

In-service Teacher Training and Self-Efficacy

Jørgen Thorsnes¹, Majid Rouhani¹, Monica Divitini¹

¹ Department of Information and Computer Science, NTNU, Trondheim, Norway
{jorgen.thorsnes;majid.rouhani;monica.divitini}@ntnu.no

Abstract. Programming is increasingly introduced in secondary schools, both as a stand-alone subject or integrated into other subjects, leading to growing attention to the training of in-service teachers. Teachers need to learn both (a) how to program and (b) how to teach programming, often in the context of different disciplines. The paper explores the impact of a university-level training program offered to in-service teachers, with a focus on teachers' self-efficacy in teaching programming. The paper reports the interviews with ten teachers after about one year they have completed the program. The results indicate that the training has improved teachers' self-efficacy, and the impact is lasting in time. Also, some teachers expressed concerns about their skill level in programming, but this does not necessarily associate with lower self-efficacy in teaching programming. The paper presents the results from the study and some implications for the design of training of in-service programming teachers.

Keywords: In-service Teacher Training, Self-Efficacy, Programming.

1 Introduction

In a recent report on the status of Informatics education in Europe, it is recommended that all pupils must have access to ongoing education in Informatics, and the teaching must be undertaken only by teachers who have formal education in Informatics [1]. However, there are several challenges to meet this recommendation and a growing demand for teacher training, in particular for the re-skilling of in-service teachers. In this context, it is essential to understand in-service teacher training to define relevant and effective training. In this paper, we focus on how formal training of in-service teachers impacts self-efficacy, with a focus on long-term impact. Self-efficacy refers to the belief in one's capabilities to organize and execute the courses of action required to produce a given result [2]. In terms of teaching a specific subject, the teacher's self-efficacy refers to their belief in their capabilities to teach the subject, such that pupils achieve the desired learning outcomes. Positive self-efficacy is connected to increased student and teacher outcomes, and it has a positive influence on teachers' psychological well-being [3]. However, several studies identify challenges with self-efficacy connected to programming education. For example, in two recent Swedish studies on teachers' attitudes and self-efficacy towards programming, the researchers found that many Swedish teachers lack confidence in teaching programming [4, 5]. Similar results were reported in UK schools, with many teachers

worried that they miss practical and theoretical knowledge of computing [6]. Given this background, we investigate how formal education at the university level for in-service programming teachers affects their self-efficacy. The main research question explored in this paper is: *How do in-service teachers perceive the lasting effect of programming education concerning their self-efficacy in programming and teaching programming?* To answer this question, we interviewed ten teachers who attended a university-level program on programming and programming education. Interviews were conducted almost one year after the completion of the course.

2 Case and method

Case description. Our study is connected to the in-service teacher training program at our university. The program consists of two courses of 7,5 ECTS each, the first with a focus on basic programming and the second on teaching programming. In the first course, teachers get an introduction to Python. The second course is more flexible, and teachers can select programming languages and topics on which to specialize based on their interests and needs [7]. Though there are no requirements for teachers to follow both of the courses, most do. The study program is aimed at in-service teachers in grades 8-13 (Lower and Upper Secondary School). The program is an online study, with web-based lectures and weekly activities such as online lectures and regular compulsory work exercises. Students participate in the course with the support of their school, which is committing to provide some free time to teachers to complete the course, though they continue their primary duties during the two semesters. The additional costs for the schools are partly covered by a national program of the Ministry of Education. This support leads to a very high completion rate. A survey distributed at the end of the program indicates high levels of satisfaction with the course. With this follow up study, we investigate how the training has contributed to teachers' self-efficacy and the long-term impact of the educational program.

Overall Method. This study is based on interviews with teachers that participated in the continuing education program in 2018-19. The study uses semi-structured interviews to explore the research questions by capturing teachers' reflections on their self-efficacy towards teaching programming.

Interview guide. The interview guide was constructed based on three main elements: (a) Attitudes towards programming in school; (b) Self-efficacy in teaching programming; (c) Self-efficacy in programming. Since we are interested in understanding the impact of the program, for each element, we added questions connected to the perceived impact of the program and changes since its completion.

For Attitude (a), we asked teachers their opinion about the ongoing introduction of programming in different subjects and, specifically, in the subjects that they teach. The questions related to Self-Efficacy in Programming (b) are inspired by the Teachers' Sense of Efficacy Scale (TSES) [8], with a focus on Efficacy for instructional strategies. Concerning teachers' programming skills (c), which is also relevant to teachers' ability to teach programming [9], questions were created to investigate how the teachers feel towards their own programming skills.

The questions were tested through test-interviews with three pre-service math teachers. The interviews were conducted during the COVID-19 pandemic. Schools were closed, with all the teaching taking place online and some elective subjects being postponed. We, therefore, added a final question on COVID-19. Our goal with this question was not to investigate its broader impact, but simply to check the validity of our study.

Participants. An invitation letter was sent to all the participants of the 2018/2019 cohort. An interview was then planned with the ones who expressed interest in participating in the study. Table 1 provides an overview of the participants.

Table 1. Overview of interviewees with Gender (M/F), school level (Upper Secondary School, USS, or Lower Secondary School, LSS), Subjects they are teaching (in italics the ones where they are *not* expected to use programming); Type of school (General, G, or Vocational, V)

<i>ID</i>	<i>G</i>	<i>Level</i>	<i>Subject(s)</i>	<i>Type</i>
1	F	USS	Economics	V
2	M	USS	Math, physics	G
3	F	USS	Math, natural science	V
4	M	LSS	Math, natural science, programming	G
5	F	LSS	<i>English, social studies, gymnastics, programming</i>	G
6	M	USS	Math, physics	G
7	F	USS	Math, <i>computer and electronics</i>	V
8	F	LSS	Math, natural science, <i>religion, programming, work-related training</i>	G
9	F	USS	<i>Physics, math, programming, natural science, technology and research</i>	G
10	F	USS	Construction - and control technique	V

Interviews were conducted via Zoom, using the service offered internally by our university for GDPR-compliance. The interviews were recorded with an external recorder and then transcribed by the interviewer. The relevant national agency approved the research. All the participants have been informed about the study, their rights and have been explicitly given their consent.

Analysis. After the transcription of the interviews, a thematic analysis [10] was performed by one of the authors using themes connected to the Teachers' Sense of Efficacy scale. The coding process was done with NVivo (QSR International, 2018). The final categories are: Attitudes towards programming in school; Teaching programming self-efficacy; Programming skill; Impact of programming education; Impact of time after programming education; COVID-19.

3 Results

3.1 Attitudes towards programming in school

The interviewees are generally positive towards teaching programming in school. They also express that they are positive towards using programming in an interdisciplinary context and underline the relevance for future jobs.

However, they also expressed some concerns that the inclusion of programming will be a long process, without a quick fix, especially considering the number of teachers who do not have any competence or education in programming. One of the issues that emerged from the analysis is the importance of teachers' community and collaboration. Four of the teachers expressed that they found collaboration with colleagues important when dealing with programming in school. Those who had someone to collaborate with reported that it was beneficial. Some other teachers indicated that they do not have colleagues to work within programming and that they would like to have that. Also, some teachers reported to have colleagues who are rather hostile to programming in school:

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

Some teachers also talked about gender differences related to programming. Particularly worrying is the resistance of gender stereotypes. One male teacher expressed that male teachers were more interested in programming than female teachers. One female teacher experienced that male teachers got much of the responsibility of programming related tasks in school, even if she is the only teacher with programming education:

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

3.2 Teaching programming and self-efficacy

In general, all ten teachers responded that they could teach programming, even if this is mostly connected to a specific subject or school level, for example:

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

Adapted Teaching. All of the teachers expressed that they could provide suitable challenges to capable pupils in programming, for example:

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

Some teachers explained that even though some pupils might be better than them in programming, this is not a major challenge. Teachers could usually find suitable assignments together with the pupils, or the capable pupils could get appropriate challenges through open tasks or freely choosing what they work with. In a recent study [11], US K-12 computer science teachers reported that it was challenging to meet all pupils' needs on an individual level. However, in our study, only one of the

teachers indicated a lower sense of self-efficacy in adapting her teaching to her pupils. Despite this, she still felt that she could provide appropriate challenges for highly skilled pupils by giving them freedom in what they were doing.

Assessment. The data analysis reveals that the teachers had a more varied sense of self-efficacy when it comes to assessment in programming. Four of the teachers expressed that they find assessment in programming difficult, for example because it is easy to find solutions online for the pupils and that they need strategies and tools for assessment. As a teacher explains:

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

Other teachers, however, are confident that they can assess their pupils, and some suggested oral presentations as a useful method, also to unveil whether the pupils understand their solutions or if they have copied it. One of the teachers that finds assessment challenging feels that it was little focus on this in the program.

Motivation. In general, the teachers seemed to have a relatively high sense of self-efficacy in motivating their pupils in programming. Nine of the teachers expressed that they felt they could motivate their pupils to learn to program. Some also thought that it was easier to motivate than in other subjects. For example:

...Yes. ... they [pupils] get to try something new. And those who have some prior knowledge get to do something they master. So yes, I think it is easier to motivate them in programming than in accounting, for example...(Teacher 1-Female-Economics-USS)

However, one of the teachers reported challenges with motivating students in elective courses:

... It's about how you meet the students. And in discussing with them, attempt to find angles of attack that motivate them. I feel that I manage that with some pupils, but then there is a problem in that not all pupils in elective programming are necessarily motivated to learn to program. ... Some of the pupils are there just because they did not get into the elective they wanted... that's a challenge... (Teacher 4-Male-Math-SS)

Explaining and conveying programming knowledge. Of the ten teachers, eight of them believe that, to some extent, they can explain programming concepts and come up with alternative explanations when pupils do not fully understand. Two of the teachers expressed that they could explain some programming concepts, but probably not all. Six of the teachers believed that they would not be able to answer difficult programming questions from the more capable pupils. The other four thought they could answer some difficult questions, but not all. However, nine of the teachers

believed that they could either come back to the pupil with the answer later or find the answer together:

... the pupils are also quite understanding when you say, "I can't do this very well, but I find it very fun! And I want to show you, and then we can figure it out together". They understand that, kind of. (Teacher 1-Female-Economics-HS)

The teachers seemed to have a moderately high sense of self-efficacy in this theme, but the main challenge is that the teachers do not perceive their programming competence as very advanced.

Developing teaching material. Eight of the teachers expressed that they can create good lesson plans and exercises, though it might be time-consuming, and that it would be beneficial with more time for planning lessons. Two of the teachers stated that they could not create suitable lessons from scratch, but they could by adapting existing teaching resources:

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

Most of the teachers state that they are using and finding teaching material online and adapt it to their classes. Four of the teachers expressed that they would like more relevant teaching material resources available. In general, the teachers indicated that they had a relatively high sense of self-efficacy in designing lessons in or with programming when there is relevant teaching material that they can adapt to their teaching. However, the willingness of experimenting with new lesson plans might be limited:

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

Challenges in teaching programming. When the teachers were asked what they perceived as the biggest challenge in teaching programming, two themes were prominent: Pupils' digital competences and technical issues. Three of the teachers talked about pupils' computer skills as a challenge. For example:

... Several of them name their files "one" "two" "three" and such, they do not have any system. So it will be difficult when you need to help them find a structure in programming when they can't even structure other things, so I think that might be most challenging...(Teacher 1-Female-Economics-USS)

Three of the teachers also talked about technical challenges. One of the teachers answered that some computer programs are challenging to use and cause technical problems and that she would like more user-friendly programs. Two other teachers explained that there are many technical issues, for example:

...The biggest challenge is technical. For example, when we code in Python, there are a lot of libraries and stuff that one needs, and then the pupils may have different

versions, and different modules and libraries, and nothing matches ... so we end up turning the computer off and on again, restarting, get frustrated because something that works, or code that works on one PC doesn't work on another PC. And that is by far the most frustrating, and what we spend the most time on - unnecessary time. Sometimes we also give up ... And there I have no competence to find out what the problem is.... (Teacher 9-Female- Math-USS)

3.3 Programming skill

The teachers were asked how they perceive their own programming skills. Eight of the teachers expressed that their programming skills were sufficiently good for teaching in their subjects and grades. However, six of the teachers indicated that their programming skill was relatively low, as one teacher states:

...For example, it is when we have embarked on slightly larger projects, which I may not have complete control over the development in. But so far, I have not been on extremely thin ice, but I have felt that "oh, I have to go home and pick up the book and read some more" I have had some of those rounds with myself....(Teacher 5-Female-English-LSS)

In general, the answers indicate that the teachers did not have a very high sense of self-efficacy towards programming, but that this did not severely impact their self-efficacy towards teaching programming. It also seems that even when their programming skills are relatively low, teachers perceive that they can increase this skill. Three of the teachers explained that they would like to have more follow-up in term of exercises or a local programming group to get better at programming:

...Yes, that it becomes just like an anonymous alcoholics group, that you have a follow-up group, "anonymous coders" who need some follow-up. Get some challenges and keep up (Teacher 4-Male-Math-LSS)

Previous research suggests that it is essential for teachers of programming to attain skills in programming [9]. This is also found in this study. However, even when the teachers perceive their own programming skills as not very high, they still feel capable of teaching programming in their grades and subjects. Many of the teachers also report that they feel capable of increasing their programming skills on their own or with colleagues and that they will get better with experience. This result indicates high self-efficacy towards getting better in programming.

3.4 Impact of programming education on self-efficacy in teaching programming

Six of the teachers expressed that they could not teach programming before the courses, but that they felt they were able to teach it in their subjects and grades after the program. This result indicates a very positive impact on the teacher's self-efficacy towards teaching programming. Three of the teachers felt they could have taught programming before the courses, but express that they felt more secure in their

teaching afterward. One teacher thought that the first course had a negative impact on her ability to teach programming because of the demanding workload and difficult exercises made her more confused than competent. However, this improved during the second course. Five of the teachers also stated that programming education had a positive impact on their attitudes towards programming in school, especially in terms of why programming in school can be relevant and beneficial in other subjects.

Some of the teachers felt that the learning curve in the first course was very steep. That programming can be hard to learn is often reported in the literature, e.g., [12]. Most of the teachers still felt that they learned a lot from the first course, and most were happy with both the learning outcome and the workload in the second course. Two of the teachers, however, reported that they thought the introductory programming course was very good, while they felt the second course was either too easy or had little impact on their competence.

3.5 Impact of time after course

The interviews were conducted close to one year after the teachers finished their programming studies. Therefore, they were also asked how the time that had passed since the completion of the program had impacted on their teaching. Most teachers perceived their programming skills lowered over time, when not used actively:

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

At the same time, these teachers expressed that they could "refresh" their programming skills and knowledge with little trouble. Also, a lower self-efficacy towards programming does not seem to relate to lower self-efficacy for teaching it.

The teachers that have taught or used programming in their classes in the last year explained that they also feel more capable of teaching programming due to the experience, for example:

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

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

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

Some of the teachers stated that they would have liked more follow up exercises or a community of teachers to help in maintaining programming skills after the courses. This suggestion can be seen in relation to the fact that many programming teachers work in schools without other teachers in their content area [11].

3.6 Impact of COVID-19

Five of the teachers stated that the COVID-19 situation might have impacted their answers. Interesting is, for example, that two of the teachers reported that they had found new teaching methods. For example, one stated:

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

One teacher reported that the answers might have been affected by insecurity about how to continue teaching programming in the current situation. Other teachers reported the lack of contact with colleagues and general concerns for the future, considering that many other teachers have not been able to follow planned training in the Spring. Interesting is also to see that the situation might have increased concerns about the general digital competence of other teachers:

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

4 Discussion and implications

The results of our study indicate that the teachers perceive that the education that they received had a positive impact on their self-efficacy towards teaching programming. Some of the teachers report that the studies also had a positive effect on their attitudes towards programming. The teachers say that their self-efficacy in teaching programming increases with experience in teaching programming, and does not significantly decrease without experience over time. The teachers report that their programming skills lower over time when not used, but that they can quickly refresh their programming skills when needed. This can be seen as a positive trend in relation to other studies where teachers showed a low level of self-efficacy in teaching programming [4–6]. Based on the results from the interviews, we identify some issues that need attention when designing training for in-service programming teachers.

Profiling and flexible paths. The evaluation of the two courses is rather varied. For some, the first course was too demanding, for others appropriate. The same holds for

the second course. Though there is a space for improvement of the actual course content, we think that these differences are strongly connected with the nature itself of the program. Participating teachers have different competence levels and different needs since they teach different subjects to different students. One could advocate for more specific courses, focusing on specific needs or requiring defined competencies for admittance. However, this is a model that is difficult to implement for economic reasons, and with pedagogical limitations, caging teachers in a specific subject and level. The alternative is then to create modular courses that support different learning trajectories, as in the program of our study [7]. However, this requires identifying ways of profiling participants and scaffolding their participation in the course.

Mini-courses/continuous education. One introductory course on programming and teaching is not enough to meet the challenges that teachers meet every day and to stay updated. Teacher training should be seen as a continuous process, with the possibility to follow up a more formal and structured training with shorter training activities in the form of, e.g., seminars and workshops on specific programming and pedagogical issues. This training requires a commitment from the providers of training to design their courses as a continuous process that looks at long term opportunities.

Importance to promote community. Collaboration among programming teachers is useful, both to increase and maintain programming skills and to share and discuss the teaching of programming. Some teachers also experience that their colleagues are negative towards programming in school. There is a need to develop and strengthen communities of practice in the domain of teaching programming. This can be done at different levels. Many countries have nation-wide communities to support programming teachers, as the CAS network in the UK [13]. These constitute an essential model, but it might be equally important to create a landscape of communities, locally at the teachers' school and around specific training courses. In this perspective, it is essential during a course to promote a sense of community among the teachers attending the courses and nurturing this community after the course is completed as a form of continuous education.

Re-use of resources. In our study, most of the teachers expressed that they can develop and adapt varied teaching material in programming and seem to have a relatively high sense of self-efficacy towards developing and adapting teaching material. However, some express that it can be difficult to find the most suitable teaching resources in the abundance of teaching resources in programming found online. A similar result is also found in the interview study of US K-12 computer science teachers [11]. Some teachers express that they would like more teaching resources in programming, as also indicated by the teachers in the study reported in [14]. The results also suggest that the teachers have a relatively high sense of self-efficacy towards motivating their pupils in programming, but that it can be difficult to create exercises that engage the lesser capable pupils in programming, while also giving them a sense of mastery. This indicates the need to create resources that can be adapted at different levels and to different national contexts, considering both language and study plan. As part of the training, it is important to include information about available resources and how to adapt them.

Assessment. Our respondents identify assessment as challenging, mirroring other studies that pointed out the need for quality assessment tools in computer science and coding education [11]. Though we do not have conclusive evidence, it also seems that assessment in programming is a more significant challenge than the teachers without experience perceive. It is, therefore, important that any teacher training program explicitly addresses this issue and challenges teachers to address assessment as an integrated part of the definition of their lesson plan.

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

5 Conclusions

In this paper, we presented the results from interviews conducted with ten teachers, one year after they completed formal university-level training in programming. The teachers express high self-efficacy about teaching programming, though some of them are reporting low skills in programming. The study shows that teachers perceive formal training at the university level as having a positive and long-lasting impact.

The interviewees differ in terms of subjects they teach and school type. Therefore, the study covers different perspectives. However, we are fully aware that they are all connected to the same course, and it might be difficult to generalize the results. More studies are necessary. Considering that this type of training requires a heavy investment, both at the individual and the school level, we claim it is crucial to investigate which types of training have the highest impact, not only in terms of acquired knowledge but also in terms of attitudes and self-efficacy. From our study, formal training at the university level seems a promising option.

Acknowledgments

We thank the teachers who participated in the interviews. The work is partly funded by Excited, The Norwegian Center for Excellent IT Education (<https://www.ntnu.edu/excited>).

References

1. The Committee on European Computing Education (CECE): Informatics Education in Europe: Are We All In The Same Boat? Association for Computing Machinery, New York, NY, USA (2017).

2. Bandura, A.: Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*. 84, 191–215 (1977). <https://doi.org/10.1037/0033-295X.84.2.191>.
3. Zee, M., Koomen, H.M.Y.: Teacher Self-Efficacy and Its Effects on Classroom Processes, Student Academic Adjustment, and Teacher Well-Being: A Synthesis of 40 Years of Research. *Review of Educational Research*. 86, 981–1015 (2016). <https://doi.org/10.3102/0034654315626801>.
4. Hartell, E., Doyle, A., Gumaelius, L.: Teachers' attitudes towards teaching programming in Swedish Technology education. Presented at the PATT 37, Pupils Attitudes Towards Technology (2019).
5. Mannila, L., Nordén, L.-Å., Pears, A.: Digital Competence, Teacher Self-Efficacy and Training Needs. In: *Proceedings of the 2018 ACM Conference on International Computing Education Research*. pp. 78–85. Association for Computing Machinery, New York, NY, USA (2018). <https://doi.org/10.1145/3230977.3230993>.
6. The Royal Society: After the reboot: computing education in UK schools. The Royal Society - November 2017.
7. Rouhani, M., Divitini, M., Vujosevic, V., Stai, S., Olstad, H.A.: Design of a programming course for teachers supporting flexible learning trajectories. In: *Proceedings of the 8th Computer Science Education Research Conference*. pp. 33–38. Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3375258.3375263>.
8. Tschannen-Moran, M., Hoy, A.W.: Teacher efficacy: capturing an elusive construct. *Teaching and Teacher Education*. 17, 783–805 (2001). [https://doi.org/10.1016/S0742-051X\(01\)00036-1](https://doi.org/10.1016/S0742-051X(01)00036-1).
9. Korkmaz, O.: Prospective CITE Teachers' Self-efficacy Perceptions on Programming. *Procedia - Social and Behavioral Sciences*. 83, 639–643 (2013). <https://doi.org/10.1016/j.sbspro.2013.06.121>.
10. Braun, V., Clarke, V.: Using thematic analysis in psychology. *Qualitative Research in Psychology*. 3, 77–101 (2006). <https://doi.org/10.1191/1478088706qp063oa>.
11. Yadav, A., Gretter, S., Hambrusch, S., Sands, P.: Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education*. 26, 235–254 (2016). <https://doi.org/10.1080/08993408.2016.1257418>.
12. Guzdial, M.: Learner-Centered Design of Computing Education: Research on Computing for Everyone. *Synthesis Lectures on Human-Centered Informatics*. 8, 1–165 (2015). <https://doi.org/10.2200/S00684ED1V01Y201511HCI033>.
13. Brown, N.C.C., Sentance, S., Crick, T., Humphreys, S.: Restart: The Resurgence of Computer Science in UK Schools. *ACM Trans. Comput. Educ.* 14, 9:1–9:22 (2014). <https://doi.org/10.1145/2602484>.
14. Kadirhan, Z., Gül, A., Battal, A.: Self-Efficacy to Teach Coding in K-12 Education. In: Hodges, C.B. (ed.) *Self-Efficacy in Instructional Technology Contexts*. pp. 205–226. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-99858-9_12.