Presented at Madif–12, Växjö, January 15, 2020

"Programming is a new way of thinking" – teacher views on programming as a part of the new mathematics curriculum in Finland

Ray Pörn[1], Kirsti Hemmi[1,2] and Paula Kallio-Kujala[1]
[1]Åbo Akademi University, Finland; ray.poern@abo.fi; kirsti.hemmi@abo.fi; pakallio@abo.fi,
[2]Uppsala University, Sweden; kirsti.hemmi@edu.uu.se

Programming has recently been included as a part of the mathematics curriculum throughout the grades in several countries. This is also the case in Finland. In this explorative study, we focus on Finnish Swedish primary school teachers' views of programming in school mathematics and on the connection that they spontaneously draw between mathematics and programming. Most teachers connect programming in primary school to the explicit activity of writing, giving or following instructions and to different aspects related to logical thinking. In addition, some teachers consider programming as an important problem-solving tool and still some of them mainly as an activity in mathematics. Only a few teachers connect programming to central computer science concepts as algorithms and abstractions and to specific mathematical areas.

Background

In line with several other countries, Finland has recently included programming as a part of mathematics curriculum in primary school (Duncan & Bell, 2015; Hemmi, Krzywacki & Partanen, 2017). This implies that primary school teachers (grades 1-6) have to integrate programming in their mathematics lessons. Finnish primary school teachers have a masters' degree in pedagogy but only a small fraction of them have mathematics as their minor subject (25 ECTS), and programming has not usually been a part of their university studies. Primary school teachers as generalists need widespread professional development concerning technical skills and understanding of suitable pedagogies to successfully implement new curriculum ideas (Benton, Hoyles, Kalas and Noss, 2017). Moreover, it is not quite clear what exactly is to be focused on at different grades as the Finnish national core curriculum is written in a very general way. The path from inclusion of programming in the mathematics curriculum to enacting lessons targeting it in a relevant manner is complex (Mannila, Dagiene, Demo, Grgurina, Mirolo, Rolandsson & Settle, 2014). The authors point out there are several issues to be discussed and defined to succeed in the implementation process of programming in the primary school classroom.

The present paper contributes by reporting the findings of an explorative study among primary school teachers the first year after the introduction of the new national core curriculum in Finland. We focus on what primary school teachers spontaneously ascribe to programming, and investigate particularly features connected to different aspects of computational thinking and the connections they make to topics typically included in mathematics learning. The research question is the following: *How do primary school teachers view programming as a part of the new mathematics curriculum?*

The Finnish context

The general task of mathematics education as stated in the national core curriculum (applied from 2016) is to develop students' logical, accurate and creative thinking (FNBE, 2014). Programming is included in the content of Mathematical thinking skills and applies to all students from first grade up to the end of grade nine (see Hemmi, Krzywacki & Partanen, 2017). Learning programming in mathematics starts in grades 1-2 with constructing simple algorithmic instructions by using symbols in written or oral form and testing them. During grades 3-6, the emphasis is on formulating instructions in a graphical programming environment.

There has not been any national efforts to systematically offer all teachers in-service education in programming, but different agents, such as regional authorities, universities, the National Board of Education and private companies, have frequently organized courses for teaching programming. In the Swedish-speaking part of Finland, that is the focus of our study, in-service education has been offered by one university. The courses offered for class teachers have focused for example on visual programming with block-oriented tools like Scratch, code.org and various applets, educational robotics, algorithmic thinking and elements of programmable electronics and making. Teaching programming has often been technology-driven and enthusiastic teachers and other actors have considered what they can do with a particular tool. Therefore, there might be a danger that a holistic picture of the learning path of children is not so clear for primary school teachers (cf. Hemmi et al., 2017).

Relevant literature

Computational thinking

The origins of computational thinking in early mathematics education can be traced back, more than thirty years, to the work of Papert who developed computer software to facilitate children to engage and explore computer programming as a natural problem-solving tool in their mathematics studies (Papert, 1980, 1996). Later, Wing (2006) defined computational thinking as "representing a universally applicable attitude and skill set involving solving problems, designing systems and understanding human behavior, by drawing on the concepts fundamental to computer science” (Wing, 2006 p. 33). After that, several organizations and authors have presented different definitions of computational thinking (Grover and Pea, 2013). Many of these definitions are quite general and may indeed involve activities not necessarily directly connected to programming and coding.

Brennan and Resnick (2012) introduced a framework with three dimensions of computational thinking: computational concepts, computational practices and computational perspectives. The first dimension includes common concepts that programmers use as they develop programs, such as sequence, iteration and function. Computational practices reflect different activities and problem solving practices that occur in the programming process, such as planning, testing, debugging, reusing and remixing. The third dimension, perspectives, involves the programmer’s connection and relationship to other members of the programming community and to the surrounding technological world. These dimensions are appropriate for understanding how K-12 students approach and connect to programming with Scratch and they are well in line with the programming content and ambition of the newly launched curriculum in Finland. Recently, another pedagogical framework for computational thinking was presented with focus on four pedagogical experiences: unplugged, tinkering, making and remixing (Kotsopoulos, Floyd, Namukasa, Somanath, Weber and Yiu, 2017). In our study, we used Brennan and Resnick’s (2012) dimensions as starting point for the data analysis, but broadened it taking an open iterative approach as our study is of explorative character.

While various studies about the relation between programming and mathematics in school curriculum have been conducted, the possible effects of programming on the learning of mathematics have not been clearly stated (Benton et al., 2017). However, many researchers highlight the critical role of the teachers in making explicit and systematic links between programming and students’ existing and developing mathematical knowledge (Benton et al., 2017).

Teachers’ views on programming

There is little knowledge in the field of mathematics education about teachers’ views of programming and teaching of programming as a part of mathematics curriculum and even less on how primary school teachers cope with the recent reforms in different countries. The paper by Mannila et al. (2014) surveyed teachers’ experiences about and perceptions of computational thinking in five European countries. Hijón-Neira, Santacruz-Valencia, Pérez-Marín and Gómez-Gómez (2017) investigated primary school teachers’ views on programming in schools in one region in Spain through a questionnaire and they analyzed the responses of 46 teachers. The teachers agreed on the benefits that programming provides in several areas, for example the development of thinking skills, the organization of ideas, the ability of abstraction and problem solving, motivational aspects, and the opportunities offered by teaching through games. The respondents remarked the importance of having properly trained teachers to teach this subject. Funke, Geldreich and Hubwieser (2016) interviewed six primary school teachers about their opinions on computer science and the findings pointed out that the teachers had no clear image on what computer science in school is, but they highlighted the importance of implementing computer science at an early educational stage. Recently Nouri, Zhang, Mannila & Norén (2019) investigated which skills 19 teachers interested in programming themselves aimed to develop among pupils. Apart from Brennan and Resnick's (2012) dimensions, they found some general skills related to digital competency and 21st century skills.

Pointing out the earlier concerns, for example about children bypassing mathematical ideas within less structured learning activities and without teacher guidance, Benton et al. (2017), examine the relationship between learning to program and learning to express mathematical ideas through programming with Scratch on primary level mathematics (age 9-11). The teachers in the study expressed that they needed the powerful ideas of mathematics curriculum to be clearly connected to the programming aspects of the computing curriculum. The study shows that it is possible to connect programming (with Scratch) to mathematical learning among students of different abilities and it had a positive effect on students' motivation. Yet, the study raised a number of concerns with respect to teachers' confidence and subsequent use of the technology within their teaching to support the learning of both computational and mathematical concepts.

Data collection and analysis

The target group of this study is primary school teachers that are working in schools where the instructional language is Swedish.[1] The empirical data for this study was obtained using a web-based survey that was distributed to teachers through Swedish Finnish primary school principals (190) during the spring term 2017. The survey contained 24, mostly multiple-choice questions and took about 30 minutes to complete. The final group of respondents consisted of 91 teachers, 70 female and 21 men. The age and regional distribution of the respondents were satisfactory. Of the 91 participants, 71 had participated in at least one in-service training course in programming. In this paper, we solely explore and analyze teachers' answers to one open question in the questionnaire: "What is programming? You should focus on programming in primary school but you can also relate to programming in general." From the context of the questionnaire, it is clear that this question is directly related to how the current change of the national curriculum (inclusion of programming) affected the mathematics content.

[1] Finland has two official languages and approximately 5% of students in compulsory education attend a school where Swedish is the language of instruction.

Possibly, due to the current nature of the topic, many respondents gave relatively rich answers. The number of words in the different answers varied from one word to 108 words and the mean number of words in the answers were 25. The teachers' responses (in Swedish) were first read and interpreted separately by two first authors of this paper. We initially started the analysis with the categories of Brennan and Resnick (2012) in order to identify what kind of computational thinking teachers' utterances were possibly expressing. Yet, these categories were not helpful in identifying other important views, such as curriculum issues and views on connections between programming and mathematics. Therefore, we also conducted data-driven iterative analysis (e.g. Bryman, 2001) starting by identifying certain similarities and generalities among the answers. Finally, we found the following six categories suitable for the final analysis: Writing, giving and following instructions; Logical thinking and identifying patterns; Algorithms, abstractions, modularization and testing; Problem solving; Use of modern technology and digitalization; Curricula, progression and future aspects. Due to the openness of the question, one answer could be assigned to several categories. In the Results section below, we describe the categories in more detail and exemplify them with teachers' expressions translated to English in order to make the analysis transparent.

Results

The distribution of the teachers' views on programming in relation to the six analytical categories can be seen in Table 1.

Category	***n* (% of *N*)**
1. Writing, giving and following instructions	59 (65)
2. Logical thinking and identifying patterns	29 (32)
3. Algorithms, abstractions, modularization and testing	10 (11)
4. Problem solving	17 (19)
5. Use of modern technology and digitalization	9 (10)
6. Curricula, progression and future aspects	15 (16)
Total	139

Table 1: Distribution of teachers' views on programming with respect to the six categories.

The number of answers assigned to different number of categories are; 0 (6), 1 (48), 2 (24), 3 (10), 4 (2), 5 (1). That is, six answers were uncategorized and 24 answers belonged to two different categories. No answer was assigned to all six categories. Below we describe and exemplify the categories identified for teachers' spontaneous views on programming in school.

Writing, giving and following instructions

This category is the most common among the answers as 65 % of the teachers connected programming to the explicit action of writing, giving or following instructions. The next extract shows a typical teacher answer in this category.

> In primary school education, it is important to let students test to program a computer, give instructions to another person or to a robot and try to make it complete the desired task. (Teacher 10)

As shown below, several teachers connect these kind of actions to activities associated to spatial thinking.

> [...] A simple way is to say; Go two steps to the right, one backwards and then five steps forward. Then you have come to the finish. (Teacher 33)

> For example, to be able to get a Beebot to go from one place to another by programming it. (Teacher 78)

A step-by-step procedure is sometimes connected to teachers' interpretations of instructions.

> Programming is to give detailed step-by-step instructions that do not offer space for misinterpretations or ambiguity. (Teacher 72)

Several teachers pointed out that the instructions need not to be given to a computer or robot, but equally well to a fellow student.

Logical thinking and identifying patterns

This category was the second most common as 32 % of the teachers point out that programming is connected to logical thinking and/or the identification of patterns. Most responses in this category state that programming promotes learning of logical thinking as shown in the extract below.

> Programming is, for example, to split a problem in to smaller parts, to see relations, to learn to think logically, to create something new." (Teacher 64)

Others express that programming is very similar to logical thinking.

> I think programming is very much about logical thinking and recognizing patterns. (Teacher 45)

Most of the teachers connect programming to a combination of handling instructions and applying logical thinking

Algorithms, abstractions, modularization and testing

These programming terms are common concepts in computer science and 11 % of the teachers claim that a central aspect of programming is the explicit constructing of algorithms, abstractions and the modularization or testing of a program.

> It is about coding, solving complex problems by splitting them into smaller pieces, identifying patterns, creating abstractions and writing algorithms. (Teacher 16).

Teachers that connect to these concepts are likely to have more in depth knowledge of the process of applying programming to solve problems.

Problem solving

The important problem solving aspect of programming is highlighted in 19 % of the answers.

> Programming is a really good activity that trains the ability to solve problems. (Teacher 43).

Another teacher who connects programming to “logical thinking, ability to solve problems, systematics and creativity” concludes with “Programming is mathematics.” (Teacher 72). Despite the close and important connection between mathematical problem solving and programming, no teacher answer is giving any explicit example of such a problem solving activity.

Use of modern technology and digitalization

A few teacher descriptions (10 %) consider programming in school from a more general perspective. Some responses address the importance of understanding the relation between human and machines.

> […] to realize that everything a machine can do is due to a human that has programmed it. (Teacher 10)

Others stress that programming is a part of modern technology.

> Several things in our close environment work with help of programming, e.g. machines, computer games and telephones. Industry uses robots that have been programmed. (Teacher 75)

This category captures more general aspects of programming not directly related to a school context.

Curricula, progression and future aspects

The aspects of curriculum and progression concerning programming in primary school are the focus in 16 % of the answers. Several teachers saw programming as a positive element in mathematics lessons and important for all students to learn, for example to prepare for future work life.

> We have to prepare them for the working life after school when they must be prepared to think creatively. (Teacher 58)

On the other hand, there were teachers who were not convinced about the importance of learning programming for all students.

> I think programming is fun, but I do not see it as a useful “subject”. That type of thinking can be acquired in many other ways. (Teacher 22)

Some of the teachers express a concern about the lack of information about the progress throughout the grades 1-6 concerning programming.

> Interesting, but I would like to have a clearer plan about what to do each school year. (Teacher 69)

Several teachers also stated that they lacked relevant curriculum materials for teaching programming.

Variation in teachers' descriptions

The range and the qualities in teachers' responses varied a lot. Some of the teachers touched several categories while others only responded with short sentences categorized into one category. The following extract is an example of the former and was coded into categories 1, 2, 3 and 6.

> Programming is a working process were you construct an algorithm, a hypothesis or a plan of how something should be executed or work. This plan is then tested and updated in order to work correctly. On a basic level, it can be as easy as working with numbered instructions. For older students it proceeds to the creation of block-based events using apps and computer programs and then finally in the highest grades by coding using a text-based language. (Teacher 62)

This teacher captures several important concepts and practices in computational thinking, such as instructions, events, algorithm, planning and testing as well as the progression of the topic. The second example is coded into categories 2, 3 and 4.

> Programming is all about logical thinking and problem solving. It is about coding, solving complex problems by splitting them into smaller pieces, identifying patterns, creating abstractions and writing algorithms. You can practice programming using different programs, games and languages. Programming is a new way of thinking. (Teacher 16)

The focus in this answer is on problem solving, the thinking aspect and the creation of algorithms and abstractions. The teacher also claims that programming really adds a new color to the classroom activity palette. The last example is coded into categories 2, 4 and 5.

> Programming is a way to teach students logical thinking, understanding of relations and problem solving. They develop both cognitively and linguistically. In time, they will understand that all new technology they use is based on programming. (Teacher 40)

This teacher specifically lifts logical thinking and problem solving as important learning goals and the technology connection is mentioned. The answer also highlights the communicative aspect of programming as being important.

Discussion and conclusion

The aim of this study was to explore and describe the way Finnish Swedish-speaking primary school teachers' view programming as a part of the new mathematics curriculum. Several teachers saw programming as a positive element in mathematics lessons and important for all students to learn. Programming was primarily connected to the explicit writing, giving or following of instructions. It was also considered to contribute to the development of logical thinking, serve as a valuable tool in problem solving and be a useful skill in future work life.

We next briefly comment on the relation of the answers to the framework by Brennan and Resnick. Many Finnish primary school teachers use Scratch as a programming tool and many of them have attended in-service training courses addressing Scratch. When they are to describe what they consider as programming, it might be that they view programming mainly through the lens of Scratch. Some answers that belong to category 1 and a few answers connected to category 2 and 3 are closely related to Brennan and Resnick's concept dimension. Common words in the teachers' answers are instruction, command, sequence, variable, conditional and loops. However, most answers relate to the practices dimension. The teachers typically describe how a certain programming activity connects to some other learning aspect such as logical thinking and problem solving. They focus on how the students are working and learning. As one answer reveals, "Programming is, for example, to split a problem in to smaller parts, to see relations, to learn to think logically, to create something new". The focus in this answer is clearly on practices modularizing and remixing.

The explicit connections to specific mathematical content were very scarce. For example, no teacher answer relates to the application of programming to arithmetic, algebraic expressions nor probability. The examples given can only be connected to spatial thinking and geometrical shapes. Along the lines with the concerns mentioned by Benton et al. (2017), it might be that primary school teachers' do not fully apprehend the interplay between mathematical and computational content and learning. This also relates to some teachers' concerns about lacking of information and relevant materials to be able to concretize the general goals of the national core curriculum. One suggestion for future action could be to organize educational courses tailored for primary school teachers that focuses on explicit connections between programming and mathematical content and learning.

References

Benton, L., Hoyles, C., Kalas, I. and Noss, R. (2017). Bridging Primary Programming and Mathematics: Some Findings of Design Research in England. *Digital Perspectives in Mathematics Education, 3*, 115-138. doi: 10.1007/s40751-017-0028-x

Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012* *annual meeting of the American Educational Research Association, Vancouver, Canada*. Retrieved from http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Bryman, A. (2001). *Social Research Methods*. Oxford: Oxford University Press.

Duncan, C., & Bell, T. (2015). A pilot computer science and programming course for primary school students. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, ACM, 39-48.

Hemmi, K., Krzywacki, H., & Partanen, A-M. (2017). Mathematics curriculum. The case of Finland. In D. R: Thomson, M. A. Huntley, & C. Suurtamm (Eds.), *International Perspectives on Mathematics Curriculum* (pp.71 – 102). USA: Information Age Publishing Inc.

Hijón-Neira, R., Santacruz-Valencia, L., Pérez-Marín, D., & Gómez-Gómez, M. (2017). An analysis of the current situation of teaching programming in Primary Education. In *Computers in Education (SIIE), International Symposium on Computers in Education (IEEE)*, 9-11 Nov. 2017, Lisbon, Portugal, (pp. 1-6). IEEE. doi: 10.1109/SIIE.2017.8259650

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. Educational Researcher, 42(1), 38–43.

FNBE (2014). *Perusopetuksen opetussuunnitelman perusteet 2014. [Core curriculum for basic education].* Finnish National Board of Education.

Funke, A., Geldreich, K., & Hubwieser P. (2016). Primary school teachers' opinions about early computer science education. In *Koli Calling '16: Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (pp. 135-139). ACM. doi: 10.1145/2999541.2999547

Kotsopoulos, D., Floyd L., Khan S., Namukasa I.K., Somanath, S., Weber, J. and Yiu, C. (2017). A Pedagogical Framework for Computational Thinking. *Digital Experiences in Mathematics Education*, 3, 154–171.

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. and Settle, A. (2014) In *ITiCSE '14 Proceedings of the 2014 conference on Innovation & technology in computer science education.*, 1-29. ACM.

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2019). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 1-17.

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas.* NY: Basic Books.

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning, 1*(1), 95-123.

Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35, 2006.