https://doi.org/10.36315/2023v1end041
Education and New Developments 2023

TEACHER TRAINING ON FUNDAMENTAL PROGRAMMING FOR MATHEMATICS AND TECHNOLOGY – WHAT ARE THE COURSE TAKEAWAYS?

Peter Mozelius
Department of Communication, Quality Management and Information Systems, Mid Sweden University (Sweden)

Abstract

The introduction of computer programming in K-12 education is a prioritised initiative in many countries. In the Swedish context this is an ongoing process with support from the government and the National Agency for Education. The focus has been set on secondary school and how to implement programming as an extension for teaching and learning activities in mathematics and technology. Several studies have reported on how to optimise the teacher training, but less has been presented on learning outcomes that can have an impact on teaching and learning in secondary school. The aim of this study was to analyse and discuss how learning outcomes could be of use in secondary school teachers daily teaching and learning activities. The main research question to answer was: *"In what parts of secondary school mathematics and technology can programming add value, and how might this be related to the learning outcomes of the teacher training?"* Data have been gathered from course participants essays that were submitted in two batches of the teacher training course on programming. Essays were analysed thematically with the research strategy of analytical induction. Found themes have been grouped into main categories that are related to the essayists course outcomes. Most of the essayists have completed the course, but not all of them. Findings indicate that the subset of course takeaways that the essayists have chosen to reuse are related to their learning outcomes, but also to the stage of secondary school where they teach. Lower secondary school teachers with low learning outcomes have a tendency to choose the themes of visualisation of geometrical concepts and computational thinking. To be compared with upper secondary school teachers with high learning outcomes that had a preference for using programming to illustrate number series and statistics. However, there are a lot of variations from these themes and counterpoints, which is the main discussion in this paper. Interesting new themes that emerged from the analysis of the essays were Debugging, General problem solving

Keywords: *Programming education, teacher training, learning outcomes, Python, K-12 STEM.*

1. Introduction

To involve computer programming in K-12 education is a prioritised and ongoing initiative in Europe, and in many other parts of the world (Lye & Koh, 2014; Balanskat & Engelhardt, 2015; Lindberg, Laine & Haaranen, 2019). In Sweden context this process was initiated by the government and the National Agency for Education. A revised a revised curriculum was presented in 2017 where it also was stated that programming should be involved in secondary school mathematics and technology (Heintz et al., 2017). This reform would obviously require a large-scale and long-term teacher training, since more or less all secondary school teacher in Science, Technology, Engineering and Mathematics (STEM) ought to participate, but cannot participate at the same time. What has been highlighted in several research publications is that STEM teachers without appropriate training have severe problems to handle the involvement of programming in their teaching and learning activities (Royal Society, 2017; Pörn, Hemmi & Kallio-Kujala, 2021).

The rationale for this study is that several research studies have reported on challenges and opportunities in the teacher training (Menekse, 2015; Humble & Mozelius, 2021), but less has been presented on learning outcomes that can have an impact on the actual teaching and learning in secondary school. As described in Mozelius and Humble (2022), this process should be seen as a two-stage rocket with the second stage, post teacher training, as the most important for a long-term success. The aim of this study was to analyse and discuss how learning outcomes could be of use in secondary school teachers

daily teaching and learning activities. The main research question to answer was: *"In what parts of secondary school mathematics and technology can programming add value, and how might this be related to the learning outcomes of the teacher training?"*

2. Research context – A course on programming for secondary school teachers

Several Swedish universities have developed contract courses to address the necessary teacher training. At the Mid Sweden University tailored courses on programming in K-12 settings have been given since 2018. The course is where this research has been conducted is one of many courses to support the curriculum development initiative. A lesson learnt is that a reuse of standard courses from Bachelor's programmes does not fit this target group. As an example, this kind of teacher training courses need a higher emphasis on didactic concepts and programming assignments aligned to STEM. With the idea that didactic concepts and STEM aligned assignments could be reused in course participants' daily teaching and learning activities. Examples of didactic concepts in the course are pair programming, visualisation, computational thinking and game-based learning (Mozelius, 2018).

From the beginning, and according to the instructions from the National Agency for Education, the course was of 7,5 ECTS, divided in the five main sections: 1) Programming in school, why, what and how? 2) The fundamental building block of programming; 3) Didactics for Technology and Mathematics; 4) Didactics for programming education; and 5) Mini-project. Considering the fact that the course participants are full-time working teachers, the course has been run at a 25% study pace. In 2020 the syllabus framework was redesigned to comprise 5 ECTS, and to have a stricter focus on textual programming. To compare with the original version that presented various other types of programming such as block programming and unplugged programming. In this new course design, Python was chosen as the only programming language.

Due to the outburst of the Corona pandemic, the new version of the course had to shift from a blend of synchronous face-to-face and asynchronous online activities to a full online course. Important to provide new alternatives for synchronous group activities. The video conference system Zoom has replaced the previous face-to-face meetings and activities such as ensemble programming and problem solving in smaller groups. What could be seen as a major challenge for the course, has turned out to be successful and efficient without lowering the pass rate. When the older version of this course only had an uptake of teachers from the Mid-Sweden region, he redesigned online version of the course have had a nation-wide enrolment.

3. Methods and materials

The overall strategy for this study was the action research approach where the author also has been teachers and facilitators in the involved course instances. Action research can be described as a practical but systematic research method that "enables teachers to investigate their own teaching and their student's learning" (Nolen & Putten, 2007, p. 401). The action research design in this study was in the British tradition of pedagogical action research, and as described by Norton (2018, 266 p. 71), an approach "that links research to improvement of practice and is education orientated". Data was gathered from course participants essays submitted in two batches of the described teacher training course. Essays were analysed thematically with the strategy of analytical induction, an iterative process in which preliminary themes and categories were developed early in the analysis process. Later, new found themes were compared to the earlier, to either support or refute existing categories and the relationships between them. Most of the essayists completed the course, and the analysed essay was the eighth assignment of ten in the course instances. As inspiration for the essay writing, four research articles aligned to the topic, were provided in the virtual learning environment. All essayists have been kept as anonymous as possible, and quotes have been translated from Swedish to English by the author. Some Swedish expressions and idioms have been replaced with their English equivalents to improve the readability and understanding.

4. Findings and discussions

Found themes were grouped into categories that are related to the essayists course outcomes. Findings show that the subset of the course content that the essayists have chosen to reuse are related to their learning outcomes, which also was an overall idea in the course design. Moreover, the different stages of secondary school where the essayists carry out their daily teaching and learning activities make an obvious difference. Therefore, the results from lower and upper secondary school are presented

separately here below. There have been more teachers from lower secondary school in the selected course batches and there were more findings to report in this category.

4.1. Lower secondary school

The two different trails for how to integrate the course takeaways in teaching and learning activities are not solely, but more clearly identified for lower secondary school. As expressed by one of the essayists "Currently I see programming in two different tracks; on one hand programming for its own sake, and on the other hand as a tool to better understand concepts such as geometry". However, regarding geometry several essayists found relevant ways to combine the two trails.

Geometry

Several good examples were found in the essays where relatively short snippets of programming code could support fundamental geometrical algorithms such as calculating circumferences, areas and volumes of various kinds of polygons. In one essay it is mentioned that "I think that programming should be involved as a natural part of geometry and algebra in the beginning", and that "Regarding geometry, I believe that it could be a way to understand the various algorithms that calculates areas and circumferences for geometrical figures". Another essayist points out that "Programming could visualise geometry and how different objects change, when different variables change" and that "Geometrical figures could of course be drawn on a whiteboard, but programming gives us a faster and more interactive tool". There are also ideas for how to calculate angle sums in geometrical figures, and how angles in o polygon could be related to the number of corners in a polygon. To involve programming in basic geometry is in some essays combined with the next category of visualisation.

Visualisation

Many teachers bring up visualisation of the geometry concepts in the previous section. An easy way of doing this is to use the Python Turtle module, a ingoing library in the standard Python environment that enables users to create shapes and figures with a pen on a virtual canvas. One of the teachers that started the course with no earlier programming experience found it harder to start with text-based programming compared to the more visual block programming. This teacher that also teach mathematics for primary school students (grade 4), tells that it was interesting to draw geometrical figures with Python Turtle, but that that younger students can get stuck in an “unreflected doing”, and just drawing patterns. There are also other forms of visualisation mentioned in the essays, such as when a teacher presents the idea of a program that visualises exponential growth, and the growth of high interest rates. All suggestions in the essays are for 2D-graphics, and that 3D-graphics would require add-on modules that could not be part of an introductory course. Several essayists also suggest to start out with block programming to get a more visual introduction to programming and computational thinking. With the idea that "Scratch could be a tool that creates motivation for further programming".

Programming in technology

The courses had a majority of teachers with mathematics as the main subject, but there were also some teachers in technology, and teachers teaching in both, and more subjects. It seems like teachers in technology have got a lot of new hardware resources during the last years, and one of them wrote that "Our school has recently purchased a robot and a 3D printer that the students should program and steer. It feels both exciting and scary, and will be a moment when my recently achieved programming skills will be tested." Technology teachers often bring up programming as a useful and necessary tool for control and regulation technology. A teacher brings up in his essay that "the initial programming was problematic, and that the municipalities 'IT Office' could not install Python properly on the school computers". His way forward was to switch to makecode.microbit.org, to get rid of installation issues. Microbit is a small pocket-sized computer that is good for illustrating how software can interact with hardware. This gave a new concrete touch to programming, and that "Two girls that earlier 'hated' programming became the new facilitator that helped other students". There are also essayists sharing experiences on non-working technology, and the frustrating feeling of not being able to carry out an intended lesson plan due to tech trouble. One teacher highlighted that "Tech trouble is a realty that we have to live with, and unfortunately technology-based assignments sometimes have to be skipped".

4.2. Upper secondary school

The essays from the upper secondary school teachers showed more specific suggestions, and a preference for using programming to work with algorithms and statistics, to illustrate number series and to solve equations. One teacher mentions that he wants to involve more concrete programming than the more general suggestions that have been given from the Swedish National Agency for Education.

He highlights that "As a upper secondary school teacher I'd expect that the fundamentals of programming have been learnt in lower secondary. There is no time for going through programming languages and computational thinking". One of hid concrete example is to implement the so called ‘pq-formula’ to solve second grade equations, that is mentioned by several upper secondary school teachers. Two of them have also inserted programming code and a lesson plan for how to apply this in the Python programming language. Another idea is to import the Python statistics module, to "To reuse ideas in the course assignments to calculate mean and median values". One of the upper secondary school teachers points out that "It will take quite a lot of time to introduce programming, and in the beginning, it should be with the idea of using programming where it obviously is more efficient. An example of that is in the solving of numerical problems where a computer iterates so much faster than a human". This idea is partly new, but on the other hand programmable pocket calculators has been used for many years for numerical solutions such as the Newton-Raphson method.

4.3. Common emerging themes

Debugging

One lower secondary school teacher wrote that she during the course "has found strategies for error tracing, and internet forums with help on debugging". Several other teachers bring up that the general debugging, that they have been forced to in the solving of the course assignments, have given them the confidence to program in front of their student groups. A bit the same for the upper secondary school teachers as well, where one of them writes that "I've learnt a lot, and especially when I've got stuck and have had to error trace my code". Another teacher, that needed two attempts to complete the course, writes that "It felt like a personal victory when I, in my second attempt, was able to handle everything myself.", and that "programming is a good training for finding errors in solutions".

Computational thinking and general problem solving

In one answer the essay writer brings up the discussions on computational thinking that were a part of the course. That students could improve their computational thinking skills and their general problem solving "motivates programming as part of the curriculum to a higher degree, and might make more teachers interested in learning to program". Some of the teachers that took this course had no earlier experience at all, on the other hand there were also teacher with very good prerequisites. One teacher mentions in her essay that she "just one year ago had plans to leave teaching for a programming job in the industry". With the richer experience of programming, it was claimed that "much in programming is about problem solving, which of course can be of use in mathematics as well", and that "students probably could see the value of problem solving more clearly when they start programming, and even stronger if they program in pairs or in groups". One of the teachers writes that problem solving will be one of the focus themes in her school, next semester when programming should be integrated in mathematics. Other more general themes that were suggested be aligned to the mathematical syllabi are 'Probability & Statistics' and 'Relationships & Change'.

5. General discussion

There is a clear alignment between the categories that were found in the analysis, and the stated learning objectives in the teacher training course, which also was intended in the original course design (Mozelius, 2018). Examples of programming assignments that have been brought on from the course assignments and group exercises are to write code for classical algorithms such as Eratosthenes sieve, and the Fibonacci series. At the same time, teachers present new and creative ideas on how to apply programming in their daily teaching and learning activities. The teacher attitudes in this study are probably significantly more positive than the ones of the average technology and mathematic teachers, since this is a course that the most reluctant teachers never will take. To quote one of the teacher essays: "Based on my own experiences I know that there is a certain resistance among the teachers that have no earlier programming experiences. They look like they are longing for retirement when this question is raised. Here I know that there will not be any programming in their classes - which probably is understood by the National Agency for Education as well.". The first years of the described initiative is a good start, but the challenge that remains is how to achieve the goal of 'Programming for all'.

6. Conclusion and future work

In a brief answer to the research question *"In what parts of secondary school mathematics and technology can programming add value, and how might this be related to the learning outcomes of the teacher training?"*. The conclusion is that lower secondary school teachers with low learning outcomes

had a tendency to choose the themes of visualisation of geometrical concepts and computational thinking. This should be compared with the upper secondary school teachers with high learning outcomes that had a preference for using programming to illustrate number series and statistics. However, were also several individual variations and counterpoints, with new emerging themes such as debugging and general problem solving. Moreover, there is a clear relation to the learning outcomes of the teacher training course, but there are also a few surprising exceptions that give ideas for future course updates. Findings from this study will serve as input for the next iteration of intervention and evaluation in the spirit of pedagogical action research. It would also be interesting to compare the findings from this study with the analysis of teachers' lesson plans that was conducted in Mozelius and Humble (2022).

References

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

Heintz, F., Mannila, L., Nordén, L. Å., Parnes, P., & Regnell, B. (2017). Introducing programming and digital competence in Swedish K-9 education. In V. Dagienė, & A. Hellas (Eds.), *Informatics in Schools: Focus on Learning Programming: 10th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 117-128). Springer, Cham.

Humble, N., & Mozelius, P. (2021). Enhancing Pedagogy to Andragogy in the Redesign of Teacher Training Courses on Programming. In C. Busch, M. Steinicke, R. Frieß, & T. Wendler (Eds.), *Proceedings of the 20th European Conference on e-Learning (ECEL 2021): A Virtual Conference Supported by University of Applied Sciences HTW Berlin, Germany* (pp. 210-217). Reading, UK: Academic Conferences and Publishing International Limited.

Lindberg, R. S., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology, 50*(4), 1979-1995.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior, 41*, 51-61.

Menekse, M. (2015). Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014. *Computer Science Education, 25*(4), 325-350.

Mozelius, P. (2018). Teaching The Teachers To Teach Programming: On Course Design and Didactic Concepts. In *11th annual International Conference of Education, Research and Innovation* (Vol. 11, pp. 8031-8037). The International Academy of Technology, Education and Development.

Mozelius, P., & Humble, N. (2022). Programming in K-12 Mathematics–A two-step rocket. In *INTED 2022, 16th annual International Technology, Education and Development Conference*, Valencia, Spain, [DIGITAL], March 7-8, 2022. (Vol. 16, pp. 2389-2397). IATED Academy.

Nolen, A. L., & Putten, J. V. (2007). Action research in education: Addressing gaps in ethical principles and practices. *Educational researcher, 36*(7), 401-407.

Norton, L. (2018). *Action research in teaching and learning: A practical guide to conducting pedagogical research in universities*. Routledge.

Pörn, R., Hemmi, K., & Kallio-Kujala, P. (2021). Inspiring or confusing–a study of Finnish 1–6 teachers' relation to teaching programming. *LUMAT: International Journal on Math, Science and Technology Education, 9*(1), 366-396.

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