathematical investigations. Kilhamn et al. (2021a) argue that teachers often adopt the curriculum by focusing on the tool itself and acknowledging its general benefits. In line with this, this study found that teachers tend to align closely with curriculum descriptions, viewing programming primarily as a tool within those parameters. Teachers' conceptions also suggest that, similar to findings from other studies on digital tools in Norway, programming remains a time-consuming and demanding tool in mathematics education (Meaney et al., 2023; Turgut et al., 2024).

The demanding nature of programming does not support teachers' confidence or preparedness to teach it (C3). As shown in the questionnaire, over half of the participating teachers reported feeling unprepared, indicating a lack of confidence in addressing classroom challenges related to programming (C3). This finding aligns with Misfeldt et al. (2019) and Johansson et al. (2023), who reported similar concerns among Swedish teachers. However, Norwegian teachers in this study appeared somewhat more positive, similar to pre-service teachers (Kaufmann et al., 2022), which may reflect differences in national implementation timelines or evolving professional attitudes.

Despite acknowledging programming's value (C2) and curricular demands (C1), many teachers in this study feel underprepared and unsupported (C3), highlighting a gap between professional expectations and available resources. This aligns with Rolandsson's (2013) and Dilling et al.'s (2024) observations that curriculum reforms introducing programming often leave teachers uncertain about what to teach and how to teach it. While these studies focus on different contexts, they similarly point to lack of clarity and support structures. Turgut et al. (2024) also note that although teachers frequently report challenges with programming, they rarely seek collegial support – suggesting that professional communities around programming remain underdeveloped, which might lead to inconsistent terminology (C4). These findings resonate with the present study, where teachers express a strong sense of responsibility but limited access to collaborative or institutional support.

This tension between curriculum mandates (C1) and teachers' sense of preparedness (C3) reflects a broader challenge in implementing programming reforms. As Broley et al. (2018) also found, without structured support, teachers may struggle to translate curricular goals into effective classroom practice. Tossavainen et al. (2024) similarly note that integrating programming as a mediator of mathematical thinking imposes higher demands. While it can enrich mathematical activity, it also raises the bar for aligning curricular expectations with classroom practice.

Norwegian mathematics teachers also report that terms relating to programming are vague and often understood differently (Nordby et al., 2022). The lack of a shared professional discourse (C4) may exacerbate feelings of insecurity (C3), as teachers struggle to articulate and align their conceptions of programming and CT/AT, or between the broad and narrow view of programming (Bergqvist, 2021). This suggests a need for clearer conceptual frameworks and professional dialogue at primary and lower secondary level. Terminological difficulties appear particularly pronounced in the Norwegian context, partly due to translation issues – such as the rendering of "computational thinking" as "algorithmic thinking" (Nordby et al., 2022). While teachers express a strong sense of professional obligation to follow the curriculum and prepare pupils for assessment (C1), this institutional alignment does not seem to alleviate their conceptual uncertainty (C4). Instead, the lack of definitional clarity may exacerbate confusion, suggesting a disconnect between curricular mandates and teachers' understanding.

Building on these findings, this final section considers implications for practice and research. Teachers in this study recognise potential benefits of programming in mathematics education, but the extent to which these benefits are realised in practice remains unclear. The specific integration of programming into classroom practice, and the relationship between teachers' conceptions and their teaching decisions, require further investigation. Broley et al. (2018) note that learners engage with programming at varying levels of depth and confidence, and that collaborative professional development can help bridge the gap between curriculum expectations and classroom realities. Similarly, Misfeldt et al. (2020) and Kilhamn et al. (2021b) emphasise the importance of understanding how programming practices intersect with mathematical learning.

These four conceptions underscore the need for targeted professional development, clearer curricular guidance, and a shared professional language around programming in mathematics education. While the sample size was limited, the findings offer valuable insights into the complexities teachers face when implementing programming. Future research should examine how these conceptions influence classroom practice and pupil learning outcomes, particularly in relation to curriculum alignment, teacher agency and pedagogical decision-making. A deeper understanding of how teachers navigate the intersection of programming and mathematics is therefore essential for supporting integration efforts that are both pedagogically meaningful and aligned with curricular expectations.

Research ethics

Funding

This project did not receive any external funding.

Institutional review board statement

The ethical principles of research have been followed. Informed consent was obtained. Project has been registered and approved by SIKT, the Norwegian Council of Ethics.

Informed consent statement

Informed consent was obtained from all research participants.

Data availability statement

Those interested in the data can contact the author for further information.

Acknowledgements

Thanks to all teachers who participated and made it possible to carry out this research.

Conflicts of interest

The author declares no conflicts of interest.

References

Bergqvist, E. (2021). An inquiry of different interpretations of programming in conjunction with mathematics teaching. Norma 20, The ninth Nordic Conference on Mathematics Education, Oslo, Norway, 1–4 June 2021.

Bingolbali, E., & Monaghan, J. (2008). Concept image revisited. *Educational studies in Mathematics*, *68*(1), 19–35. https://doi.org/https://doi.org/10.1007/s10649-007-9112-2

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M. A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V., & Stupurienė, G. (2022). *Reviewing computational thinking in compulsory education: State of play and practices from computing education*. https://doi.org/10.2760/126955

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), 77–101. https://doi.org/https://doi.org/10.1191/1478088706qp063oa

Broley, L., Caron, F., & Saint-Aubin, Y. (2018). Levels of programming in mathematical research and university mathematics education. *International Journal of Research in Undergraduate Mathematics Education*, *4*(1), 38–55. https://doi.org/https://doi.org/10.1007/s40753-017-0066-1

Clark, T., Foster, L., Sloan, L., & Bryman, A. (2021). *Bryman's social research methods*. Oxford University Press.

Denning, P. J., & Tedre, M. (2019). *Computational thinking*. Mit Press.

Dilling, F., Köster, J., & Vogler, A. (2024). Beliefs of Undergraduate Mathematics Education Students in a Teacher Education Program about Visual Programming in Mathematics Classes. *International Journal of Research in Undergraduate Mathematics Education*, *10*(3), 700–731. https://doi.org/https://doi.org/10.1007/s40753-024-00248-0

Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? Proceedings of the 9th Workshop In Primary and Secondary Computing Education (WIPSCE 2014),

Fojcik, M. K. (2022). Perspectives and challenges on programming as a tool to learn mathematics. Proceedings of the Twelfth Congress of the European Society for Research in Mathematics Education (CERME12),

Furinghetti, F., & Pehkonen, E. (2002). Rethinking characterizations of beliefs. In G. C. Leder, E. Pehkonen, & G. Törner (Eds.), *Beliefs: A hidden variable in mathematics education?* (pp. 39–57). Springer Dordrecht. https://doi.org/https://doi.org/10.1007/0-306-47958-3

Johansson, C., Juhlin, A., Tossavainen, T., & Wedestig, A. (2023). Nyfikenhet och tillräcklighet. Gymnasielärares erfarenheter av att undervisa matematik med programmering. *Nämnaren*, *49*(3), 35–41. https://ncm.gu.se/wp-content/uploads/2024/09/3541_23_3.pdf

Kaufmann, O. T., Maugesten, M., & Meaney, T. J. (2022). Views of pre-service teachers in Norway on the value of programming in teaching mathematical and pedagogical topics. Proceedings of the 13th ERME Topic Conference on Mathematics Education in Digital Age (MEDA3) (ETC13),

Kilhamn, C., Bråting, K., & Rolandsson, L. (2021a). Teachers' arguments for including programming in mathematics education. NORMA 20, The ninth Nordic Conference on Mathematics Education, Oslo, Norway, 1–4 June 2021.

Kilhamn, C., Rolandsson, L., & Bråting, K. (2021b). Programmering i svensk skolmatematik: Programming in Swedish school mathematics. *LUMAT: International Journal on Math, Science and Technology Education*, *9*(1), 283–312. https://doi.org/https://doi.org/10.31129/LUMAT.9.2.1457

Meaney, T. J., Huru, H. L., & Kvivesen, M. (2023). Preservice and inservice teachers' views on digital tools for diverse learners in mathematics education. *Nordic Studies in Mathematics Education*, *28*(3-4), 103–123. https://hdl.handle.net/10037/33305

Ministry of Education and Research. (2019). *Curriculum for Mathematics year 1-10 (MAT01-05).* Retrieved from https://data.udir.no/kl06/v201906/laereplaner-lk20/MAT01-05.pdf?lang=eng

Misfeldt, M., Jankvist, U. T., Geraniou, E., & Bråting, K. (2020). Relations between mathematics and programming in school: Juxtaposing three different cases. Proceedings of the 10th ERME Topic Conference on Mathematics Education in the Digital Age (MEDA 2020),

Misfeldt, M., Szabo, A., & Helenius, O. (2019). Surveying teachers' conception of programming as a mathematics topic following the implementation of a new mathematics curriculum. Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education (CERME11),

Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022). Primary mathematics teachers' understanding of computational thinking. *KI-Künstliche Intelligenz*, *36*(1), 35–46. https://doi.org/https://doi.org/10.1007/s40751-022-00102-5

Opre, D. (2015). Teachers' conceptions of assessment. *Procedia-Social and Behavioral Sciences*, *209*, 229–233. https://doi.org/https://doi.org/10.1016/j.sbspro.2015.11.222

Papert, S. A. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic books.

Philipp, R. A. (2007). Mathematics teachers' beliefs and affect. In F. K. Lester (Ed.), *Second handbook of research on mathematics teaching and learning* (Vol. 1, pp. 257–315). https://www.researchgate.net/profile/Randolph-

Philipp/publication/284039669_Mathematics_teachers%27_beliefs_and_affect/links/674214f5868c966b93271978/Mathematics-teachers-beliefs-and-affect.pdf
Refvik, K. A. S., & Opsal, H. (2023). Do optional programming courses affect eighth-grade students' mathematical problem solving? *Computers in the Schools*, *40*(3), 244–261. https://doi.org/https://doi.org/10.1080/07380569.2023.2175634
Rolandsson, L. (2013). Teachers' Beliefs Regarding Programming Education. In *Technology Teachers as Researchers: Philosophical and Empirical Technology Education Studies in the Swedish TUFF Research School* (pp. 285–309). Sense Publishers. https://doi.org/https://doi.org/10.1007/978-94-6209-443-7_13
Tall, D., & Vinner, S. (1981). Concept image and concept definition in mathematics with particular reference to limits and continuity. *Educational studies in Mathematics*, *12*(2), 151–169. https://doi.org/https://doi.org/10.1007/BF00305619
Tossavainen, T., Johansson, C., Juhlin, A., & Wedestig, A. (2024). Programming as a mediator of mathematical thinking: Examples from upper secondary students exploring the definite integral. *LUMAT – International Journal on Math, Science and Technology Education*, *12*(3), 78–99. https://www.diva-portal.org/smash/get/diva2:1858392/FULLTEXT01.pdf
Turgut, M., Kohanová, I., & Gjøvik, Ø. (2024). Developing survey-based measures of mathematics teachers' pedagogical technology knowledge: a focus on computational thinking and programming tools. *Research in Mathematics Education*, 1–24. https://doi.org/https://doi.org/10.1080/14794802.2024.2401482
Vergnaud, G. (1998). Towards a cognitive theory of practice. In A. Sierpinska & J. Kilpatrick (Eds.), *Mathematics Education as a Research Domain: A Search for Identity: An ICMI Study Book 1* (pp. 227–240). Springer.
Vergnaud, G. (2009). The theory of conceptual fields. *Human development*, *52*(2), 83–94. https://doi.org/http://dx.doi.org/10.1159%2F000202727
Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, *18*(1), 1–25. https://doi.org/https://doi.org/10.1145/3089799