Students Undertaking an Elective Programming Course: Their Views on the Connection Between Programming and Mathematics

Kim André Stavenæs Refvik, Annette Hessen Bjerke, Morten Misfeldt

Students Undertaking an Elective Programming Course: Their Views on the Connection Between Programming and Mathematics

Kim André Stavenæs Refvik[1], Annette Hessen Bjerke[2] and Morten Misfeldt[3]

[1]Volda University College, Norway; refvikki@hivolda.no, [2]Oslo Metropolitan University, Norway; annette.hessen@oslomet.no, [3]University of Copenhagen; Denmark, misfeldt@ind.ku.dk

In this study, we investigate how students undertaking an elective programming course experience the connection between programming and mathematics. Based on stimulated recall interviews with six Grade 8 students who have solved test items from PISA 2003, we identify various experiences among the students regarding how they draw on programming skills in mathematics and vice versa. We propose that three findings, in particular, are worthy of further discussion and investigation: the way some experience programming as a context for mathematics and mathematics as a context for programming; the challenges that follow from how some students assert that programming and mathematics complement each other while others struggle to see a connection; and, perhaps the most important, the way in which the students report that experiences from programming have equipped them with skills such as being systematic and exact, which they find they need in mathematics.

Keywords: mathematics education, problem solving, computational thinking, programming

Introduction

After recognising computational thinking (CT) as a vital 21st century skill (Bocconi et al., 2018), several countries have recently introduced CT to their curricula, either as a separate programming subject or as part of existing subjects. In Norway, CT is introduced in mathematics across grades (Directorate of Education, 2020), adding to being an elective programming subject in lower secondary school (Grades 8–10, ages 12–16). In the new Norwegian curricula, after Grade 4 in primary school, CT is exclusively addressed and talked about as *programming*. In programming, CT entails the ability to develop programmes to perform different algorithms and is considered a systematic description of how to solve a problem using a specific approach (Kaufmann & Stenseth, 2021, p. 1030).

As several studies have asserted that programming can induce improved understanding in mathematics (Moreno León et al., 2016), and even more precise – improved problem-solving skills in mathematics (Husain et al., 2017; Sinclair & Patterson, 2018) – it is easy to understand why many European countries introduce programming in mathematics (Balanskat & Engelhardt, 2015). However, Kaufmann and Stenseth (2021) warned against assuming that the programming's effect on problem solving is automatic. Exploring these findings while considering Kaufmann and Stenseth's (2021) warning, we seek to better understand whether those students undertaking an elective programming course in lower secondary school can communicate how mathematics and programming are interrelated. Hence, we give students a voice in the matter. It is imperative that the students themselves experience this connection. Hence, we ask the following research question: *How and to what extent do students undertaking an elective programming course experience a connection between programming and mathematics?*

Theoretical Framework

To investigate the interplay between programming and mathematics, we turned to Kilhamn et al. (2021) and Elicer and Tamborg (2021). Kilhamn et al. (2021) investigated the relationship between mathematics and programming in 32 mathematics lessons and found four 'categories' of relationships: *only programming* (lessons only focusing on programming); *mathematics as context for programming* (lessons where no new mathematical concepts are used, but programming can be used to repeat or confirm mathematical knowledge); *programming as a computational tool in mathematics* (lessons that clearly have mathematical content, and programming is used to carry out calculations efficiently); and *programming as a tool for exploring mathematical concepts* (lessons where programming is used to explore mathematical concepts and relations, where programming adds new insights and develops one's mathematical competence). Similarly, Elicer and Tamborg (2021) analysed activities from the Danish 'Texforsøget' and described the role of mathematics and what they refer to as programming and computational thinking (PCT) in different activities. They found six categories, where the first two are *no mathematics involved* and *no PCT involved*, implying either no mathematics or no PCT involved in the tasks, and that the focus of the task is either purely mathematical or purely programming. The two following categories, *mathematics as a context* and *PCT as a context*, are categories where mathematical concepts are explored using PCT operations or where PCT concepts are explored using mathematical operations. The next category is *conceptual integration*, which are tasks that are "not solved with mathematical or PCT actions but involve concepts in both math and PCT" (Elicer & Tamborg, 2021, p. 5). The last category is *operational integration*, where "mathematical and PCT competencies are interdependent" (Elicer & Tamborg, 2021, p. 6).

While exploring what these two studies revealed on the relationship between programming and mathematics, both at the level of a lesson and at the level of the tasks given, we sought to see how a discussion around some problem-solving tasks can work as a starting point for investigating how students see the relationship between programming and mathematics. Problem-solving tasks were a natural point of departure, as research seems to unanimously assert that problem solving is the most obvious overlap between mathematics and programming. For instance, in pursuing the definition of CT for mathematics and science classrooms, Weintrop et al. (2016) proposed a taxonomy comprising four practices, one of which focuses on computational problem-solving practices. In this practice, Weintrop et al. (2016) identified problem solving as a central practice for developing CT in mathematics (and science), which sometimes takes the form of conventional programming (p. 138). This made us take a set of mathematics problem-solving tasks as our point of departure when interviewing students on how and if they see a connection between programming and mathematics.

Methods

In Norway, students choose one elective subject in each of their three years in lower secondary school. This study used data collected from a larger project, where all participants (N = 247) completed a test comprising five different problem-solving tasks from the PISA 2003 test (these were the tasks on the library system, course design, transit system, irrigation and energy needs; see Faculty of Educational

Sciences UiO [2003] for details). Due to how the library system task was mentioned and discussed in most interviews, we give a brief account of it here to inform the reader:

> The task starts by giving a flowchart that shows a library lending system: If you are a teacher (yes), you are allowed to lend books for 28 days; if you are not (no), you are allowed to lend books for 7 days. Based on this system, the students are asked to build more complicated flowcharts.

Participants selected for this subproject are six Grade 8 students (ages 13–14) who chose programming as their elective subject (hereafter referred to as the *elective programming course*). The participants (five girls, one boy) came from four different schools (a mix of city-centre and rural schools) in four different counties in the western and southern parts of Norway and were chosen for interviews based on a) their choice of elective, b) their answers on the initial test, and c) how their methods on the initial test stand out from the rest of the students. Also, they were selected to ensure distribution across schools. The names presented later in the Results section are fictional.

Each of the six participants was interviewed by the first author of this paper on the same day as they had completed the initial PISA-2003-based test. The semi-structured interviews lasted for about 15 minutes (between 11–28 minutes) and were recorded. The overall intention of the semi-structured interviews was to explore the students' accounts of their solutions and to see if there were any cross-references between their experiences in mathematics and the elective programming course. The first author used stimulated recall when asking questions about what knowledge they drew on when solving the five selected problems. Additionally, the participants were asked if they had examples of when they had drawn on knowledge learned in their elective programming course in their mathematics class and vice versa.

The interviews were transcribed in full in their original language (Norwegian) and analysed in two steps. Step one involved a process of consistent coding (Mason, 2017) that entailed reading and re-reading the transcript to identify if and how the interviewees related programming and mathematics and their accounts of how they drew on their programming knowledge in mathematics and vice versa. Step two was to identify the central experiences and understanding of the students, which led to two main categories in the results: programming in mathematics and mathematics in programming. These two categories also build on the categories where mathematics is the context for programming and programming is the context for mathematics from Kilhamn et al. (2021) and Elicer and Tamborg (2021).

Results

This results section comprises two parts. In the first part, we will consider how the 8th grade students saw the possibility of using methods, knowledge and skills from their elective programming course when solving problems in mathematics, while in the second part, we will focus on how students used methods, knowledge and skills from mathematics in their elective programming course.

Drawing on Elective Programming Skills in Mathematics

When talking about the library system task, three of the students explained how they recognised the need to use a flowchart, which they had been introduced to in their programming elective course.

Peter immediately recognised the flowchart and explained how he had used it as part of his programming process:

> I recognised this one immediately [points at the flowchart] because we have had many flowcharts in the elective programming course when making programmes in different programming languages. The first thing we have to do is make pseudocodes and flowcharts.

We see how Peter drew on knowledge from his programming elective course when solving the Library system task. This also applies to two other students, Andy and Nick, who recognised the flowcharts as a method from the elective programming course.

While the section above provides examples of how students draw on methods from their programming elective course when solving problems in mathematics, we also found that the programming elective course provided the students with a new vocabulary that came in handy in their mathematical problem solving. During interviews, the students revealed how they tended to use programming concepts to solve and explain *how* they solved the library system task. Nick and Michael used TRUE/FALSE statements or IF-ELSE statements to explain how they solved the library system task and related this problem to earlier experiences:

> (…) is it a magazine? If yes, seven days. If no, go to the next step. If you do not return books or magazines, then you are not eligible to borrow (…) (Nick)

Peter, however, described how he solved other problems during regular mathematics classes using the programming concept variable and loops:

> One time, I developed a programme in Trinket or Python. Then I used a counting variable. I got a task where I could use the counting variable to solve it. This is a formula I can use in the programme because I have learnt to calculate this way using variables and 'x in range' [a command for loops in Python].

The third finding is connected to how the students identified whether or how the elective programming course changed their mathematics problem-solving approach. Due to how they had learnt to work in the elective programming course, Robbie revealed that he made more use of diverse strategies and focused more on finding multiple solutions to mathematical problems, and both Nick and Peter said they saw themselves as working more systematically when solving mathematical problems. Adding to this, Nick talked about how he felt more accurate in solving mathematical problems since this is vital in the elective programming course:

> My mathematical view has become slightly more systematic. (…) In programming, one must be more exact to make a programme work.

Overall, these findings indicate that students acquire general skills through the elective programming course that they see as positive for their mathematical problem solving. Andy even highlighted that he sees a clear connection between programming and mathematics – he sees that both subjects are about solving problems:

> Programming is much about solving different problems in a code, and some of the tasks in mathematics are about the same: solving a problem.

However, some students did not see a connection between the elective programming course and mathematics. Nora, like Michael, claimed that there is no connection between programming and mathematics, adding that there is much she has not understood in the elective programming course. As she finds the elective programming course challenging, kind of softenings her statement on the disconnect. When asked if the elective programming course had affected him in mathematics, Michael put it like this:

> That is a good question; I do not think so since programming does not involve, or it involves numbers, of course, but I am not sure if it is under [has to do with] mathematics. (…) It is not addition or subtraction (…)

Adding to this and despite his vague idea that the elective programming course might have changed his mathematical competency, Robbie revealed that he had never experienced any programming in mathematics and therefore saw no apparent connection. However, he said that one must know mathematics to programme, which we will return to in the second part of the results, which we turn to now.

Drawing on Mathematics Skills in the Elective Programming Course

Our data revealed three ways in which the students reported how they drew on mathematics skills in their elective programming course. The first entails how they report on the use of mathematical concepts in programming tasks, such as figure numbers and geometry. Five of the students had experience with mathematics in programming, one of which is Nora. She explained that she had experienced using rotation and angles when programming how to make different figures move. Likewise, Peter connected experiences from algebra to the concept of variables in the elective programming course:

> There is much mathematics in programming because there is much algebra in programming. One uses many variables to make calculations, and that is almost the same as in algebra.

The second entails that students report on how they find their experiences of using different digital tools in mathematics to better understand what is going on in the elective programming course. Andy described an experience with GeoGebra, where he talked about making and using different functions to solve problems in mathematics. Similarly, Nick drew on an experience of using Excel and claimed this to be programming when he was asked if he used programming to solve mathematical problems: "Excel is technically programming". Both Andy's and Nick's experiences come from well-known computational tools in mathematics, and they reported that their experiences with those tools can help them in the elective programming course.

Third, our analysis shows that all the students, in one way or another, agreed that one needs to understand mathematics to programme. For example, Robbie, who mentioned the four arithmetic operations earlier, used arithmetic to argue that one needs to know them to develop a programme. Nick adds to this, saying "… one has to know much mathematics to programme properly". From this, we can see that Nick sees how mathematics plays a role in the elective programming course.

Discussion

In answering the research question – How and to what extent do students undertaking an elective programming course experience a connection between programming and mathematics? – we decided to investigate how students tend to draw on programming in mathematics, and vice versa. Due to how, for instance, Weintrop et al. (2016) identified problem solving as a central practice for developing CT (and conventional programming) in mathematics, we decided to use the students' solutions on a set of five PISA 2003 problem-solving tasks for knowing their thoughts on the connection between programming and mathematics.

Our analysis resulted in three main findings, where the first two confirm, and considerably elaborate on, the findings of Kilhamn et al. (2021) and Elicer and Tamborg (2021). First, we saw that our students reported programming as a context for mathematics and mathematics as a context for programming. This became apparent in how some students argued that one needs to know mathematics to programme, and conversely, in how they revealed that they had experienced programming in mathematics. Andy, for instance, saw problem solving as a potential connection between programming and mathematics. We believe that this result argues for a potentially fruitful integration between mathematics and programming.

Second, we found differences in how the students talked about and experienced the relationship between mathematics and programming. If we lend the vocabulary of Kilhamn et al. (2021), we can say that some students tended to talk about how programming can work as a computational tool in mathematics. Peter, for instance, explained how he uses concepts from programming to solve problems in mathematics. Moreover, he asserted that programming tools made him approach and view mathematics differently. Robie added to this picture when revealing how he saw programming helping him find multiple solutions to a given problem. This adds insight into Kilhamn et al.'s (2021) category of "programming as a tool for exploring mathematical concepts", where they propose that programming can be used to explore mathematical concepts and add new insights. However, there are students, Nora and Michael in particular, who experienced the elective programming course and mathematics as two different subjects. We see how this adds insight into the categories of Elicer and Tamborg (2021) in the way Nora and Michael's experiences seem to reflect an understanding of "no PCT in mathematics" and "no mathematics in PCT". Hence, we see how some students enhance their understanding of mathematics when engaging in programming activities, while others struggle to see how they can leverage programming in mathematics and vice versa.

Our third finding adds to the "categories of interplay" between programming and mathematics set forward by Kilhamn et al. (2021) and Elicer and Tamborg (2021). Both Nick and Peter stated that the elective programming course provided them with new skills. They emphasised that their experiences with programming had made them see the added value of being *systematic* and *accurate* in mathematics. We propose that this is a new category, and more research is needed to elaborate on it.

Concluding Remarks

This paper contributes to the ongoing discussion on how (and if?) programming and mathematics should be integrated in an educational context. We report on the experiences and views of the students, which, in sum, speak for an integration of the two. In addition, we assert, that by comparing our results with those of Kilhamn et al. (2021) and Elicer and Tamborg (2021), we strengthen their findings regarding how we see that the students' utterances confirm most of their categories of interplay.

References

Balanskat, A., & Engelhardt, K. (2015). Computing our future. Computer programming and coding–Priorities, school curricula and initiatives across Europe–update 2015. Brussels: European Schoolnet.

Bocconi, S., Chioccariello, A., & Earp, J. (2018). The Nordic approach to introducing computational thinking and programming in compulsory education. Report prepared for the Nordic@ BETT2018 Steering Group, 397–400.

Directorate of Education (2020, 28.05.2020). Nye læreplaner–grunnskolen. Retrieved 12.10.2020 from https://www.udir.no/laring-og-trivsel/lareplanverket/Nye-lareplaner-i-grunnskolen-og-gjennomgaende-fag-vgo/

Elicer, R., & Tamborg, A. L. (2021). Nature of the relations between programming and computational thinking and mathematics in danish teaching resources. Presented at ICTMT 15 international conference on technology in mathematics teaching, Copenhagen, Denmark

Faculty of Educational Sciences UiO. (2003). Released PISA 2003 mathematics tasks. University of Oslo.https://www.uv.uio.no/ils/forskning/prosjekter/pisa/frigitte-oppgaver/dokumenter/frigitt_problem.03.pdf

Husain, H., Kamal, N., Ibrahim, M. F., Huddin, A. B., & Alim, A. A. (2017). Engendering problem solving skills and mathematical knowledge via programming. *Journal of Engineering Science and Technology*, 12(12), 1–11.

Kaufmann, O. T., & Stenseth, B. (2021). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*, 52(7), 1029–1048. https://doi.org/10.1080/0020739X.2020.1736349

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

Mason, J. (2017). Qualitative researching. Sage.

Moreno León, J., Robles, G., & Román-González, M. (2016). Code to learn: Where does it belong in the K-12 curriculum? *Journal of Information Technology Education*. Research: Technology Publishing Educational, 15, 283–303. https://doi.org/10.28945/3521

Sinclair, N., & Patterson, M. (2018). The dynamic geometrisation of computer programming. *Mathematical Thinking and Learning*, 20(1), 54–74. https://doi.org/10.1080/10986065.2018.1403541

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25, 127–147. https://doi.org/https:/doi.org/10.1007/s10956-015-9581-5