



## OPEN ACCESS

## EDITED BY

Fien Depaepe,  
KU Leuven,  
Belgium

## REVIEWED BY

Charlotte Struyve,  
KU Leuven,  
Belgium

Rani Van Schoors,  
KU Leuven Kulak,  
Belgium

## \*CORRESPONDENCE

Miriam Leuchter  
leuchter@uni-landau.de

## SPECIALTY SECTION

This article was submitted to  
Educational Psychology,  
a section of the journal  
Frontiers in Psychology

RECEIVED 06 July 2022

ACCEPTED 18 August 2022

PUBLISHED 28 September 2022

## CITATION

Weber AM, Bastian M, Barkela V,  
Mühling A and Leuchter M (2022) Fostering  
preservice teachers' expectancies and  
values towards computational thinking.  
*Front. Psychol.* 13:987761.  
doi: 10.3389/fpsyg.2022.987761

## COPYRIGHT

© 2022 Weber, Bastian, Barkela, Mühling  
and Leuchter. This is an open-access  
article distributed under the terms of the  
[Creative Commons Attribution License \(CC  
BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in  
other forums is permitted, provided the  
original author(s) and the copyright  
owner(s) are credited and that the original  
publication in this journal is cited, in  
accordance with accepted academic  
practice. No use, distribution or  
reproduction is permitted which does not  
comply with these terms.

# Fostering preservice teachers' expectancies and values towards computational thinking

Anke M. Weber<sup>1</sup>, Morten Bastian<sup>2</sup>, Veronika Barkela<sup>3</sup>,  
Andreas Mühling<sup>2</sup> and Miriam Leuchter<sup>3\*</sup>

<sup>1</sup>Computer-Based Assessment Research Group, Department of Behavioural and Cognitive Sciences, University of Luxembourg, Esch-sur-Alzette, Luxembourg, <sup>2</sup>Computing Education Research Group, Department of Computer Science, University of Kiel, Kiel, Germany, <sup>3</sup>Department of Educational Sciences, Institute for Children and Youth Education, University of Koblenz-Landau, Landau, Germany

**Theory:** Digital technologies have become an integral part of everyday life that children are exposed to. Therefore, it is important for children to acquire an understanding of these technologies early on by teaching them computational thinking (CT) as a part of STEM. However, primary school teachers are often reluctant to teach CT. Expectancy-value theory suggests that motivational components play an important role in teaching and learning. Thus, one hindrance to teachers' willingness to teach CT might be their low expectancies of success and high emotional costs, e.g., anxiety towards CT. Thus, introducing preservice teachers to CT during their university years might be a promising way to support their expectancies and values, while simultaneously alleviating their emotional costs. Prior CT competences might contribute to these outcomes.

**Aims:** We investigated whether a specifically designed seminar on CT affected preservice teachers' expectancies and values towards programming.

**Method:** A total of 311 German primary school and special education preservice teachers took part in the study. The primary school preservice teachers received a seminar on CT and programming with low-threshold programming tasks, while the special education teachers served as a baseline group. The seminar was specifically designed to enhance expectancies and values and decrease emotional costs, following implications of research on expectancy-value theory.

**Results:** The preservice teachers who visited the seminar gained higher expectancies and values towards CT and programming compared to the baseline group. Moreover, their emotional costs decreased. CT was positively related to change in expectancies and values and negatively related to emotional costs.

**Discussion:** Interventions with low-threshold programming tasks can support primary school preservice teachers in finding trust in their abilities and values towards CT. Moreover, their anxiety towards CT and programming can be alleviated. Thus, first steps in preparing preservice teachers to teach CT in their future classrooms can be taken in university.

## KEYWORDS

expectancy-value theory, computational thinking, preservice teacher education, primary school, values, emotional costs, expectancies of success

## Introduction

Today's society is becoming increasingly digitized which means that children come into contact with and use programmed devices such as tablets, smartphones, and computers at an early age. Therefore, it is important that even young children acquire an understanding of these technologies. More specifically, children should not only be able to use these technologies, but also have a basic understanding of how they work (Barr and Stephenson, 2011). Solving problems with the help of computers, e.g., by writing a program, is commonly referred to as computational thinking (CT). More specifically, CT encompasses cognitive processes that formulate problems and their solutions so that they can be processed by an information processing agent (e.g., a computer or a processor; Newell et al., 1959; Wing, 2006; Faber et al., 2017; Shute et al., 2017; Weber et al., 2021).

The teaching of CT and its implementation as a part of STEM is called for by different stakeholders (e.g., Wing, 2006; National Research Council, 2010; Ministerium für Bildung, Wissenschaft, Jugend und Kultur, 2018; OECD, 2018). While advancements to implement CT into secondary education have been made, teaching CT at primary school level has seen fewer efforts (Angeli et al., 2016). However, learning CT skills early might be beneficial as it might foster children's problem-solving skills as a part of their 21st century skills (Barr and Stephenson, 2011; Jaipal-Jamani and Angeli, 2017; Yadav et al., 2017; Lamprou and Repenning, 2018). Moreover, it can support their metacognition and cognitive flexibility, which are goals of primary school education (National Research Council, 2010; Di Lieto et al., 2017; OECD, 2018). CT can be implemented into primary school classrooms by using simple block-based programming tasks or unplugged CT activities (Wing, 2006; Brackmann et al., 2017). For the most part programming tasks are used at this education level because children have fun programming robots or computers (Hsu et al., 2018) and solving problems through programming is a central part of CT (Román-González et al., 2017; Shute et al., 2017).

Teaching CT as early as in primary school is possible (Angeli et al., 2016; Dickes et al., 2016; Di Lieto et al., 2017). However, implementing CT poses challenges for primary school teachers. In order to teach CT, teachers require CT competences and knowledge of support measures in informatics problem solving that they often do not possess (see, TPACK; Angeli et al., 2016). Thus, it is desirable to implement CT education into preservice teacher education at the university level to target this challenge early on (Yadav et al., 2014; Butler and Leahy, 2021).

However, political, administrative, educational, and interindividual hindrances need to be overcome to integrate CT education into primary school (Angeli et al., 2016). This paper addresses some of these interindividual obstacles in teacher education because preservice teachers often are reluctant concerning the learning and teaching of CT (Gal-Ezer and Stephenson, 2010; Weber et al., 2021). Following expectancy-value theory (EVT; e.g., Eccles et al., 1983), one of the most influential motivational theories, reasons might include low expectancies of

success as well as low values towards and fears of CT and programming. These fears might be heightened for women who make up a large percentage of primary school preservice teachers (OECD, 2021) due to stereotype threats (see Shapiro and Williams, 2012; Marsh et al., 2019).

Eccles (2009) claims that expectancies and values predict a person's choices and decisions. As CT and programming grow ever more important, it is crucial that primary school teachers choose to teach these important topics to young learners and actively decide to implement them into their classrooms. Teachers might be more likely to teach this content if they perceive themselves as able, value CT, and are under the impression that it is important for children to learn basic CT competences. Therefore, one step to take in order to address interindividual obstacles that stand in the way of CT education might be to implement a preservice teacher education program focusing on CT. This program can encompass measures that foster preservice teachers' expectancies of success as well as positive values regarding CT. Consequently, this study investigates preservice teachers' expectancies and values regarding CT as a part of STEM.

## Expectancy-value theory

According to EVT (e.g., Eccles et al., 1983), both the expectancy of success and the value a person places on a task or a domain influence persistence and the pursuit of plans (Bong et al., 2012; Musu-Gillette et al., 2015). Therefore, preservice teachers' expectancies of success and values might be important for their willingness to implement complex STEM topics such as CT into their classrooms (Rich et al., 2021). Considering the challenges CT poses for primary school teachers (Yadav et al., 2014)—e.g., overcoming anxieties, having to learn a programming language—teachers with higher expectancies and values regarding the teaching of CT might be more likely to implement it in their classrooms.

## Expectancies of success

Expectancies of success are often operationalized in the form of competence beliefs, such as self-concept and self-efficacy (Marsh et al., 2019). For example, Eccles (2009) has stated that expectancy of success and self-concept are linked in a way that makes it hard to distinguish between the two constructs. Therefore, self-concept can be viewed as one representative of a person's expectancy of success in a domain. Self-concept refers to broad self-beliefs about one's competence and includes self-evaluations that often draw on past experiences (Marsh et al., 2012). For example, a person with a high self-concept in CT might view programming – a central part of CT – as one of their strengths. Teachers might differ in their self-concept regarding CT and therefore some teachers might view themselves as rather good at programming, while others might consider themselves as rather bad at programming. For example, Faherty et al. (2021) found that

secondary school teachers' self-concepts in programming ranged from very low to very high, when asked about their programming skills. Moreover, Bouck et al. (2021) state that special education preservice teachers report relatively low self-concepts in teaching CT to students.

Another component indicative of expectancy of success is self-efficacy as a specific belief about certain tasks within a domain (Trautwein et al., 2013; Wigfield et al., 2017). In contrast to self-concept, self-efficacy is often construed as a belief that is more descriptive in nature and refers to tasks that lie in the future (Marsh et al., 2012). For example, a person with a high self-efficacy in CT might determine that they will be able to solve specific programming tasks, such as programming loops or conditions. Teachers might differ in their self-efficacy regarding CT as well. Thus, some might be quite confident that they can solve specific programming tasks. Others might hold more reluctant beliefs about their ability to solve the same tasks. Studies by Boulden et al. (2021) and Rich et al. (2021) found that primary school preservice teachers held different self-efficacy beliefs towards programming and the teaching of CT that were not influenced by their demographic characteristics. Moreover, Zhao et al. (2020) discovered that teachers' self-efficacy beliefs concerning the teaching of CT increased during a three-week Scratch online course.

## Values

Following Eccles (2009), there are four different value components, intrinsic value, utility value, attainment value, and costs. Intrinsic value is conceptualized as intrinsic motivation, i.e., the enjoyment of a certain task. Teachers might differ in their enjoyment of programming. Jaipal-Jamani and Angeli (2017) found that primary school preservice teachers mostly reported that they enjoyed learning about programming in the context of robotics. Moreover, their enjoyment increased after taking part in a programming activity with educational robots.

Utility value refers to the usefulness of something to achieve short- or long-term goals that a person has (Wigfield and Cambria, 2010; Marsh et al., 2019). Teachers might differ in their perception of needing to learn the basics of CT and programming themselves, based on their own willingness to teach CT and programming to young learners. Jaipal-Jamani and Angeli (2017) investigated whether primary school preservice teachers would teach CT in their classrooms. They found that most preservice teachers disagreed to teaching CT. However, their values regarding the teaching changed towards a more positive attitude after participating in a robotics programming task.

Attainment value describes the personal value a person attributes to something (Eccles, 2009). Teachers might differ in whether they value programming or the development of CT competences as an important learning goal for children. In a qualitative study with special education preservice teachers, Bouck et al. (2021) report that most preservice teachers value the importance of teaching CT to their students. This might also be the case for primary school preservice teachers.

Last, costs imply the negative consequences a certain task might have, e.g., emotional costs such as anticipated anxiety. Teachers might differ in their anxiety towards programming and CT. In the context of programming, Nolan and Bergin (2016) found that even computer science students often are anxious about programming. This anxiety might be even higher for primary school teachers or preservice teachers (Weber et al., 2021), because they are mostly female. Studies found that women often tend to think they have low abilities in STEM (Shapiro and Williams, 2012; Marsh et al., 2019). Moreover, CT courses are often absent from teachers' university education (Angeli et al., 2016), which might contribute to their anxiety towards the subject.

The studies cited above did not specifically investigate EVT and all its components. As both expectancy and value can influence persistence and the pursuit of plans (Bong et al., 2012; Musu-Gillette et al., 2015), it is worthwhile to investigate preservice primary school teachers' expectancies of success and values towards programming as indicators for their later willingness to teach CT to young learners. Additionally, their EVT components might be changed during a specifically designed intervention (O'Mara et al., 2006; Dowker et al., 2016; Wigfield et al., 2017).

## Designing a university seminar to support expectancies and values

Wigfield and his colleagues (Guthrie et al., 2004; Wigfield and Guthrie, 2010; Wigfield et al., 2017) found that students' expectancies of success and values towards different learning outcomes can be fostered with specifically designed interventions. Such interventions that target expectations and values, as is demanded by Nagengast et al. (2011), should include six characteristics. (1) Interventions should have content goals. These can help students structure their learning and provide them with a sense of purpose by making them aware why they are completing certain tasks or why they need to read a certain article (van de Pol et al., 2010; Belland et al., 2013). For example, the lecturer can explain why and how a task supports students' programming competences. (2) Students should be able to make choices and have control over the tasks whenever possible. This can support their desire for autonomy and prevent them from experiencing frustrations (Wigfield and Tonks, 2004). Even if students need to complete all tasks for a seminar, they can be offered control over the framework conditions (van de Pol et al., 2010). For example, students can decide when to complete a task and which resources to use. (3) Hands-on activities might help students to realize the value of what they study. For example, completing low-threshold programming tasks might allow students to appreciate that CT can be implemented into their future classrooms even with young learners (Angeli et al., 2016; Sengupta et al., 2018; Weber et al., 2021). This might in turn enhance their values towards CT and programming as preservice teachers take great interest in example tasks that they can easily implement later in their career

(Garet et al., 2001). (4) Interesting tasks should be used for instruction to enhance students' intrinsic value to engage with the subject matter. For example, a programming task that can easily be implemented in a future classroom and consists of everyday activities or objects, e.g., programming a bicycle lamp, serves this purpose (Angeli et al., 2016; Reuter et al., 2021). Again, such tasks are interesting especially for preservice teachers (Garet et al., 2001). (5) Collaborations between students regarding the learning objective can increase students' values towards learning through working with others (Wigfield et al., 2014). For example, students can program in pairs or groups. Moreover, a seminar lecturer can offer the students the possibility of exchange in an online forum (Kong et al., 2018). (6) The importance of the learning objective should be emphasized to foster students' value beliefs. Students should also be encouraged to reflect about the importance themselves in order to value their autonomy (Wigfield et al., 2017). Thus, a lecturer can highlight the importance of learning CT for young children and therefore the need for preservice teachers to learn it as well (Yadav et al., 2014). The students can then reflect on this themselves.

## Required content knowledge about CT for the primary school classroom

To teach CT in future primary school classrooms, preservice teachers need TPACK (technological pedagogical content knowledge). TPACK encompasses a basic understanding of CT, at least rudimentary programming skills, and knowledge of pedagogical approaches to teach CT to young learners (Angeli et al., 2016). To acquire knowledge about CT and programming, block-based visual programming languages like Scratch or NEPO<sup>®</sup> are suited, because they lighten the burden of writing syntactical correct programs (Resnick et al., 2009).

CT is a set of thought processes in which problems and their solutions are formulated in a way that can be processed by an information processing agent (e.g., a computer; a processor; Newell et al., 1959; Wing, 2006; Faber et al., 2017; Shute et al., 2017; Weber et al., 2021). Thus, CT encompasses a set of cognitive processes, such as decomposing problems into subproblems, abstracting, sequencing algorithms, control flow, debugging and generalizing. These processes enable learners to solve problems in a specific way, i.e., by creating algorithms (White and Sivanides, 2003; Wing, 2006; National Research Council, 2010; Angeli et al., 2016). To plan a solution, a given problem needs to be decomposed into subproblems to reduce complexity. Thus, the decomposition will facilitate the understanding and the solving of the problem (Newell et al., 1959), e.g., because solution strategies for the subproblems might already be known (Faber et al., 2017). Abstraction excludes irrelevant properties of a problem to expose the underlying algorithm and can make problem-solving activities more efficient, given that certain algorithms represent certain subproblems (Jaipal-Jamani and Angeli, 2017). Sequencing is the ability to put actions in the correct order (Kazakoff et al., 2013).

Flow control verifies that each instructional step is sequenced correctly to ensure the activity runs flawlessly. When writing an algorithm, each step of a solution must be defined in detail and in the correct order. Therefore, sequencing and flow control are important cognitive processes to create algorithms without errors (Angeli et al., 2016). An algorithm can be evaluated by executing and monitoring it for potential errors. The debugging process identifies and eliminates these potential errors (Hsu et al., 2018). Generalization fosters flexible computational problem-solving by transferring devised solutions to other tasks with the same properties. Algorithms addressing subproblems can then be used synergistically and thus simplify future problem-solving processes (Curzon et al., 2014). Even though these processes are tailored very specifically to the design of algorithms in problem solving, they can also be seen more generally as a skill set relevant for problem solving or scientific inquiry (Shute, 1991; Bers, 2018). Nevertheless, the ability to solve problems by writing a program is typically viewed as very central to CT (Román-González et al., 2017; Shute et al., 2017). Acquiring and training CT skills as early as in primary school—usually using simple programming tools—may foster problem solving skills, cognitive flexibility and metacognition, which are main goals of primary education (National Research Council, 2010; Di Lieto et al., 2017; OECD, 2018). Furthermore, CT as a part of STEM education can be implemented at an early stage, maybe even serving as a motivator to pursue STEM when growing up (Shute, 1991; Nugent et al., 2015). Thus, preservice teachers need to understand CT and its importance and learn about possibilities to implement CT in primary school.

When teaching CT to preservice teachers, the complex subject of programming can be simplified by implementing low-threshold tasks (Weber et al., 2021). Low-threshold tasks can be short and simple programming exercises in a context that is familiar to preservice teachers and easy to implement into their future classrooms (e.g., a bicycle light, a traffic light, a metronome). Furthermore, low-threshold tasks may lower the preservice teachers' anxiety of failure and their reluctance to program (Bescherer and Fest, 2019). Nevertheless, CT skills can also be acquired using logical reasoning or means-ends analyses and do not necessarily have to be programming tasks (Zhang and Nouri, 2019). However, using robots and computers may additionally motivate preservice teachers to learn CT, because they might expect their future students to have a fun experience by programming robots and computers (Garet et al., 2001).

## The present investigation

Even though the teaching of CT is a goal for educational policy (Ministerium für Bildung, Wissenschaft, Jugend und Kultur, 2018), CT is severely neglected in German primary schools (Feierabend et al., 2019). Studies suggest that introducing CT in primary educational levels is beneficial for developing problem-solving skills as a part of the 21st century skills (Jaipal-Jamani and

Angeli, 2017; Yadav et al., 2017; Lamprou and Repenning, 2018). As has been shown in science education in primary school, high emotional costs and uncertainty about teaching CT may discourage teachers from integrating this important learning objective into their classrooms (Jaipal-Jamani and Angeli, 2017). To address these reservations, we implemented a university seminar to educate preservice teachers in CT, while addressing their expectancies and values towards this learning objective to motivate and increase their readiness to implement CT into their future classrooms. According to the literature, both components can be fostered with interventions that target expectancies and values (Wigfield et al., 2017). The literature suggests six characteristics that should be fulfilled to foster expectancies and values as stated above. (1) Interventions should have content goals; (2) preservice teachers should have control over tasks; (3) interventions should include hands-on-activities; (4) interesting tasks should be implemented; (5) collaborations between preservice teachers should be allowed, and (6) lecturers should emphasize the importance of the learning objective.

Due to the Covid-19 pandemic, the 11-week seminar was conducted online.

In this context, we specify the following research questions:  
Can instruction in the context of this seminar

1. Improve preservice teachers' expectancies of success—i.e., their self-concepts and self-efficacy?
2. Improve the value preservice teachers attribute to programming—i.e., intrinsic value, utility value, and attainment value?
3. Decrease preservice teachers' programming anxiety, i.e., their emotional costs?
4. Are these changes dependent on preservice teachers' prior CT competences?

## Materials and methods

### Participants

In total, 311 German university students participated in the study, of which 153 (133 female, 20 male) studied M.Ed. primary school education and attended a science education seminar on learning and teaching CT and programming to primary school children (experimental group, EG). The students were on average  $M=24.32$  years old,  $SD=2.67$ , and in their  $M=1.39$ ,  $SD=1.39$  master semester, range = 1–12.

The remaining 158 participants (128 female, 30 male) studied M.Ed. special education at the same university and did not attend the science education seminar (baseline group, BG). The BG was implemented to detect possible retest effects. On average, the students were  $M=24.47$  years old,  $SD=2.61$ , and in their  $M=2.41$ ,  $SD=1.49$ , master semester, range = 1–10.

In Germany, primary school preservice teachers take courses in German, a foreign language (English or French), mathematics,

and science throughout their 6-semester B.Ed. as well as 2-semester M.Ed. programs. These courses focus on both the domains and the teaching of domain-specific contents to primary school children. Special education preservice teachers study subject-related courses such as German, foreign languages, mathematics, or science without a focus on teaching these subjects during the first four semesters of their B. Ed. studies. The fifth and sixth semester as well as their 2-semester master program focus on special education needs and pedagogy, while subject-related courses are not implemented.

All students were informed about the goal of the study and gave their written consent to participation.

### Procedure

The EG attended an online science seminar targeting CT, including the use of a block-based programming language and teaching CT in primary school (see [Supplementary Material 1](#) for the seminar plan). The seminar aimed at enhancing preservice teachers' expectancies of success in programming and teaching CT, to reduce their programming anxiety, and to teach them why it is an important topic for primary school children (value components).

The seminar concept was designed by the authors following the six characteristics in the framework suggested by Wigfield et al. (2017). We addressed these characteristics as follows. (1) The seminar had the *content goal* of learning low-threshold programming skills and how to teach CT to primary school children. For every task the preservice teachers had to complete, they received explanations on why the task served that content goal to help students structure their learning and provide them with a sense of purpose (van de Pol et al., 2010; Belland et al., 2013). For example, the lecturer explained why and how a task supported students' programming competences. (2) Preservice teachers needed to complete all tasks to pass the seminar. However, as the seminar was asynchronous, preservice teachers had *control* over when they wanted to work on the tasks and what resources they used. Control was provided to support students' desire for autonomy and prevent them from getting frustrated (Wigfield and Tonks, 2004; van de Pol et al., 2010). (3) The preservice teachers solved ten *hands-on low-threshold programming tasks* that are also suited for primary school students and could be implemented into their classrooms. This might have helped preservice teachers to realize the value of the learning content for their later career, as CT can be implemented into classrooms even with young learners (Angeli et al., 2016; Sengupta et al., 2018; Weber et al., 2021). (4) Tasks were made *interesting* through relating the programming tasks to everyday activities, such as programming a bicycle light or a dice. Throughout the course, it was also explained to the preservice teacher students that these tasks could be implemented directly into future classrooms, which aimed at increasing students' interests (Garet et al., 2001). Moreover, the texts they had to read related to ways of

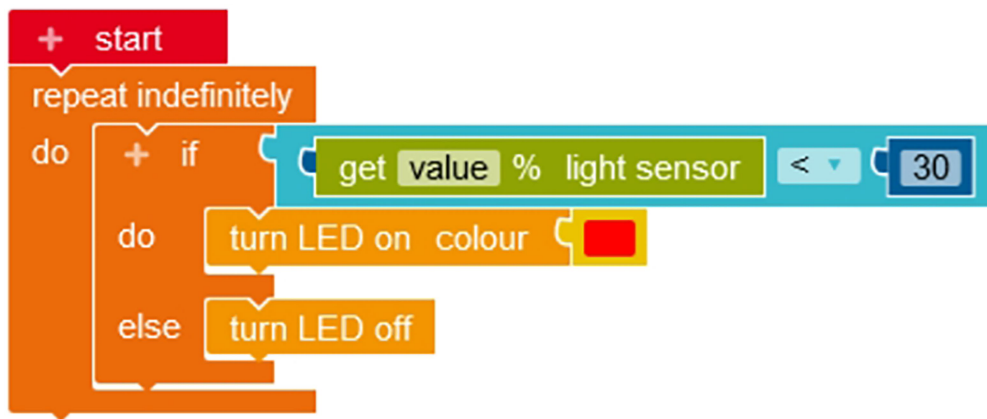


FIGURE 1

Example program for Calliope Mini® using the NEPO® programming language. Reproduced with permission from Open Roberta Lab, Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS.

implementing these tasks into their future classrooms (Angeli et al., 2016; Reuter et al., 2021). (5) Preservice teachers were encouraged to *support each other* during programming and have conversations about the learning content through an online forum implemented into the course. From the beginning of the semester, the preservice teachers were invited to ask all their questions *via* this online forum. Moreover, an online tutorial led by the lecturer allowed students to ask their questions directly to the lecturer and talk to their fellow students (Kong et al., 2018). (6) The lecturer *emphasized the importance of CT* for young children and highlighted that preservice teachers needed to learn it as well (Yadav et al., 2014). Moreover, the students were encouraged to think about reasons why CT might be important for their future profession or their everyday lives to foster their autonomy (Wigfield et al., 2017). This was supported by texts concerned with possibilities to teach CT to young learners.

The seminar was piloted in a prior study focusing on programming anxiety (Weber et al., 2021). Moreover, students in the pilot cohort were asked to evaluate the seminar. Following this procedure, the seminar was slightly altered. First, the theoretical input was altered to focus stronger on scaffolding CT skills. Second, the students received a low-threshold introduction to CT and programming. The students were provided more time to familiarize themselves with the theoretical foundations of CT and received simple logic tasks before starting on the programming tasks.

Programming was taught with problems that could be solved with Calliope Mini®, a microcontroller developed in Germany for primary school children. This controller can be programmed using the block-based programming language NEPO®. In such block-based programming languages, contrary to text-based programming languages, the commands are clustered in blocks that can be matched to form a program, so that codes do not need to be typed out which minimizes potential sources of errors. The microcontroller has multiple features, including different sensors for light, compass, sound, and temperature, can play tunes, and has multiple LED lights.

To use Calliope Mini®, a tablet, computer or another mobile device is necessary. The programming platform Open Roberta Lab<sup>1</sup> can be used for programming. An example program for one of the programming tasks of the seminar (program a bicycle light that lights up automatically when it gets dark) is presented in Figure 1 (for all programming tasks implemented in the seminar and their correct solutions, please see Supplementary Material 2).

To program the bicycle light that lights up automatically in darkness, the following steps need to be taken:

1. Determine value of light sensor.
2. Turn on the lamp if this value is below 30 (if-do-else, <30), otherwise turn off the lamp.

These two steps must be executed repeatedly and indefinitely. Thus, the students needed to program a loop, an if-then-else condition, and an action command.

In the seminar, the preservice teachers first received theoretical explanations about CT and an introduction into programming with Calliope Mini®. For this, the preservice teachers were first left to explore the programming environment by themselves for as long as they liked. Then they received a video with explanations and step-by-step programming instructions. Students were free to pause or rewind the video at any given time.

The lecturer explained why CT is an important competence for primary school children that they can acquire through low-threshold programming tasks. She also informed the students that studies demonstrated that many preservice teachers are skeptical or fearful of programming. She explained that one goal of the seminar was to alleviate such inhibitions. Moreover, the preservice teachers received explanations on the relevance of each assignment for programming and the teaching of CT. During the

<sup>1</sup> <https://lab.open-roberta.org>

course of the semester, the preservice teachers solved ten programming assignments to foster their CT, such as programming a dice, a bicycle lamp, and a metronome. The students each handed in a solution for the programming task in the form of a screenshot. They were free to collaborate on the assignments *via* the provided online forum or through other means of their own choice.

The students received different forms of feedback during the seminar. After handing in their solutions, sample solutions for the programming tasks were provided. Moreover, students could ask questions in the online forum to receive feedback by the peers or ask the lecturer for help. Additionally, the lecturer was available *via* email, online during her office hours as well as during a voluntary online tutorial in which she met with students to discuss their questions about programming and the other seminar topics.

By implementing this seminar, we aimed to investigate whether learning to program with easy assignments and how to teach CT through programming to primary school children could change preservice teachers' expectancies and values towards programming.

The BG did not receive any formal introduction into programming or CT, but only filled out the survey concerned with EVT and solved the CT tasks.

## Measures

All EVT measures were administered as pre- and posttests at the beginning and the end of the 11-week course for the EG. For the BG, the measures were administered as pre- and posttests at a four-week interval. The items for the EVT measures were adapted from the PISA items to measure participants' expectancies, values and anxiety towards programming following Marsh et al. (2019). Each item was rated on a 4-point Likert scale from strongly agree to strongly disagree. All EVT scales were tested for their structure with confirmatory factor analyses (CFAs) with the robust Yuan-Bentler correction. The results showed that all items loaded significantly on their corresponding factor, all standardized loadings > 0.40, all  $p < 0.001$  (Matsunaga, 2010; Eid et al., 2015). Moreover, overall all CFAs showed satisfactory fits (Eid et al., 2015). For more information on the CFAs, please refer to the supplementary materials (Supplementary Material 3).

## Programming expectancies

Programming self-concept focused on programming abilities (Eccles and Wigfield, 1995), and was measured with five items, e.g., *I learn programming quickly*. Cronbach's  $\alpha$  for programming self-concept was  $\alpha = 0.75$  at pretest and  $\alpha = 0.88$  at posttest. Programming self-efficacy relied on Bandura's (1977) conceptualization of self-efficacy as self-confidence in specific tasks. It was measured with four items, of which each concerned a different programming task, e.g., *I think I can program loops*. Cronbach's  $\alpha$  for programming self-efficacy was  $\alpha = 0.90$  at pretest and  $\alpha = 0.91$  at posttest.

## Programming values

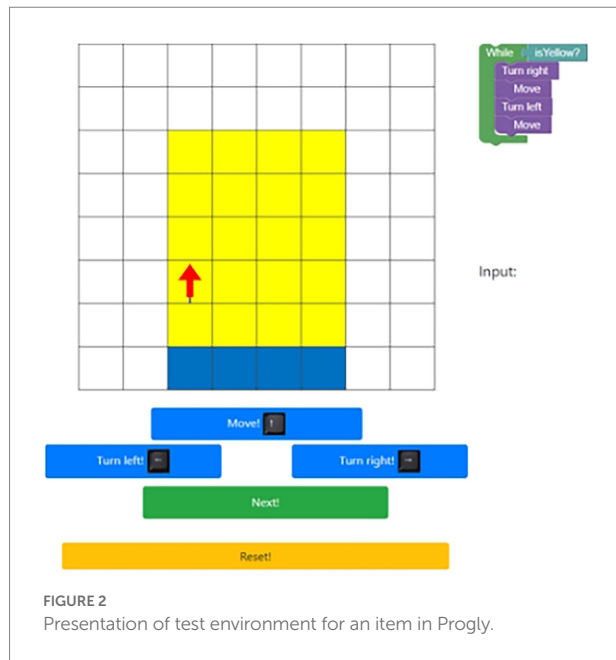
Intrinsic value for programming was construed as the enjoyment of programming (Wigfield et al., 1997; Eccles, 2009), and measured with five items, e.g., *I enjoy programming*. Cronbach's  $\alpha$  for intrinsic value was  $\alpha = 0.85$  at pretest and  $\alpha = 0.92$  at posttest. Moreover, programming utility value measured the value preservice teachers placed on learning to program in relation to their later work as (science) teachers, e.g., *Learning to program is worthwhile for me because I will teach it to children later*. Cronbach's  $\alpha$  for programming utility value was  $\alpha = 0.87$  at pretest and  $\alpha = 0.91$  at posttest. Programming attainment value was conceptualized as preservice teachers' perceived value of children learning to program and was measured with four items, e.g., *It is important to me that children understand basic algorithms*. Cronbach's  $\alpha$  for programming attainment value was  $\alpha = 0.84$  at pretest and  $\alpha = 0.85$  at posttest. The fourth value component, emotional costs, was conceptualized as programming anxiety following Wigfield and Meece's (1988) conceptualization of mathematics anxiety as feelings of worry, stress, and helplessness. It was assessed with six items, e.g., *I worry that it will be difficult for me to solve programming tasks*. Cronbach's  $\alpha$  for programming anxiety was  $\alpha = 0.84$  at pretest and  $\alpha = 0.85$  at posttest.

## Computational thinking

CT was assessed with Progly (Bastian et al., 2021) through programming tasks, which is in line with other studies on CT (Román-González et al., 2017; Shute et al., 2017). Progly is a test system in which the preservice teachers need to solve tracing tasks. It presents the preservice teachers with nine different items which were adapted from a psychometric test developed by Mühling et al. (2015), and validated by Bastian et al. (2021). For each item the test system presents the preservice teachers with an  $8 \times 8$  grid, a figure in the grid, a program snippet and control buttons to move the figure in the grid. The order of the items is presented in Table 1 and an example for one of the items is presented in Figure 2.

TABLE 1 Order of the Progly items measuring CT.

Item	Description
1	Sequence of simple commands
2	Repetition with a fixed number of iterations
3	Two nested repetitions with a fixed number of iterations
4	Repetition with an exit condition
5	Conditional command with a true condition
6	Conditional command with a false condition
7	Conditional command with a repetition with a fixed number of iterations
8	Repetition with a fixed number of iterations in a repetition with an exit condition
9	Repetition with an exit condition in a repetition with a fixed number of iterations



The test assesses the ability to execute common computational concepts like sequences, loops (with and without conditions) or conditional statements (Brennan and Resnick, 2012). During the measurement every interaction with the test system is recorded.

The task for each item is to move the figure exactly like the presented code would move it. The answer, thus, is a sequence of steps. An item is marked as correct if the sequence matches the single correct sequence for the presented code. The test score is the sum of correctly solved items out of the nine items presented. After the measurement the preservice teachers received feedback on how many items they had solved correctly based on the global score. Cronbach's  $\alpha$  for CT was  $\alpha=0.75$ .

## Data analysis

The statistics program R, version 4.2.0 (R Core Team, 2022), was used for data analyses. We used the psych package (Revelle, 2021) for descriptive and correlation analyses, and the lme4 (Bates et al., 2015) and lmerTest packages (Kuznetsova et al., 2017) for the specification of mixed effects models. To investigate the change in the EVT components as well as group differences between the EG and the BG depending on prior CT, we specified mixed-effects models with person on level-2 and time of measurement on level-1.

## Results

The descriptive statistics are presented in Table 2 and the correlations between the constructs are presented in Table 3. The descriptive statistics indicate that both the expectancy as well as the value components increased from pretest to posttest in both groups, while programming anxiety decreased. In the EG,

TABLE 2 Descriptive statistics by condition and point of measurement.

	Experimental group						Baseline group					
	Pretest			Posttest			Pretest			Posttest		
	<i>n</i>	<i>M</i>	<i>SD</i>	<i>n</i>	<i>M</i>	<i>SD</i>	<i>n</i>	<i>M</i>	<i>SD</i>	<i>n</i>	<i>M</i>	<i>SD</i>
SC	152	0.70	0.50	153	1.50	0.49	158	0.50	0.46	158	0.64	0.47
SE	152	0.69	0.66	153	1.58	0.57	158	0.22	0.42	158	0.22	0.39
IV	152	1.34	0.60	153	1.51	0.71	158	0.78	0.55	158	0.87	0.62
UV	152	1.44	0.67	153	1.61	0.61	158	0.66	0.57	158	0.66	0.62
AV	152	1.72	0.61	153	1.87	0.59	158	1.61	0.67	158	1.59	0.69
PA	152	1.73	0.77	153	1.29	0.67	158	1.83	0.66	158	1.68	0.68
CT	153	3.71	2.29	-	-	-	158	3.59	2.30	-	-	-

SC, Self-concept. SE, Self-efficacy. IV, Intrinsic value. UV, Utility value. AV, Attainment value. PA, Programming anxiety. CT, Computational thinking.

self-concept and self-efficacy showed large increases, while the increase in intrinsic value, utility value and attainment value were smaller. Moreover, programming anxiety showed a rather large decrease from pre-to posttest. A similar trend can be observed in the BG. However, increase in self-concept, self-efficacy and intrinsic value was descriptively smaller than in the EG. Moreover, there was no increase in utility value and attainment value and the decrease in programming anxiety was again smaller than in the EG. A *t*-test implied that CT did not differ between the two groups,  $t(308.77)=0.45$ ,  $p=0.653$ .

Moreover, the correlations imply that the expectancy and value components are positively intercorrelated between the two measurement points. This indicates that higher expectancies of success, i.e., a higher self-concept and self-efficacy, are related to higher values as well, i.e., intrinsic, utility and attainment value. Programming anxiety is negatively correlated with the other constructs. CT is negatively correlated with programming anxiety at pretest. This indicates that preservice teachers with high prior CT skills are less likely to be anxious about programming. Furthermore, CT is positively correlated with intrinsic value at pretest. This implies that preservice teachers with high CT are more likely to enjoy programming before taking part in the seminar. CT was positively correlated with self-concept, intrinsic, utility and attainment value, and negatively correlated with programming anxiety at posttest.

To answer the research questions, we examined group differences in change. Differences between participants as indicated by the intraclass correlations explained 16% of variance in self-concept, 23% in self-efficacy, 60% in intrinsic value, 60% in utility value, 63% in attainment value, and 41% in programming anxiety. This indicates that the points of measurement are nested in persons. Thus, we specified mixed-effects models with preservice teachers on level-2 and included time on level-1. The results are presented in Table 4.

Concerning research question 1 for the expectancy components, the EG had a higher self-concept as well as a higher self-efficacy than the BG at pretest. Moreover, in both groups



TABLE 3 Correlations of the constructs.

	SC Pre	SE Pre	IV Pre	UV Pre	AV Pre	PA Pre	CT	SC Post	SE Post	IV Post	UV Post	AV Post
SE Pre	0.62***											
IV Pre	0.54***	0.44***										
UV Pre	0.32***	0.41***	0.65***									
AV Pre	0.20***	0.24***	0.47***	0.57***								
PA Pre	-0.48***	-0.26***	-0.37***	-0.18**	-0.03							
CT	0.09	0.03	0.13*	-0.01	0.09	-0.13*						
SC Post	0.46***	0.43***	0.62***	0.49***	0.21***	-0.30***	0.12*					
SE Post	0.35***	0.48***	0.52***	0.53***	0.19**	-0.18**	0.07	0.80***				
IV Post	0.37***	0.32***	0.67***	0.44***	0.33***	-0.22***	0.20***	0.75***	0.61***			
UV Post	0.28***	0.35***	0.55***	0.65***	0.42***	-0.09	0.13*	0.65***	0.72***	0.68***		
AV Post	0.17**	0.15**	0.39***	0.38***	0.65***	-0.09	0.14*	0.39***	0.38***	0.50***	0.63***	
PA Post	-0.39***	-0.29***	-0.37***	-0.21***	-0.09	0.50***	-0.18**	-0.57***	-0.43***	-0.46***	-0.32***	-0.16**

SC, Self-concept. SE, Self-efficacy. IV, Intrinsic value. UV, Utility value. AV, Attainment value. PA, Programming anxiety. PE, Prior experience with programming. CT, Computational thinking; \* $p < 0.05$ ; \*\* $p < 0.01$ ; \*\*\* $p < 0.001$ .

TABLE 4 Mixed level models.

	Self-concept			Self-efficacy			Intrinsic value			Utility value			Attainment value			Programming anxiety		
	$\rho$			$\rho$			$\rho$			$\rho$			$\rho$			$\rho$		
<b>ICC</b>	<b>0.16</b>			<b>0.23</b>			<b>0.60</b>			<b>0.60</b>			<b>0.63</b>			<b>0.41</b>		
<b>Fixed effects</b>	<b><math>\gamma</math></b>	<b><math>p</math></b>	<b>SE</b>	<b><math>\gamma</math></b>	<b><math>p</math></b>	<b>SE</b>	<b><math>\gamma</math></b>	<b><math>p</math></b>	<b>SE</b>	<b><math>\gamma</math></b>	<b><math>p</math></b>	<b>SE</b>	<b><math>\gamma</math></b>	<b><math>p</math></b>	<b>SE</b>	<b><math>\gamma</math></b>	<b><math>p</math></b>	<b>SE</b>
$\Delta$ Intercept	0.19	0.000	0.05	0.47	0.000	0.06	0.55	0.000	0.07	0.76	0.000	0.07	0.11	0.132	0.07	-0.10	0.207	0.08
BG-EG																		
Time*BG	0.13	0.010	0.05	0.04	0.552	0.06	0.07	0.236	0.06	0.00	0.971	0.06	-0.04	0.515	0.05	-0.15	0.037	0.07
Time*EG	0.79	0.000	0.05	0.89	0.000	0.05	0.16	0.000	0.05	0.17	0.000	0.05	0.15	0.000	0.04	-0.44	0.000	0.05
$\Delta$ Time*BG-EG	0.66	0.000	0.07	0.85	0.000	0.08	0.10	0.208	0.08	0.17	0.044	0.08	0.19	0.009	0.07	-0.29	0.002	0.09
CT	0.02	0.017	0.01	0.02	0.143	0.01	0.04	0.001	0.01	0.01	0.253	0.01	0.03	0.017	0.01	-0.04	0.003	0.01
<b>Random effects</b>	<b>Var</b>	<b>SD</b>		<b>Var</b>	<b>SD</b>		<b>Var</b>	<b>SD</b>		<b>Var</b>	<b>SD</b>		<b>Var</b>	<b>SD</b>		<b>Var</b>	<b>SD</b>	
Person	0.09	0.30		0.07	0.27		0.19	0.44		0.16	0.41		0.25	0.50		0.23	0.48	
Level-1	0.14	0.38		0.22	0.46		0.18	0.43		0.22	0.47		0.15	0.39		0.26	0.51	
Residuum																		

self-concept increased from pretest to posttest, while self-efficacy only increased in the EG. The increase was higher in the EG compared to the BG for both self-concept and self-efficacy.

Research question 2 regarded the value components. The EG showed higher values on intrinsic value and utility value compared to the BG. However, for attainment value, there were no group differences between the EG and the BG at pretest, indicating that both groups viewed it as equally important for children to learn about programming. All three value components increased in the EG, but not in the BG. However, the difference in change between the two groups for intrinsic value was small and thus not significant.

For research question 3 concerned with programming anxiety, at pretest, both groups had equally high programming anxiety. It decreased in both groups from pretest to posttest and the decrease was larger in the EG compared to the BG.

Research question 4 regarded the effect of CT on the change in the EVT components. CT positively affected the change in preservice teachers' self-concepts, intrinsic and attainment values. Moreover, it was negatively related to their programming anxiety, indicating that higher CT skills affected the decrease in emotional costs. These results suggest that higher prior CT skills are related to positive outcomes and underline the need for the implantation of CT into teacher education.

## Discussion

Supporting preservice teachers in acquiring CT skills and preparing them to teach CT to children is of importance (National Research Council, 2010; Yadav et al., 2014; OECD, 2018). This

might be achieved through enhancing their expectancies and values regarding the subject and the teaching of the subject (Wigfield and Guthrie, 2010; Wigfield et al., 2017). However, studies on ways of implementing interventions into higher education that foster expectancies and values towards CT remain sparse.

Therefore, we conducted a study on whether a seminar with low-threshold programming tasks can foster preservice teachers' expectancies of success and values towards teaching CT and programming to young learners. Moreover, we examined whether the seminar decreased their emotional costs, i.e., their programming anxiety. Thus, we compared primary school preservice teachers who attended a mandatory science seminar on CT and programming to special education preservice teachers who did not attend the science seminar and did not learn about CT or programming during their course of study. The results suggest that a science seminar implementing low-threshold programming tasks fostered preservice teachers' expectancies and values towards CT and programming compared to the BG.

Our results implied that expectancies and values were interrelated at pretest as well as at posttest. Moreover, we found interrelations between the constructs between pretest and posttest. These relations are in line with previous research (e.g., Nagengast et al., 2011) and indicate that preservice teachers who expect to be successful in a domain tend to value this domain as well. Programming anxiety was negatively related to the other EVT components. This implies that higher emotional costs come at the expense of expectancies of success as well as values. Thus, preservice teachers who had higher anxiety towards programming, did not expect to do well and might have concluded that programming is not enjoyable or valuable for themselves or their future students (Lee et al., 2013).

For CT, results implied positive relations with self-concept, intrinsic value, utility value and attainment value and a negative relation with programming anxiety. This indicates that preservice teachers who had higher CT skills were more likely to have higher expectancies and values towards programming and were less anxious about it (Dowker et al., 2016; Marsh et al., 2019). Thus, competences were related to expectancies of success and values, as suggested by EVT (Eccles, 2009).

## Group differences at pretest

We found differences at pretest between the primary school preservice teachers who attended the seminar and the special education preservice teachers who did not. Primary school preservice teachers held higher expectancies of success in programming, i.e., self-concept and self-efficacy, and higher intrinsic and utility values towards programming than the special education preservice teachers. A reason for these differences might be that the seminar was a mandatory course and primary school preservice teachers concluded that they will need to teach CT later in their career (Yadav et al., 2014; Jaipal-Jamani and

Angeli, 2017; Butler and Leahy, 2021). Thus, they might have deduced that it is important for them to learn it as well, which might explain the higher intrinsic value and utility value. Moreover, they might have assumed that they will be able to learn it at least to some degree, because the university designed and implemented a seminar for primary school preservice teachers. This implies that the university trusted in their abilities to learn CT and programming, which might explain their higher self-concepts and self-efficacy compared to the special education preservice teachers (Bouck et al., 2021; Faherty et al., 2021). However, primary school preservice teachers' expectancies of success were rather low at the beginning of the semester. None of the participants had gained prior experiences with CT during their studies. This might explain why they viewed themselves as quite unable at programming as a central part of CT. On the other hand, the special education preservice teachers might believe that they will never need to teach CT to children themselves and thus do not value it for themselves and do not expect to be successful in programming (Jaipal-Jamani and Angeli, 2017; Bouck et al., 2021). In Germany, special education teachers mostly take a supporting role and focus on special education needs and pedagogy, while the primary school teachers will mainly be responsible for the teaching of the domain-specific subjects, in our case CT and programming.

However, the two groups did not differ in their attainment value and their emotional costs. This implies that both groups perceived programming as an important learning goal for primary school children and is in line with other studies (Bouck et al., 2021). Moreover, both had high emotional costs, suggesting that even though they think it is important for young children to learn about CT and programming, they are anxious about it themselves. This is in line with results on programming anxiety with computer science students (Nolan and Bergin, 2016) and might be partly explained by a stereotype threat, because most preservice teachers are women, who often tend to be anxious about science activities (Shapiro and Williams, 2012; Marsh et al., 2019).

The low expectancy beliefs, high values and emotional costs suggest that the students need support in CT and programming to prepare them to teach it to their future students. Therefore, the intervention in the form of a specifically designed seminar was called for.

## Change in the EVT components

Research questions 1–3 were concerned with changes in the EVT components during the seminar. The science seminar featured all six characteristics called for by Wigfield et al. (2017). Thus, the science seminar might have changed preservice teachers' expectancies and values.

Expectancies and values increased in the EG, and this increase was higher than in the BG for all EVT components except intrinsic value. This is in line with literature on interventions that target EVT (Guthrie et al., 2004; Wigfield and Guthrie, 2010; Wigfield

et al., 2017). Motivational components are often ignored at the university level, even though fostering expectancies and values is worthwhile, because motivational processes influence learning, plans and persistence (Bong et al., 2012; Lee et al., 2013). This study revealed that supporting expectancies and values can be achieved in an online seminar on CT and programming, which are particularly suitable for an online seminar.

For self-concept and programming anxiety, simply presenting preservice teachers with CT tasks in the form of programming such as Progry (Bastian et al., 2021), as in this study, caused changes for the special education preservice teachers. A reason might be that preservice teachers in the BG thought of programming as writing complex commands and codes. However, a short problem-solving task that involved a graphic programming language might have increased their beliefs that they could solve programming tasks and decreased their anxiety, because they realized that coding is just one part of CT and programming.

This poses the question why self-concept increased in the BG, but self-efficacy did not. Self-concept is a broader construct than self-efficacy (Trautwein et al., 2013). The CT task used in this study is based on programming (Román-González et al., 2017; Shute et al., 2017), but the special education preservice teachers did not code themselves. Thus, it might have increased their self-concepts, i.e., their expectancy that they are at least a little able at programming, but not their more specific self-efficacy that they can solve specific programming tasks such as being able to program loops. To foster self-efficacy beliefs about programming, these beliefs probably need to be addressed more specifically through programming tasks the preservice teachers solve themselves such as implemented in the seminar. This might be the reason that self-efficacy increased in the EG but not the BG. Nevertheless, both self-concept as well as programming anxiety changes were higher for the EG that visited the science seminar and gained more experiences with programming tasks (O'Mara et al., 2006; Dowker et al., 2016).

## Relations with CT

The fourth research question was concerned with the relation of CT with the EVT components. Results implied that CT was positively related with changes in self-concept, intrinsic and attainment value, but negatively related with changes in programming anxiety. As preservice teachers gained experiences with programming during the seminar, their changes in self-concepts probably reflected their abilities more correctly, because they realized that they were able to write low-threshold programs (Nagengast et al., 2011; Marsh et al., 2019). Moreover, CT was related to change in intrinsic value. This indicates that preservice teachers who enjoy programming are more likely to have gained experiences with it and students who are better at CT likely enjoy programming more (Hidi et al., 2017). Furthermore, preservice teachers who had higher CT skills might have had higher confidence in children's ability to learn CT and programming.

Thus, they might have become more open to the idea of children learning CT (Lee et al., 2013).

For the emotional costs, preservice teachers with higher CT were probably less anxious, because they might have been more familiar with problem-solving activities, which programming and CT are a part of (Wing, 2006; Schmader et al., 2008). Thus, higher CT might have increased students' programming enjoyment and decreased their programming anxiety by providing them with confidence about their competences (Lee et al., 2013).

Concluding, the science seminar fostered expectancy and value components and decreased emotional costs for primary school preservice teachers through specifically tailored activities according to the six characteristics suggested by Wigfield et al. (2017).

## Limitations

The limitations of this study concern the implementation of the seminar as well as the implementation of the measurement.

The science seminar for the primary preservice teachers was conducted as an online seminar, because of the Covid-19 pandemic. However, implementing the seminar as an attendance seminar might have an advantage over an online course. In an attendance seminar, the lecturer would be able to cater to the preservice teachers' needs while the students program and could apply support measures to help the preservice teachers learn. Thus, preservice teachers would be able to learn and experience the support measures they could use with children in their own future classrooms. The effect of an online seminar on the EVT components should thus be compared to an attendance seminar in a future study.

The limitations concerning the measures regard (1) that we measured CT only at pretest and (2) that the EVT components were only assessed at two points of measurement. (1) Our goal was to create and implement a science seminar that would foster the EVT components for CT. However, it would be interesting to investigate whether the seminar fostered CT at the same time and how a possible change in CT relates to the EVT component (Eccles, 2009). Thus, change in CT and its relation with expectancies and values could be investigated in a future study. (2) Moreover, it might be worthwhile to implement more points of measurement for the EVT components during the seminar to investigate changes over time. By doing that, it could be investigated at what point in the seminar the EVT components change the most and if this change is linked to certain components proposed by Wigfield et al. (2017).

Our baseline group consisted of M.Ed. special education preservice teachers. The reason for this decision was that the seminar on CT and programming was mandatory and thus, all M.Ed. primary school preservice teachers had to take it. Therefore, it was impossible to recruit primary school preservice teachers in their master programme as a baseline group. Thus, it remains unclear whether there might be a systematic difference between

special education and primary school preservice teachers due to their course of study or because the special education preservice teachers thought that they would never have to teach CT or programming themselves. To investigate this further, the seminar could be made available as special credit for special education preservice teachers. However, the preservice teachers willing to participate in this seminar would probably be the ones with the highest motivation to learn CT and programming. Therefore, they would not be comparable to the primary school preservice teachers who have to attend the seminar.

We chose to target expectancies and values concerning teaching CT and programming through interventions that have content goals; give students control over tasks; include hands-on-activities; interesting tasks; allow collaborations between students; and emphasize the importance of the learning objective. However, it remains unclear whether the characteristics Wigfield and his colleagues (Guthrie et al., 2004; Wigfield and Guthrie, 2010; Wigfield et al., 2017) suggest are even relevant for preservice teachers' expectancies and values towards CT and programming. Maybe primary school preservice teachers' knowledge that they will have to teach CT eventually (Yadav et al., 2014) was enough to change the EVT components. This could be investigated by implementing two additional groups in addition to the one that already exists. In all three groups, the lecturer could explain that the preservice teachers will have to teach CT and programming during their career and that it is important for young children. The first group would follow the suggestions by Wigfield et al. (2017) as implemented in the present study. The second group would receive the same programming tasks but would be left to work on them by themselves and the third group would just receive the information without the programming tasks. Thus, it could be investigated whether the sole knowledge, a combination of knowledge and programming or the specifically designed seminar change the EVT components. The need for more seminars is a methodological issue that we could not investigate in this study, as this methodological desideratum is not feasible for the third group that would just receive information. Moreover, concerning the second group that would receive information and tasks, such an approach is not desirable, because there are more engaging ways to teach CT and programming, e.g., the seminar in this study.

Nevertheless, our study shows that students' expectancies and values can be fostered and their emotional costs decreased during a science seminar following the suggestions by Wigfield et al. (2017). This can be achieved by using low-threshold programming tasks that offer the students ways to engage with CT and prepare them to teach it to primary school children in their own classrooms.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## Ethics statement

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. The patients/participants provided their written informed consent to participate in this study.

## Author contributions

AW, MB, and ML contributed to the conception and the design of the study. AW organized the database, performed the statistical analysis, and wrote the first draft of the manuscript. AW, MB, VB, AM, and ML wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

## Funding

This research was funded by the Deutsche Telekom Stiftung and the German Federal Ministry of Education and Research, Grant Number 01JA1605.

## Acknowledgments

We thank Sabrina Stiel-Dämmer for her valuable comments on the design of the seminar and specifically on the introduction to computational thinking. Moreover, we also thank Miriam Hellriegel for her help with data collection.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fpsyg.2022.987761/full#supplementary-material>

## References

- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., et al. (2016). A K-6 computational thinking curriculum framework: implications for teacher knowledge. *Educ. Technol. Soc.* 19, 47–57. Available at: <https://www.jstor.org/stable/jeductechsoci.19.3.47>
- Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioral change. *Psychol. Rev.* 84, 191–215. doi: 10.1037/0033-295X.84.2.191
- Barr, V., and Stephenson, C. (2011). Bringing computational thinking to K-12. *ACM Inroads* 2, 48–54. doi: 10.1145/1929887.1929905
- Bastian, M., Schneider, Y., and Mühlhng, A. (2021). Diagnose von Fehlvorstellungen anhand von Programmabläufen in einem elektronischen Testsystem [diagnosis of misconceptions based on program sequences in an electronic test system]. *Empirische Pädagogik* 35, 72–92.
- Bates, D., Maechler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *J. Stat. Softw.* 67, 1–48. doi: 10.18637/jss.v067.i01
- Belland, B. R., Kim, C., and Hannafin, M. J. (2013). A framework for designing scaffolds that improve motivation and cognition. *Educ. Psychol.* 48, 243–270. doi: 10.1080/00461520.2013.838920
- Bers, M. U. (2018). Coding and computational thinking in early childhood: the impact of ScratchJr in Europe. *Eur. J. STEM Educ.* 3:8. doi: 10.20897/ejsteme/3868
- Bescherer, C., and Fest, A. (2019). “Mathematik und informatische Bildung. Programmieren mit scratch [mathematics and computer science education. Programming with scratch]” in *Schriftenreihe Medienpädagogik interdisziplinär: Vol. 12. Digitale Medien in der Grundschullehrerbildung. Erfahrungen aus dem Projekt dileg-SL [Digital media in primary teacher education. Experiences from the dileg-SL project.]*, eds. T. Junge and N. Horst (München: Verlag kopaed), 117–130.
- Bong, M., Cho, C., Ahn, H. S., and Kim, H. J. (2012). Comparison of self-beliefs for predicting student motivation and achievement. *J. Educ. Res.* 105, 336–352. doi: 10.1080/00220671.2011.627401
- Bouck, E. C., Sands, P., Long, H., and Yadav, A. (2021). Preparing special education preservice teachers to teach computational thinking and computer science in mathematics. *Teach. Educ. Spec. Educ.* 44. doi: 10.1177/0888406421992376
- Boulden, D. C., Rachmatullah, A., Oliver, K. M., and Wiebe, E. (2021). Measuring in-service teacher self-efficacy for teaching computational thinking: development and validation of the T-STEM CT. *Educ. Inf. Technol.* 26, 4663–4689. doi: 10.1007/s10639-021-10487-2
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., and Barone, D. (2017). “Development of computational thinking skills through unplugged activities in primary school” in *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, ed. E. Barendsen (New York: ACM), 65–72.
- Brennan, K., and Resnick, M. (2012). “Using artifact-based interviews to study the development of computational thinking in interactive media design” in *Paper Presented at Annual American Educational Research Association Meeting (Vancouver: AME Research Group)*.
- Butler, D., and Leahy, M. (2021). Developing preservice teachers’ understanding of computational thinking: A constructionist approach. *Br. J. Educ. Technol.* 52, 1060–1077. doi: 10.1111/bjet.13090
- Curzon, P., Dorling, M., Ng, T., Selby, C., and Woollard, J. (2014). *Developing Computational Thinking in the Classroom: A Framework*. Swindon: Computing at School.
- Di Lieto, M. C., Inguaggiato, E., Castro, E., Cecchi, F., Cioni, G., Dell’Omo, M., et al. (2017). Educational robotics intervention on executive functions in preschool children: a pilot study. *Comput. Hum. Behav.* 71, 16–23. doi: 10.1016/j.chb.2017.01.018
- Dickes, A. C., Sengupta, P., Farris, A. V., and Basu, S. (2016). Development of mechanistic reasoning and multilevel explanations of ecology in third grade using agent-based models. *Sci. Educ.* 100, 734–776. doi: 10.1002/sc.21217
- Dowker, A., Sarkar, A., and Looi, C. Y. (2016). Mathematics anxiety: what have we learned in 60 years? *Front. Psychol.* 7:508. doi: 10.3389/fpsyg.2016.00508
- Eccles, J. S. (2009). Who am I and what am I going to do with my life? Personal and collective identities as motivators of action. *Educ. Psychol.* 44, 78–89. doi: 10.1080/00461520902832368
- Eccles, J. S., Adler, T. F., Futterman, R., Goff, S. B., Kaczala, C. M., Meece, J. L. [J. L.], et al. (1983). “Expectancies, values, and academic behaviors” in *Achievement and Achievement Motivation*, ed. J. T. Spence (New York: W. H. Freeman), 75–146.
- Eccles, J. S., and Wigfield, A. (1995). In the mind of the actor: the structure of adolescents’ achievement task values and expectancy-related beliefs. *Pers. Soc. Psychol. Bull.* 21, 215–225. doi: 10.1177/0146167295213003
- Eid, M., Gollwitzer, M., and Schmitt, M. (2015). *Statistik und Forschungsmethoden: Mit Online-Materialien (Vol. 4, überarbeitete und erweiterte Auflage)*. Weinheim: Beltz.
- Faber, H. H., Wiersma, M. D., Doornbos, R. P., van der Ven, J. S., and de Vette, K. (2017). Teaching computational thinking to primary school students via unplugged programming lessons. *J. Eur. Teach. Educ. Netw.* 12, 13–24.
- Faherty, R., Quille, K., Vivian, R., McGill, M. M., Becker, B. A., and Nolan, K. (2021). “Comparing programming self-esteem of upper secondary school teachers to CS1 students.” in *ITIcSE*, Dublin; June 2021
- Feierabend, S., Rathgeb, T., and Reutter, T. (2019). KIM-Studie 2018 Basisuntersuchung zum Medienumgang 6-bis 13-Jähriger KIM Studie 2018. Kindheit, Internet, Medien. Basisuntersuchung zum Medienumgang 6-bis 13-Jähriger. Available at: [https://www.mpfs.de/fileadmin/files/Studien/KIM/2018/KIM-Studie\\_2018\\_web.pdf](https://www.mpfs.de/fileadmin/files/Studien/KIM/2018/KIM-Studie_2018_web.pdf) (Accessed August 25, 2022).
- Gal-Ezer, J., and Stephenson, C. (2010). Computer science teacher preparation is critical. *ACM Inroads* 1, 61–66. doi: 10.1145/1721933.1721953
- Garet, M. S., Porter, A. C., Desimone, L., Birman, B. F., and Yoon, K. S. [J. K. S.]. (2001). What makes professional development effective? Results from a national sample of teachers. *Am. Educ. Res. J.* 38, 915–945. doi: 10.3102/00028312038004915
- Guthrie, J. T., Wigfield, A., Barbosa, P., Perencevich, K. C., Taboada, A., Davis, M. H., et al. (2004). Increasing reading comprehension and engagement through concept-oriented reading instruction. *J. Educ. Psychol.* 96, 403–423. doi: 10.1037/0022-0663.96.3.403
- Hidi, S. E., Renninger, K. A., and Northoff, G. (2017). “The development of interest and self-related processing” in *International Advances in Self Research. Self Driving Positive Psychology and Well-Being*, Vol. 6 eds. F. Guay, H. W. Marsh, D. M. McInerney and R. G. Craven (Charlotte: Information Age Publishing), 51–70.
- Hsu, T.-C., Chang, S.-C., and Hung, Y.-T. (2018). How to learn and how to teach computational thinking: suggestions based on a review of the literature. *Comput. Educ.* 126, 296–310. doi: 10.1016/j.compedu.2018.07.004
- Jaipal-Jamani, K., and Angeli, C. (2017). Effect of robotics on elementary preservice teachers’ self-efficacy, science learning, and computational thinking. *J. Sci. Educ. Technol.* 26, 175–192. doi: 10.1007/s10956-016-9663-z
- Kazakoff, E. R., Sullivan, A., and Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Educ. J.* 41, 245–255. doi: 10.1007/s10643-012-0554-5
- Kong, S. C., Chiu, M. M., and Lai, M. (2018). A study of primary school students’ interest, collaboration attitude, and programming empowerment in computational thinking education. *Comput. Educ.* 127, 178–189. doi: 10.1016/j.compedu.2018.08.026
- Kuznetsova, A., Brockhoff, P. B., and Christensen, R. H. B. (2017). lmerTest package: tests in linear mixed effects models. *J. Stat. Softw.* 82, 1–26. doi: 10.18637/jss.v082.i13
- Lamprou, A., and Repenning, A. (2018). “Teaching how to teach computation thinking”, In *Proceedings of the 23rd Annual ACM Conferences on Innovation and Technology in Computer Science Education (ITIcSE 2018)*, (pp. 69–74). New York: Association for Computing Machinery.
- Lee, J., Lee, M., and Bong, M. (2013). “High value with low perceived competence as an amplifier of self-worth threat” in *International Advances in Self Research. Theory Driving Research: New Wave Perspectives on Self-Processes and human Development*, eds. D. M. McInerney, H. W. Marsh and R. G. Craven (Charlotte: Information Age Publishing Inc.), 205–232.
- Marsh, H. W., van Zanden, B., Parker, P. D., Guo, J., Conigrave, J., and Seaton, M. (2019). Young women face disadvantage to enrollment in university STEM coursework regardless of prior achievement and attitudes. *Am. Educ. Res. J.* 56, 1629–1680. doi: 10.3102/0002831218824111
- Marsh, H. W., Xu, M., and Martin, A. J. (2012). “Self-concept: a synergy of theory, method, and application” in *APA Educational Psychology Handbook: Vol. 1. Theories, Constructs, and Critical Issues*, eds. K. R. Harris, S. Graham and T. Urdan (Washington, D.C.: American Psychological Association), 427–458.
- Matsunaga, M. (2010). How to factor-analyze your data right: do’s, don’ts, and how-to’s. *Int. J. Psychol. Res.* 3, 97–110. doi: 10.21500/20112084.854
- Ministerium für Bildung, Wissenschaft, Jugend und Kultur (Ed.). (2018). *Teilrahmenplan Sachunterricht: Richtlinie zur Digitalen Bildung in der Primarstufe [Sub-Framework for Science Teaching: Guideline for digital Education in Primary Education]*. Ministerium für Bildung, Wissenschaft, Jugend und Kultur.
- Mühlhng, A., Ruf, A., and Hubwieser, P. (2015). “Design and first results of a psychometric test for measuring basic programming abilities.” in *Proceedings of the Workshop in Primary and Secondary Computing Education*, 2–10.
- Musu-Gillette, L. E., Wigfield, A., Harring, J. R., and Eccles, J. S. (2015). Trajectories of change in students’ self-concepts of ability and values in math and college major choice. *Educ. Res. Eval.* 21, 343–370. doi: 10.1080/13803611.2015.1057161

- Nagengast, B., Marsh, H. W., Scalas, L. F., Xu, M. K., Hau, K.-T., and Trautwein, U. (2011). Who took the “x” out of expectancy-value theory? A psychological mystery, a substantive-methodological synergy, and a cross-national generalization. *Psychol. Sci.* 22, 1058–1066. doi: 10.1177/0956797611415540
- National Research Council (2010). *Committee for the Workshops on Computational Thinking: Report of a Workshop on the scope and Nature of Computational Thinking*. Washington, DC: National Academies Press
- Newell, A., Shaw, J. C., and Simon, H. A. (1959). “A variety of intelligent learning in a general problem solver” in *Paper Presented at the Interdisciplinary Conference on Self-Organizing Systems* (Chicago: Illinois).
- Nolan, K., and Bergin, S. (2016). “The role of anxiety when learning to program: a systematic review of the literature” in *Koli Calling '16 Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (Koli National Park: Finland).
- Nugent, G., Barker, B., Welch, G., Grandgenett, N., Wu, C., and Nelson, C. (2015). A model of factors contributing to STEM learning and career orientation. *Int. J. Sci. Educ.* 37, 1067–1088. doi: 10.1080/09500693.2015.1017863
- OECD (2018). “Computational thinking” in *Educational Research and Innovation. Teachers as Designers of Learning Environments: The Importance of Innovative Pedagogies*. eds. A. Paniagua and D. Istance (Berlin: OECD Publishing), 101–108.
- OECD (2021). *Education at a Glance 2021: OECD Indicators*. Berlin: OECD Publishing.
- O’Mara, A. J., Marsh, H. W., Craven, R. G., and Debus, R. L. (2006). Do self-concept interventions make a difference, a synergistic blend of construct validation and meta-analysis. *Educ. Psychol.* 41, 181–206. doi: 10.1207/s15326985ep4103\_4
- R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing [Computer software]. Vienna
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: programming for all. *Commun. ACM* 52, 60–67. doi: 10.1145/1592761.1592779
- Reuter, T., Weber, A. M., Nitz, S., and Leuchter, M. (2021). Problemlösen in digitalen Kontexten [Problem solving in digital contexts]. *Empirische Pädagogik* 35, 5–18.
- Revelle, W. (2021). *psych: Procedures for Psychological, Psychometric, and Personality Research (Version R package version 2.1.3) [Computer software]*, Evanston, IL: Northwestern University.
- Rich, P. J., Larsen, R. A., and Mason, S. L. (2021). Measuring teacher beliefs about coding and computational thinking. *J. Res. Technol. Educ.* 53, 296–316. doi: 10.1080/15391523.2020.1771232
- Román-González, M., Pérez-González, J.-C., and Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Comput. Hum. Behav.* 72, 678–691. doi: 10.1016/j.chb.2016.08.047
- Schmader, T., Johns, M., and Forbes, C. (2008). An integrated process model of stereotype threat effects on performance. *Psychol. Rev.* 115, 336–356. doi: 10.1037/0033-295X.115.2.336
- Sengupta, P., Dicks, A. C., and Farris, A. V. (2018). “Toward a phenomenology of computational thinking in K-12 STEM” in *Computational Thinking in STEM Discipline: Foundations and Research Highlights*. ed. M. S. Khine (Berlin: Springer) doi: 10.1007/978-3-319-93566-9\_4
- Shapiro, J. R., and Williams, A. M. (2012). The role of stereotype threats in undermining girls’ and women’s performance and interest in STEM fields. *Sex Roles* 66, 175–183. doi: 10.1007/s11199-011-0051-0
- Shute, V. J. (1991). Who is likely to acquire programming skills. *J. Educ. Comput. Res.* 7, 1–24. doi: 10.2190/VQJD-T1YD-5WVB-RYPJ
- Shute, V. J., Sun, C., and Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educ. Res. Rev.* 22, 142–158. doi: 10.1016/j.edurev.2017.09.003
- Trautwein, U., Nagengast, B., Marsh, H. W., Gaspard, H., Dicke, A.-L., Lüdtke, O., et al. (2013). “Expectancy-value theory revisited: From expectancy-value theory to expectancy-values theory” in *International Advances in Self Research. Theory Driving Research: New Wave Perspectives on Self-Processes and human Development*. eds. D. M. McInerney, H. W. Marsh and R. G. Craven (Charlotte: Information Age Publishing Inc.), 233–249.
- van de Pol, J., Volman, M., and Beishuizen, J. (2010). Scaffolding in teacher–student interaction: a decade of research. *Educ. Psychol. Rev.* 22, 271–296. doi: 10.1007/s10648-010-9127-6
- Weber, A. M., Barkela, V., Stiel-Dämmer, S., and Leuchter, M. (2021). Der Zusammenhang emotionaler Kosten bei Grundschullehrerstudierenden mit ihrer informatischen Problemlösekompetenz [The relation of primary school student teachers’ emotional costs and their informatic problem solving]. *Empirische Pädagogik* 35, 93–111.
- White, G., and Sivitanides, M. (2003). An empirical investigation of the relationship between success in mathematics and visual programming courses. *Journal of information systems education. J. Inf. Syst. Educ.* 14, 409–416.
- Wigfield, A., and Cambria, J. (2010). Students’ achievement values, goal orientations, and interest: definitions, development, and relations to achievement outcomes. *Dev. Rev.* 30, 1–35. doi: 10.1016/j.dr.2009.12.001
- Wigfield, A., Eccles, J. S., Yoon, K. S. [K. S.], Harold, R. D., Arbreton, A. J. A., Freedman-Doan, C., et al. (1997). Change in children’s competence beliefs and subjective task values across the elementary school years: a 3-year study. *J. Educ. Psychol.* 89, 451–469. doi: 10.1037/0022-0663.89.3.451
- Wigfield, A., Gladstone, J. R., and Turci, L. (2017). “Development of children’s ability beliefs and values and school-based interventions to improve them” in *International Advances in Self Research: Vol. 6. Self. Driving Positive Psychology and Well-Being*. eds. F. Guay, H. W. Marsh, D. M. McInerney and R. G. Craven (Charlotte: Information Age Publishing), 283–308.
- Wigfield, A., and Guthrie, J. T. (2010). “The impact of concept-oriented reading instruction on students’ reading motivation, reading engagement, and reading comprehension” in *Handbook of Research on Schools, Schooling, and Human Development*. eds. J. L. Meece and J. S. Eccles. 1st ed (Milton Park: Routledge), 463–477.
- Wigfield, A., Mason-Singh, A., Ho, A. N., and Guthrie, J. T. (2014). “Intervening to improve children’s reading motivation and comprehension: concept-oriented reading instruction” in *Advances in Motivation and Achievement: Vol. 18. Motivational Interventions, 1st Edn.* ed. S. A. Karabenick (Bingley, Emerald), 37–70.
- Wigfield, A., and Meece, J. L. (1988). Math anxiety in elementary and secondary school students. *J. Educ. Psychol.* 80, 210–216. doi: 10.1037/0022-0663.80.2.210
- Wigfield, A., and Tonks, S. [S.]. (2004). “The development of motivation for reading and how it is influenced by CORI” in *Motivating Reading Comprehension: Concept-Oriented Reading Instruction*. eds. J. T. Guthrie, A. Wigfield and K. C. Perencevich (New York, NY: L. Erlbaum Associates), 249–272.
- Wing, J. M. (2006). Computational thinking. *Commun. ACM* 49, 33–35. doi: 10.1145/1118178.1118215
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., and Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Trans. Comput. Educ.* 14, 1–16. doi: 10.1145/2576872
- Yadav, A., Stephenson, C., and Hong, H. (2017). Computational thinking for teacher education. *Commun. ACM* 60, 55–62. doi: 10.1145/2994591
- Zhang, L., and Nouri, J. (2019). A systematic review of learning computational thinking through scratch in K-9, 103607. *Comput. Educ.* 141. doi: 10.1016/j.compedu.2019.103607
- Zhao, M., Zhao, M., Wang, X.-H., and Ma, H.-L. (2020). “Promoting teaching self-efficacy in computational thinking of school teachers” in *Ninth International Conference of Educational Innovation through Technology (EITT)* (Porto: Portugal).