
Received: 21 March 2025, Accepted: 20 November 2025

DOI: <https://doi.org/10.26529/cepsj.2110>

Primary Programming: Teachers' Attitudes and Skills in the Light of Computing Reform

JIŘÍ VANÍČEK*¹ AND JAN PRŠALA²

Two global trends that can be observed in computing education all over the world are moving the beginning of teaching computing as a compulsory school subject to primary education and moving from the teaching of user approaches to digital technologies to computer science content. The Czech Republic is currently the scene of such changes within the ongoing reform of informatics education. This paper presents the visions and principles that have served as the foundational framework for the reform initiative. A significant term for the introduction of computing in schools is programming. The key figure for the implementation of the changes is the primary school teacher who has no background in computer science and no experience in the subject as a student. The topic of our inquiry is therefore the attitudes of primary teachers towards teaching programming at the time of the ongoing school reform. The aim of the research is to identify the personal, pedagogical and environmental factors that influence teachers' attitudes towards teaching programming. We measured programming attitudes using Sun's Teacher Programming Attitude Scale, while the relationship of these attitudes to teachers' computational thinking was explored using Bebras Challenge tasks. A survey of primary school teachers revealed a positive attitude towards the teaching of programming, coupled with very good computing skills. The study also found that previous experience in teaching programming is a significant factor in influencing teachers' positive attitudes towards the subject.

Keywords: national curricula innovation, programming, primary teacher, attitudes, programming skills

1 *Corresponding Author. Faculty of Education at the University of South Bohemia, České Budějovice, Czech Republic; vanicek@pf.jcu.cz.

2 Faculty of Education at the University of Hradec Králové, Hradec Králové, Czech Republic.

Programiranje v osnovni šoli: odnos in veščine učiteljev v luči reforme poučevanja računalništva in informatike

JIŘÍ VANÍČEK IN JAN PRŠALA

☞ Dva globalna trenda, ki ju lahko opazimo v računalniškem izobraževanju po vsem svetu, sta premik začetka poučevanja računalništva kot obveznega šolskega predmeta v osnovnošolsko izobraževanje in prehod od poučevanja uporabe digitalnih tehnologij k obravnavi vsebin računalništva. Češka te spremembe uresničuje s sistemsko reformo poučevanja računalništva in informatike. Članek predstavlja vizije in načela, ki so oblikovali temeljni okvir reforme. Pomemben vidik uvedbe računalništva in informatike v šolah je programiranje. Ključna oseba za izvajanje sprememb je osnovnošolski učitelj, ki pogosto nima ustreznega znanja iz računalništva in izkušenj s tem predmetom kot učenec. Raziskava se osredinja na odnos osnovnošolskih učiteljev do poučevanja programiranja med šolsko reformo. Cilj raziskave je identificirati osebne, pedagoške in okoljske dejavnike, ki vplivajo na njihov odnos do poučevanja programiranja. Odnos do programiranja smo merili z uporabo Sunove lestvice odnosa učiteljev do programiranja, medtem ko smo raziskali povezavo med temi odnosi in računalniškim mišljenjem učiteljev z uporabo nalog tekmovanja Bober. Raziskava med osnovnošolskimi učitelji je pokazala pozitiven odnos do poučevanja programiranja, ki ga spremljata dobro računalniško znanje in zadostna raven računalniškega mišljenja. Študija je tudi ugotovila, da so pretekle izkušnje s poučevanjem programiranja pomemben dejavnik, ki vpliva na pozitiven odnos učiteljev do tega predmeta.

Ključne besede: inovativnost v nacionalnih učnih načrtih, programiranje, osnovnošolski učitelj, odnos, programerske veščine

Introduction

The present study examines the implementation of a paradigm shift in the realm of information and communication technology education towards informatics education. The focus is on the shift from a user-oriented approach to the mandatory inclusion of informatics in the curriculum at primary level. The paper describes two steps to this implementation, together with the ideas and sources of inspiration for these steps, as well as the changes in the system. To illustrate the outcomes of these changes, we present the findings of a survey conducted among primary school teachers on their programming attitudes and skills several years after the reform's inception.

The terms used in the paper are defined as follows:

Information and communication technology (ICT) is used in the sense of teaching a user approach to digital technology, the area for the development of digital literacy and everyday use of computers as a learning tool. The acronym ICT is employed for the designation of the compulsory school subject of Information and Communication Technology, as it was previously designated until 2021.

Computing is used as the new name of the school subject in the sense of computer science or informatics, which is one of the STEM disciplines. It focuses on an author-oriented approach to digital technology (not only consumption, but creation) with an emphasis on understanding computers and development of computational thinking. In the article, we distinguish between computing, which is related to the school subject, and computer science as a scientific field.

Primary education (ISCED1, age 6–11 in Czechia), lower-secondary education (ISCED2, age 11–15), upper-secondary education (ISCED3, age 15–19).

Two steps to primary computing in Czechia

As civilisation becomes increasingly aware of the expanding potential for the integration of computers and automation into all facets of our lives, and the consequences for prosperity, the digital world is assuming an ever-larger share of our environment. Consequently, as Gander (2014, p. 4) asserts, computer education has become an integral component of general education. Three global trends have been observed in the field of computer and computing education over the past few decades: a shift from a user approach in computer training, from controlling the computer to understanding the principles on which it works; the transformation of the school subject dedicated to this area

from an optional to a compulsory subject; and a shift in the start of the teaching in this field towards a younger age.

The Tucker committee for the ACM K-12 curriculum claims that “the goals of a K-12 computer science curriculum are to introduce the fundamental concepts of computer science to all students, beginning at the elementary school level” (Tucker, 2003, p. 6). Gander (2014, p. 7) states the formula “*Computer science in Schools = Digital Literacy + Informatics*”, where informatics means “the science behind information technology”.

First step: Compulsory primary ICT education

Czech professional institutions responsible for running education, such as the Panel for ICT Teaching Innovations at the Pedagogical Research Institute, noted a growing discrepancy between the aforementioned trends and the reality in schools. This led to the implementation of a reform that can be described as consequentialist in accordance with Kodelja (2021). The practical implementation of these trends was reflected in two significant steps that dramatically altered the situation in Czech schools with regard to school legislation. Since the early 1990s, ICT including computing had been incorporated into the curriculum of Czech secondary schools as a non-compulsory topic within other educational areas such as mathematics or technology. Subsequent to the Czech Republic's accession to the European Union in 2004, the nation's Education Act (MŠMT, 2005) came into force in 2006, introducing compulsory ICT education at primary and lower secondary levels. The main focus of this subject was the search, processing and utilisation of information and communication (NÚV, 2005), while the pedagogical framework was rooted in office software training, file management, Internet browsing and creating graphics. Notably, the curriculum did not incorporate any computer science components, reflecting the UNESCO model of ICT development in the 2002 curriculum (UNESCO, 2002). The gradual introduction of the subject resulted in a situation where, after 2011, ICT was taught on a compulsory basis in all schools, with one lesson per week in one grade at the primary level.

At the lower secondary level, computer science activities were included in the curricula in some schools; for example, in the form of programming courses in a children's environment such as Logo or Baltie. In practice, however, this did not actually occur in primary schools, where the ICT subject was mainly used to introduce work with computers. The support provided by textbooks and teaching materials, as well as teacher training in the field, was weak and unsystematic. The vast majority of schools lacked computer labs suitable for younger pupils, and ICT was typically taught by secondary school teachers

rather than primary specialists. It should be noted that this model did not stipulate the use of digital technology in other school subjects, so computers were frequently employed solely in the context of the ICT subject, resulting in the isolation of ICT and the somewhat impractical use of computers. This model remained unchanged until 2021.

Models of inspiration for the second step

In order to develop a model for innovating teaching in this area in the Czech Republic, an analysis of international best practice was conducted. A study was undertaken of the situation in Slovakia, which introduced computing as a compulsory subject before 2010 with a focus on algorithmic thinking, procedures, problem solving, digital technologies and creativity in the user approach to digital technology (Blaho & Salanci, 2011; Blaho, 2012). The Slovakian approach – as demonstrated by the introduction of a national curriculum, educational content, teacher support, pedagogical research, and the establishment of a community of educators through the project Infovek and the DidInfo conference (UMB, 2024) – provided a valuable model for the development of informatics education in Czechia. The introduction of a new computing curriculum in Poland was also examined (Sysło, 2015). The study *Shut down or restart?*, which established the subject Computing in the United Kingdom, claimed that “ICT and informatics, which is a scientific discipline similar to mathematics and physics, are two different subjects in school education with different missions and functions, although they have common areas of synergy” (The Royal Society, 2012, p. 10). Instead of the term ICT, the concept of digital literacy was introduced, which also inspired the second step.

Conception of the second step: The strategy of digital education

The introduction of computing was facilitated by the creation of a new digital education strategy around 2014. During expert discussions involving computer specialists, educators and teachers, the public, schools, employers and IT companies, a section on the development of computational thinking was added to the original discussion on innovations in digital literacy. Subsequently, the Czech government adopted a strategic document entitled *Strategy for Digital Education until 2020*, which set out three primary objectives: “the development of computational thinking, the development of digital literacy, and the opening of education to new methods and ways of learning through digital technologies” (MŠMT, 2014).

The introduction of computational thinking in the Czech Republic was justified as enabling students to understand today’s world from the computer

science perspective and to acquire the skills required to resolve a wide array of issues that arise in the context of automation. This document initiated a series of actions that, over the subsequent six years, led to the transformation of the school and computer-assisted and computer education, the implementation of which we are currently experiencing.

Computational thinking and digital literacy

The utilisation of the prevailing terms ICT and informatics proved to be insufficient to capture the necessary substantive changes in the pedagogical context. These terms were commonly perceived as semantically equivalent, with an overreliance on the content aspect of education. There was a real risk that teachers would think of the current ICT under the term informatics and nothing would change in their teaching. The necessity for novel terminology therefore emerged, with the objective of shifting the focus towards the objectives of education and the cultivation of the individual. With the aid of the concepts of computational thinking and digital literacy, a model of change was developed, ultimately resulting in a complete change to the compulsory subject, which shifted from a focus on digital literacy to one that is more aligned with computational thinking (Vaniček, 2021).

The employment of these two terms facilitated a more precise articulation of the impending alterations within the framework of the planned reform. This approach precluded a scenario in which, by simply adding computer science topics to the existing ICT teaching, educators would impart substantially similar content, thereby ensuring the preservation of the fundamental essence of teaching, which would be predicated on the user approach. Consequently, computer science would become a mere superstructure that would not receive enough attention in the eyes of teachers, the presumed main drivers of the reform. The utilisation of these concepts enabled a more profound elucidation of the radical transition from the user to the author approach, while concomitantly providing a more comprehensive description of the state curriculum.

Figure 1

The content of subjects in relation to computer education before and after the planned innovation. The green arrows show the main intended content shift among school subjects

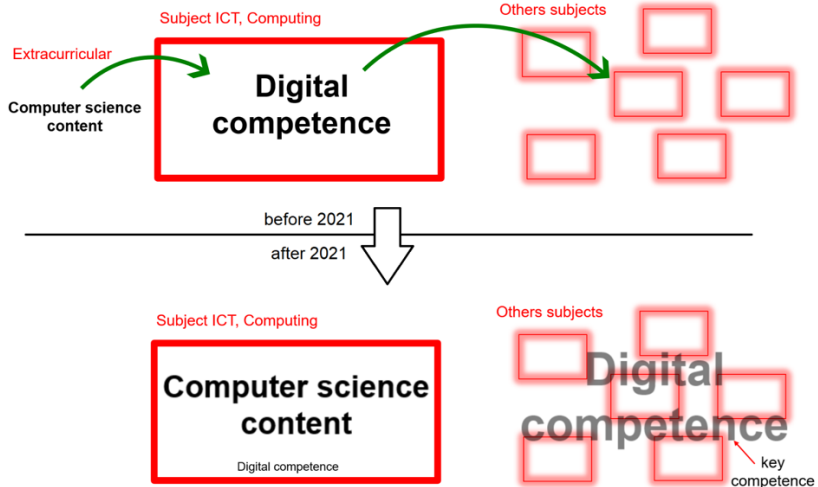


Figure 1 illustrates these changes. The big square represents the compulsory subject “computers” and the small squares represent the other school subjects. The text describes content related to computer education. The empty area on the left represents extracurricular education, organised outside of school or, rarely, in non-compulsory parts of school education.

The creation of the conditions for innovation in education and the development of computational thinking was the aim of a major strategic project, PRIM, in the years 2017–2020. All nine pedagogical faculties in the Czech Republic involved in the training of primary school teachers collaborated on the implementation of the necessary modifications to the undergraduate education of ICT teachers, including primary school teachers. These modifications included the creation of new educational content in the form of sets of freely accessible textbooks, the establishment of a system of further education for teachers for the so-called *New Informatics*, and the promotion of computing in society (JU, 2017).

New national informatics curriculum

In January 2021, the Ministry of Education published the new educational framework for primary education (MŠMT, 2021). This document defines the national curriculum, which includes the new concept of computer science.

The experience from the time of the Covid pandemic and the long period of online distance learning significantly contributed to the development of Czech teachers' digital competence (Pavlas et al., 2021) and subsequently increased their confidence in this area. This illustrates the possibilities that making and digital fabrication can offer to innovation, and exemplifies the meaningful integration of technology in educational contexts in terms of learning (Bosco et al, 2019, p. 69). Teaching is now seen as a mediated action with high potential for hybridisation (Perla et al, 2025, p. 149). The reform also addresses the readiness for the transition to the new computing curricula: the existence of pilot textbooks, in-service teacher training, and the availability of finances for the purchase of robotic learning tools assisted with the implementation. The subject of ICT was renamed Computing, with a compulsory minimum of one lesson per week in each of grades 4 to 9, which is three times more than the previous time allocation (see Figure 2).

Figure 2

Time allocation for the compulsory subject before and after curricular reform

		<2021	>2021
level	grade	ICT	Computing
primary	1	1	
	2		
	3		
	4		1
	5		1
lower-secondary	6	1	1
	7		1
	8		1
	9		1

The content of the subject was changed completely. It is now divided into four areas, which we illustrate by selected expected outcomes when finishing primary education in the fifth grade in Figure 3.

Computing was characterised as a subject that helps pupils understand the computer and the world around us from the informatics perspective and develops their computational thinking.

Figure 3*Expected outcomes for primary education in computing (MŠMT, 2021)*

<p>Data, information and modelling</p> <p>The pupil</p> <ul style="list-style-type: none"> — gives examples of data that surrounds them that can help them make better decisions; pronounces answers based on data — describes a specific situation, determines what they already know about it, and illustrates it — reads information from the given model
<p>Algorithmizing and programming</p> <p>The pupil</p> <ul style="list-style-type: none"> — compiles and tests symbolic notations of procedures — describes a simple problem, suggests and describes the individual steps of its solution — builds a program in a block-oriented programming language, recognizes repeating patterns, uses repetition and prepared subroutines — verifies the correctness of the procedure or program proposed by them, finds and corrects any errors in it
<p>Information systems</p> <p>The pupil</p> <ul style="list-style-type: none"> — recognizes the individual elements and the relationships between them in the systems that surround them — records numerical and non-numerical data in an existing table or list for a defined problem
<p>Digital technology</p> <p>The pupil</p> <ul style="list-style-type: none"> — finds and runs the application, works with data of various types — connects digital devices, lists the possible risks associated with such a connection — observes safety and other rules for working with digital technologies

With regard to the organisation of teaching, it was emphasised that pupils should actively construct their knowledge by discovering, discussing and solving problems, for example. Group activities are recommended. The emphasis is not placed on the reproduction of knowledge or on memorising information (MŠMT, 2021).

The other area of innovation, digital literacy, has become a so-called key competence, which means it must be implemented in all educational areas (NPI ČR, 2025). All school subjects are responsible for achieving outcomes related to this competence. Thus, the subject of computing is not specifically responsible for achieving goals in digital literacy such as, but not limited to, using spreadsheets for calculations in pupils' research in science classes, type-writing in Czech language classes, safety and interpersonal relationships on the Internet in civics, etc. On the other hand, it is obligatory for all subjects to use the computer as a teaching aid to fulfil their subject educational goals. Digital technologies can be expected to change teachers' working methods.

The revised curriculum emphasises the development of computational

thinking and digital literacy, aligning closely with international frameworks such as the K-12 Computer Science Framework Steering Committee (K12CS) and Informatics for All (I4ALL). While K12CS provides a comprehensive structure for integrating computing across school levels (K12CS, 2016) and I4ALL focuses on establishing computing as a foundational discipline in education (I4ALL, 2025), the Czech curriculum reflects a similar ambition by embedding computing as a core subject and fostering digital literacy as a cross-curricular skill.

Introducing innovation at schools

From September 2021, schools commenced the transition to the new computing curriculum by mixing traditional and new educational content. This transition was optional and schools could opt out of it. Since September 2024, all grades at primary and lower-secondary level must be taught according to the new national curriculum.

The publication of the new national curriculum triggered a massive discussion in the professional community. The discourse addressed the anticipated queries surrounding the curriculum and the financial implications of acquiring computers and educational robots. The pivotal question for primary education pertained to the identification of suitable candidates for the instruction of this novel discipline. The absence of new teachers with expertise in computing, coupled with the unlikelihood of higher-grade teachers providing assistance due to their own teaching loads, highlighted a critical shortage of qualified educators.

The most viable solution identified was to focus on primary teachers who teach regular subjects in grades 4 and 5 (pupils aged 9–10 years) also teaching computing in their classes. The solution in which computing would be taught at primary level by secondary teachers who are specialists in computing was not supported. Teachers who were encouraged or chose to teach computing received short voluntary training programmes and were provided with textbooks and methodological guides. These educational materials were prepared and tested for use by teachers without previous training who had never taught programming. The process was aided by the publication of exemplary school educational programmes for computing, allowing easier planning of teaching the new subject (PRIM, 2021).

Research problem and research questions

The issue under consideration in the present study is whether the decision to rely on existing teachers to reform computer science in primary

education was the correct one. The question therefore arises as to how these teachers, who have not undergone any pre-service training, perceive teaching programming and what knowledge they have in this area.

The topic of programming was chosen because it is one of four main areas (see Figure 3) of the new school subject. Moreover, programming was historically the least popular computer science topic among teachers (even when teaching it was not mandatory) (Rambousek et al., 2015).

Processes associated with programming require skills such as problem decomposition, algorithmisation, abstraction and automation, thereby helping to develop computational thinking (Yadav et al., 2017). This is consistent with our findings in the Czech national Bebras task archive (JU, 2024), where each task contains tags about its connection to some computer science topic and to the computational thinking components it develops. Tasks categorised as programming and algorithmic are consistently tagged with all of the computational thinking components.

We do not assume that primary teachers have programming skills, so these were not tested in the research, which instead focused on the components of computational thinking. Thus, the main aim of the present paper is to describe the overall level of primary teachers' attitudes towards teaching programming and their computational thinking. Another aim is to identify the personal and pedagogical factors that influence primary teachers' attitudes towards programming and algorithmic thinking. We are interested in whether there are groups of teachers who have significantly more positive or negative attitudes towards teaching programming. In addition, we would like to determine the relationship and interaction of the selected factors with teachers' attitudes towards teaching programming.

With these goals in mind, the following research questions were established:

- RQ1: What are teachers' overall attitudes towards teaching programming?
- RQ2: What is the level of teachers' computational thinking?
- RQ3: What factors influence primary teachers' attitudes towards teaching programming and their level of computational thinking?

Method

Participants

The research sample consisted of 239 primary education teachers from the Czech Republic who teach all of the regular subjects in grades 4 and 5, including computing. The research did not include specialised secondary school computing teachers who also teach computing at the primary level. The intention was for the research to reflect the specificities of computer science teaching by primary teachers who have not received any pre-service training in teaching computer science and thus are not trained in teaching this subject. The distribution of research participants per factor can be seen in Table 1.

The participants took part in the research voluntarily and anonymously. Each school participating in the Bebras Challenge contest (JU, 2025) has a designated contact person, the so-called school coordinator. The participants were approached through these school coordinators, who forwarded the request to their colleagues in the target group at school. A total of 1,450 schools from the contest database were approached, representing approximately one third of all target schools.

Table 1

Distribution of research participants per factor

Factor	<i>n</i>	%
Gender		
Female	54	22.6
Male	185	77.4
Age		
25–30	22	9.3
31–40	99	41.4
41–50	73	30.5
51–60	35	14.6
61+	10	4.2
Length of teaching		
<1	11	4.6
1–3	57	23.8
4–6	26	10.9
7–9	23	9.6
10+	122	51.1
Experience with teaching programming		
Yes	189	79.1
No	50	20.9

Research instruments

The main research instrument employed was an online questionnaire, which incorporated the Teacher Programming Attitude Scale (TPAS) and the Bebras Challenge tasks test (JU, 2025).

Created by Lihui Sun and Danhua Zhou (2023), the TPAS analyses teachers' attitudes towards programming and teaching programming using a 5-point Likert scale. The TPAS is divided into five dimensions: Programming Interests, Programming Self-Efficacy, Programming Utility, Cognition and Belief, and Reflection and Monitoring. Each item is scored from 0 (Strongly Disagree) to 4 (Strongly Agree). It is hypothesised that a higher score is indicative of a more positive attitude towards programming and teaching programming. In our work, we translated the scale into Czech and used the first three dimensions relevant to our environment and educational situation: Programming Interests, Programming Self-efficacy, and Programming Utility (17 items, 68 points in total). Although the other two dimensions – Cognition and Belief, and Reflection and Monitoring – capture important aspects of teachers' broader pedagogical approaches, they were deemed less applicable to our context, where programming is still being established as a new topic in primary schools. An example of the scale questions from the Programming Interests dimension in the interactive questionnaire environment can be seen in Figure 4.

Figure 4

A sample of the scale questions used from the TPAS, with an example of filling out the questionnaire (selected answer highlighted in green)

1. I like programming, or I think I will like programming.

Strongly Disagree	Disagree	I don't know	Agree	Strongly agree
-------------------	----------	--------------	-------	----------------

2. I think programming is very interesting.

Strongly Disagree	Disagree	I don't know	Agree	Strongly agree
-------------------	----------	--------------	-------	----------------

In order to evaluate the level of programming competencies, we designed a test consisting of so-called Bebras tasks from the Czech edition of the Bebras Challenge. These tasks are a popular way to measure computational thinking (Zapata-Cáceres, 2024). Bebras tasks require no prior knowledge or experience in programming. Each task involves at least one computer science concept (Dagienė & Futschek, 2008).

A total of eight Bebras tasks were included in our test. The individual tasks were selected from the five years of the contest (2019 to 2023), and their

use in the contest verified their validity and reliability. The tasks were selected from the age category designed for pupils in grades 3, 4 and 5, so the difficulty of the selected tasks matched that of the tasks teachers are expected to use in their classes. The overall success rate of these tasks among pupils in the challenge was less than 50%, although this increased to 70% in two tasks where pupils were permitted to adjust their solutions according to immediate feedback.

We limited ourselves to Bebras tasks involving algorithmic thinking and programming. Our tasks covered the following computational thinking components: abstraction, algorithmisation, evaluation, decomposition and generalisation.








An example of the Bebras tasks used can be seen in Figure 5. This task presents an algorithm for making a hamburger and the goal is to evaluate the correctness of the given hamburgers.

Figure 5

An example of a Bebras task used in the test (Bebras code 2022KR03)

Beaver Jessica is making hamburgers according to the rules below.

Hamburger ingredients:

Buns	Meat	Sauce	Pickles	Lettuce	Onions	Cheese
						

1. The sauce should be right above the meat.
2. Meat and cheese should be below the pickles, lettuce and onions.
3. Onions should not be in contact with the buns.
4. All ingredients must be between the buns.

Which hamburger is correctly made according to the rules?



The complete set of tasks (in a form of web link) can be found online at <https://www.ibobr.cz/papers/ceps25.pdf>.

Research design

In order to address our research questions, we employed a quantitative research design using a survey-based approach. The research follows a correlational design, as it aims to examine relationships between the participating teachers' attitudes towards teaching programming, their level of computational thinking, and various influencing factors. Data collection was conducted through an online questionnaire that combined self-reported measures of the teachers' attitudes and an objective assessment of their computational thinking skills. After anonymously signing up for the online questionnaire, the participants filled out an attitude questionnaire and a Bebras Challenge computing test. They also provided information about their gender, age, length of teaching and prior experience with teaching programming. No other personal data was collected.

Data analysis was conducted separately for each factor (gender, age, length of teaching, prior experience with programming). Using the statistical software R, we compared the respondents' scale scores and looked for differences based on the given factor. All of the statistical tests below were performed at a significance level of $\alpha = 0.05$, which was chosen considering the size of the research sample.

In order to answer RQ₁, the scale and test results were analysed using tools of descriptive statistics. The mean and median of the overall scale results were computed, followed by an examination of the results per dimension (Programming Interest, Programming Utility, Programming Self-Efficacy). The test outcomes were then considered separately. For each task, the success rate was calculated and subsequently compared across tasks.

In order to answer RQ₂ and RQ₃, we first planned to use the statistical test of analysis of variance. We used the Shapiro-Wilk test to test the normality of the data and found that our data does not follow a normal distribution, which is one of prerequisites for using the test of analysis of variance. We therefore tested for differences using the Mann-Whitney-Wilcoxon test to test the influence of gender and prior experience with teaching programming, and the Kruskal-Wallis test to test the influence of the age and length of teaching.

Results

The results are divided into sections based on each research question.

Overall teachers' attitudes towards teaching programming

Table 2

Descriptive statistics of teachers' attitudes by TPAS dimension

Dimension	Questions	Highest achievable score	Median score	%	Min	Max
Programming interests	7	28	19	68	6	28
Programming self-efficacy	4	16	9	56	0	16
Programming utility	6	24	17	71	0	24
Overall programming attitudes	17	68	45	66	16	68

The overall results of the TPAS were analysed as well as the results by dimension. Table 2 shows the number of questions used, the total number of achievable points of the teachers' attitudes towards teaching programming, the median of the points achieved by the participants, the percentage of the median to the total number of points, and the minimum and maximum achieved points by the participants.

Examining the results by dimension (Table 2), it is clear that each dimension's score is at a middle upper level, which corresponds to the findings of the scale authors (Sun & Zhou, 2023). The programming self-efficacy score, which contained four questions, is the lowest of the three dimensions (Table 3). A detailed examination of the scores reveals that the surveyed teachers are not confident about their programming skills but are optimistic about their ability to learn programming and to deal with the various difficulties encountered in the process of learning programming.

Table 3

Results of the Programming self-efficacy dimension (in % of participants)

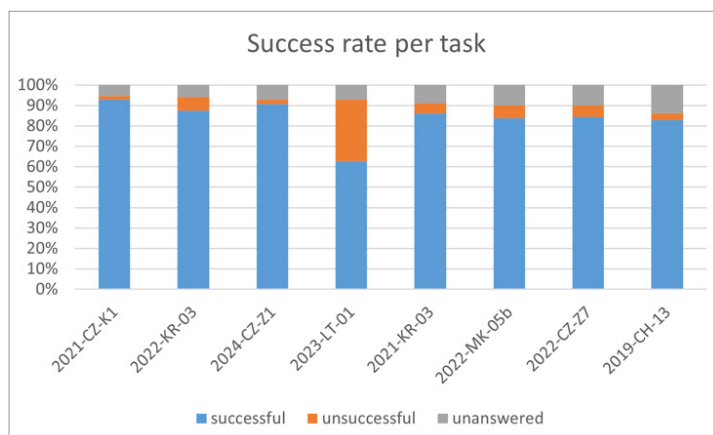
Scale question	Strongly Disagree	Disagree	I don't know	Agree	Strongly Agree
I am good at programming.	4	33	48	13	2
I believe that I am capable of dealing with the difficulties encountered in learning programming.	3	19	28	47	3
I believe I can learn to program if I work hard enough.	1	3	15	66	15
In the eyes of others, I am good at programming.	3	8	66	23	0

Overall primary teachers' computational thinking level

We then focused on the results of the Bebras tasks. The overall success rate per task can be seen in Figure 6. At least 80% of the teachers successfully answered each task, except for task 2023-LT-01, which proved to be difficult for more than one third of the respondents. According to Vaníček and Šimandl (2020), the subjective difficulty of a task is more aptly expressed in terms of the proportion of participants who did not answer the task than the proportion of participants who answered incorrectly. As shown in Figure 6, each task was answered by the vast majority of the teachers. We interpret these findings as demonstrating that the results are positive, and that the teachers surveyed have a good level of computational thinking.

Figure 6

Success rate of each of the eight used test tasks



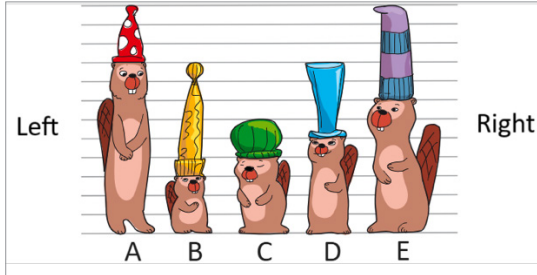
Let us look at what caused the significantly higher error rate in task 2023-LT-01, which was worded as follows:

Sort the beavers (in Figure 7) in ascending order according to their hats' heights. The beavers to the right must have taller hats than the beavers to the left. In which sequence will the beavers stand?

Check one of the answers: 1) C A D E B 2) B E D A C 3) C A E D B 4) E B D A C

Figure 7

Part of the settings of Task 4 “Sort the beavers by hats”



An analysis of the answers to this task reveals that the participants perceived the task as easy (only 7% did not answer). Nonetheless, approximately one quarter of them selected solution 2, in which the order of letters was flipped horizontally compared to the correct solution 1. This indicates a weak accommodation or misunderstanding of the rule “the beavers to the right have taller hats than the beavers to the left”. These participants sorted hats in the descending order from the left, as they were probably used to doing from previous experience.

Factors influencing teachers' attitudes towards teaching programming

We then focused on the factors that influence teachers' attitudes towards teaching programming. First, the normality of our data set was tested using the Shapiro-Wilk test. The results (p -value = 0.00011) indicate that our data comes from non-normal distribution. We then used the Mann-Whitney-Wilcoxon test to test the influence of gender and prior experience with teaching programming, and the Kruskal-Wallis test to test the influence of the age and length of teaching. The results can be seen in Table 4.

Table 4

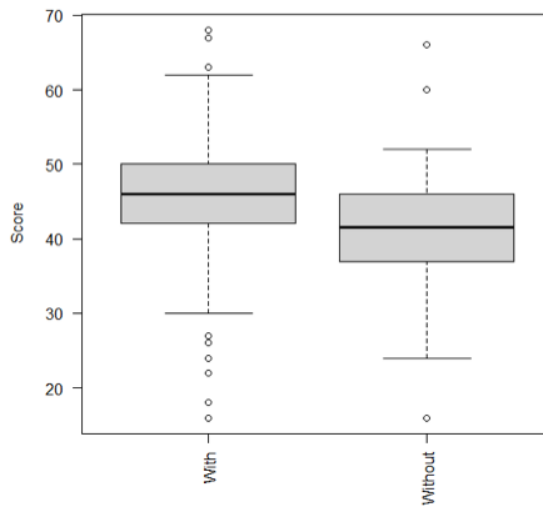
Results of the Mann-Whitney-Wilcoxon test and the Kruskal-Wallis test for teachers' attitudes towards teaching programming

Factor	p -value
Gender	0.5748
Age	0.3511
Length of teaching	0.5710
Prior experience with teaching programming	0.0001

No statistical significance was found in the influence of gender, age and length of teaching. However, the influence of prior experience with teaching programming was found to be statistically significant at the level 0.01, which means that it is hard to justify the hypothesis that prior experience of teaching programming has no influence. We interpret this result as demonstrating that teachers who have already taught programming in their classes have a more positive attitude towards teaching the topic. This finding is illustrated in the scale results in Figure 8.

Figure 8

Boxplots of TPAS scores of teachers with and without prior experience with teaching programming



We then analysed the differences between the teachers with and without prior experience with programming per scale dimension of the TPAS (Table 5). The results show statistically significant differences in both the Programming self-efficacy and the Programming utility dimensions. This can be interpreted as demonstrating that teachers with prior experience have more positive attitudes in programming self-efficacy and utility (Figure 9). However, there are no statistically significant differences between the two groups in the Programming interest dimension. We interpret this result as demonstrating that all of the surveyed teachers have similar attitudes in programming interests. It has therefore not been proven that more experienced teachers are more interested in programming.

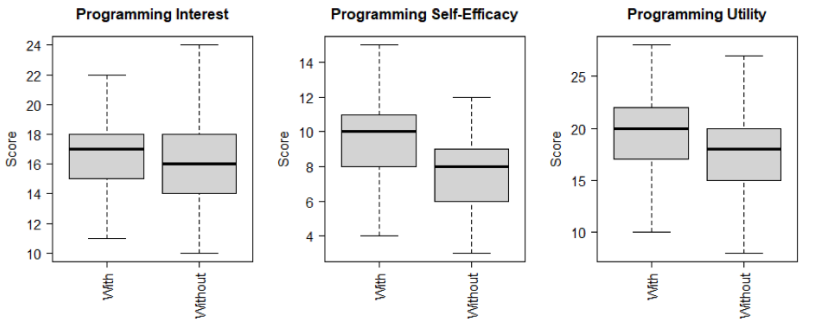
Table 5

The results of the Mann-Whitney-Wilcoxon test for teachers' attitudes in TPAS dimensions

Dimension	<i>p</i> -value
Programming interest	0.1012
Programming self-efficacy	0.0038
Programming utility	0.0000

Figure 9

Boxplots of the TPAS dimension scores of teachers with and without prior experience with teaching programming within each dimension



A similar strategy was employed to determine the factors that influence the teachers' level of computational thinking. The normality of our data set was first tested using the Shapiro-Wilk test. The results (p -value = 0.0000) indicate that our data comes from non-normal distribution. We then used Mann-Whitney-Wilcoxon test to test the influence of gender and prior experience with teaching programming, and the Kruskal-Wallis test to test the influence of the age and length of teaching. The results (Table 6) show no statistically significant influence of any of the factors on the teachers' level of computational thinking. All of the types of teachers achieved similar results in the Bebras tasks test.

Table 6

Results of the Mann-Whitney-Wilcoxon test and the Kruskal-Wallis test for teachers' level of computational thinking

Factor	p-value
Gender	0.8240
Age	0.7540
Length of teaching	0.7191
Prior experience with teaching programming	0.4200

Discussion and conclusion

The present study revealed several key insights into the attitudes of primary school teachers towards teaching programming within the context of the Czech educational reform. The findings indicate that teachers generally have a positive attitude towards programming and recognise its value in education. Programming was historically the most unpopular computer education topic in Czech schools both before (Rambousek et al., 2015) and during (Bryndová, 2021) the educational reform, suggesting that the reform has had a positive effect on the implementation of these changes, especially with regard to its realisation in the form of practical training and workshops that enabled teachers to experience programming first hand. The same effect can be seen in other countries. For example, German primary teachers who participated in training on teaching programming and algorithmisation had a significantly more positive approach to teaching these topics than before; they also had greater trust in their ability to teach these topics to their pupils and saw more meaning and benefit in such teaching (Geldreich et al., 2018). Similarly, pre-service primary teachers in Portugal had a positive attitude towards teaching programming after completing a robotics course (Piedade et al., 2020). In-service training and a feeling of participation were also key factors mentioned by Finnish teachers when asked about their attitudes towards teaching programming (Korhonen et al., 2023).

The present study also revealed that factors such as gender, age and general length of teaching did not show statistically significant effects on teachers' attitudes. The only factor that had a statistically significant effect on teachers' attitudes was their prior experience with teaching programming. This finding is not entirely consistent with previous studies, some of which found that male teachers have more positive attitudes towards programming (Sun & Lhou, 2023). In the context of primary education, however, it was found that female

teachers can have more positive attitudes towards programming, especially when using visual programming languages (Witherspoon et al., 2018). Similarly, some studies indicate that age and length of teaching can be an influencing factor, with younger teachers with less than five years of teaching having a positive attitude towards teaching programming (Sun & Lhou, 2023).

The weakest aspect of the surveyed teachers' attitudes is self-assessing their programming skills, which suggests a need for confidence-building measures. Teachers with prior experience of teaching programming exhibit significantly higher self-efficacy. Similarly, Rich et al. (2021) found that teachers who participated in a continuous professional development course showed increased self-efficacy for both programming and teaching programming.

Our analysis of primary teachers' computational thinking found that their level is high enough to effectively teach programming in primary schools. It should be noted, however, that the tasks used in our testing were designed for pupils from grades 3, 4 and 5 and had a low success rate among pupils. We also focused on factors influencing the level of teachers' computational thinking and found no statistically significant effect of any of the examined factors. This is consistent with other studies, such as that by Günbatar and Bakırcı (2019), who examined the relationship between gender (among other factors) and the level of pre-service primary school teachers' computational thinking and found no differences between male and female teachers. Moreover, Bati and Yetişir (2021) found that the general length of teaching has no significant impact on teachers' computational thinking skills.

Practical implications and recommendations

Considering that our research shows that the only important factor influencing the respondents' attitudes towards teaching programming is their previous experience with teaching programming, it follows that teachers need to be "thrown into the water": let them teach. Through their own teaching, they gain self-confidence and better self-esteem. The findings of the present study demonstrate that primary teachers can be competent teachers of programming for their pupils, as confirmed by the test results. A good level of computational thinking is a necessary foundation for building primary programming teaching on the basis of primary teachers, rather than on secondary computing teachers specialists.

This leads to one recommendation for the state policy of introducing the subject of computing: there is no need to wait until teachers are trained, can program, and somehow acquire a positive attitude towards programming. This cautious approach prevents teachers from gaining experience in teaching

programming and consequently having a better attitude to teaching, which is key to initiating these changes.

In implementing the paradigm shift in the field of information and communication technology education towards computer education in Czechia, we have gained experience at a practical level, which we can formulate in the following paragraphs. In order to facilitate the transition from novice to competent programming teacher, it is imperative that several conditions are met:

Using age-appropriate visual block-oriented programming software for pupils, which enables rapid initial progress, as well as a wide range of varied activities including advanced ones. In other words, activities with low floors, high ceilings (Papert, 1980) and wide walls (Resnick, 2017), on which pupils will be able to develop basic programming approaches without being burdened with formal knowledge.

Creating high-quality textbooks that offer engaging programming activities for pupils, that are easy for teachers to understand, and that are accompanied by clear methodologies and detailed descriptions of teaching procedures. In schools in remote areas, where introductory face-to-face training is difficult to carry out, these materials must be sufficient for teachers to prepare themselves for teaching.

Running an introductory training system that will help teachers to familiarise themselves with the subject matter. Teachers need to gain self-confidence, and this will not be achieved through theoretical lectures. Instead, this objective needs to be realised through intensive short training in which teachers complete programming lessons in the role of pupils, using tasks from a primary school textbook.

It is necessary to provide space and comfort for teachers for innovative work, i.e., a kind of time cushion of several years, during which the curriculum will be expanded to encompass additional programming topics and methodologies. During this period, teachers should not be stressed by inspections. Furthermore, the implementation of tandem teaching or the provision of mentoring assistance from the methodological centre directly in the classroom can help reduce teachers' feelings of uncertainty.

It is imperative that teachers are adequately motivated. Teacher motivation plays a central role in the successful integration of programming and computing into school curricula. Research indicates that motivation significantly shapes the quality of implementation of programming education and the extent to which teachers feel confident in engaging with the topic (Wangenheim et al., 2017).

In order for programming to be successfully integrated into teaching, it is essential that teachers experience teaching programming lessons in practice;

by seeing their pupils' engagement and their own sense of success and meaningfulness, teachers become more motivated and genuinely feel like they are part of the change. The present research did not demonstrate the influence of any of the factors examined, including previous experience with programming teaching, on increasing the level of teachers' computational thinking. It can therefore be deduced that computational thinking at the level that was examined is already good before the first actual teaching. It is questionable whether teaching experience increases the teacher's didactic competences in this area, although our research did not specifically investigate this.

Limitations of the study

Only a specific group of teachers was included in the research, i.e., teachers from schools that participate in the Bebras contest and are therefore more involved in computing innovation. This fact must be considered in interpreting the results, even though more than a third of all schools were approached. Although the method of approaching teachers through the local organisers of the contest did not affect the final selection of the teachers, as the selection was not limited to teachers involved in the competition, the selection process is nonetheless a limiting filter to some extent.

Due to the selection of questions for the test to assess the level of the teachers' computational thinking, the test became less sensitive, as it turned out to contain questions that were too easy. However, the main interest of the research was not to assess the general level of teachers' computational thinking, but rather to assess whether teachers were able to understand the tasks given to their students. Using more difficult tasks would probably give a more accurate picture of the impact of individual factors, but with the risk that many teachers would not complete the test, resulting in the segment of the weakest and least confident respondents being missing from the research sample.

Possible future research

When sufficient time has elapsed since the introduction of the reform and teaching outcomes can be collected, further research could examine whether core primary teachers have good teaching outcomes and therefore sufficient pedagogical content knowledge.

It would be possible to compare the results of the present research with those of lower secondary computing teachers who also teach computing at primary level and therefore have previous computing training.

In order to further analyse and examine core primary teachers' attitudes towards teaching programming, in-depth interviews could be conducted with them.

Ethical statement

The research study was approved by the Ethical Research Committee of the Faculty of Education, University of South Bohemia, Czechia.

Disclosure statement

The authors have no conflict of interest to declare.

References

- Bati, K., & İkbal Yetişir, M. (2021). Examination of Turkish middle school STEM teachers' knowledge about computational thinking and views regarding information and communications technology. *Computers in the Schools*, 38(1), 57–73.
- Blaho, A. (2012). Informatika v štátnom vzdelávacom programe [Informatics in a state educational programme]. In I. Kalaš (Ed.), *DidInfo 2012* (pp. 7–14). Katedra informatiky, FPV, Univerzita Mateja Bela MPC, Regionálne pracovisko v Banskej Bystrici UIPŠ, Školské výpočtové stredisko Banská Bystrica. http://www.didinfo.net/images/DidInfo/files/didinfo_2012.pdf
- Blaho, A., & Salanci, L. (2011). Informatics in primary school: Principles and experience. In I. Kalaš, & R. T. Mittermeid (Eds.), *Informatics in Schools. Contributing to 21st Century Education* (pp. 129–142). Springer.
- Bosco, A., Santiveri, N., & Tesconi, S. (2019). Digital making in educational projects. *Center for Educational Policy Studies Journal*, 9(3), 51–73. <https://doi.org/10.26529/cepsj.629>
- Bryndová, L. (2021). The approach of computer science teachers to the concepts of computational thinking and the implementation of its development in primary schools. *Journal of Technology and Information Education*, 13(2), 151–163.
- Dagienė, V., & Futschek, G. (2008). Bebras International Contest on Informatics and Computer Literacy: Criteria for good tasks. In R. T. Mittermeir, & M. M. Syslo (Eds.), *Informatics education – Supporting computational thinking* (pp. 19–30). Springer. https://doi.org/10.1007/978-3-540-69924-8_2
- Gander, W. (2014). Informatics and general education. In Y. Gülbahar, & K. Erinc (Eds.), *Informatics in schools, teaching and learning perspectives* (pp. 1–7). Springer.
- Geldreich, K., Talbot, M., & Hubwieser, P. (2018). Off to new shores: Preparing primary school teachers for teaching algorithmics and programming. In A. Mühling, & Q. Cutts (Eds.), *WiPSCE 2018: Proceedings of the 13th Workshop in Primary and Secondary Computing Education* (pp. 1–6). <https://doi.org/10.1145/3265757.3265783>

- Günbatar, M.S., & Bakirci, H. (2019). STEM teaching intention and computational thinking skills of pre-service teachers. *Educ Inf Technol*, 24, 1615–1629. <https://doi.org/10.1007/s10639-018-9849-5>
- I4ALL. (2025). *Informatics for All Coalition*. <https://www.informaticsforall.org/>
- JU. (2017). *Jihočeská univerzita. PRIM project homepage*. Jihočeská univerzita v Českých Budějovicích. <https://imysleni.cz>
- JU. (2024). *Bobřík informatiky, archiv úloh [Website of the Bebras Challenge contest, archive of Bebras tasks]*. Jihočeská univerzita v Českých Budějovicích. <https://www.ibobr.cz/archiv>
- JU. (2025). *Bobřík informatiky [Website of the Bebras Challenge contest]*. Jihočeská univerzita v Českých Budějovicích. <https://www.ibobr.cz/>
- K12CS. (2016). *K-12 Computer Science Framework*. Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative. <http://www.k12cs.org>
- Kodelja, Z. (2021). Consequentialist reasons for some education reforms. *Center for Educational Policy Studies Journal*, 11(2), 111–124. <https://doi.org/10.26529/cepsj.1061>
- Korhonen, T., Salo, L., Laakso, N., Seitamaa, A., Sormunen, K., Kukkonen, M., & Forsström, H. (2023). Finnish teachers as adopters of educational innovation: Perceptions of programming as a new part of the curriculum. *Computer Science Education*, 33(1), 94–116. <https://doi.org/10.1080/08993408.2022.2095595>
- MŠMT. (2005). *The Education Act*. Ministerstvo školství, mládeže a tělovýchovy České republiky. <https://www.msmt.cz/dokumenty-3/act-no-561-2004-collection-of-law-on-pre-school-basic>
- MŠMT. (2014). *Strategie digitálního vzdělávání [Strategy of digital education]*. Ministerstvo školství, mládeže a tělovýchovy České republiky. <https://www.msmt.cz/uploads/DigiStrategie.pdf>
- MŠMT. (2021). *Rámcový vzdělávací program pro základní vzdělávání [Frame educational programme for basic education]*. Ministerstvo školství, mládeže a tělovýchovy České republiky. <https://www.edu.cz/wp-content/uploads/2021/07/RVP-ZV-2021.pdf>
- NPI ČR. (2025). *Klíčová kompetence digitální [Key digital competence]*. National Pedagogical Institute of the Czech Republic. <https://prohlednout.rvp.cz/zakladni-vzdelavani/klicove-kompetence/kdi>
- NÚV. (2005). *Rámcový vzdělávací program pro základní vzdělávání – základní verze [Frame educational programme for basic education – basic version]*. National Institute for Education of the Czech Republic. <http://www.nuv.cz/file/493/>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Pavlas, T., Zatloukal, T., & Andrys, O. (2021). *Návrat žáků k prezenčnímu vzdělávání v základních a středních školách [Return of students to full-time education in primary and secondary schools]*. Czech School Inspectorate. <https://www.csicr.cz/cz/Dokumenty/Tematicke-zpravy/Tematicka-zprava-%E2%80%93Navrat-zaku-k-prezencnimu-vzdel>
- Perla, L., Vinci, V., & Agrati, L. S. (2025). The Italian way to the Europeanisation of teacher education: An analysis of reforms and the ongoing experience of digital transformation. *Center for Educational Policy Studies Journal*, 15(1), 147–174. <https://doi.org/10.26529/cepsj.1714>

- Piedade J., Dorotea N., Pedro, A. & Matos, J. F. (2020). On teaching programming fundamentals and computational thinking with educational robotics: A didactic experience with pre-service teachers. *Education Sciences*, 10(9), Article 214. <https://doi.org/10.3390/educsci10090214>
- PRIM. (2021). *Modelové školní vzdělávací programy* [Exemplary school educational programmes]. Jihočeská univerzita v Českých Budějovicích. <https://imysleni.cz/svp>
- Rambousek, V., Štípek, J., & Wildová, R. (2015). ICT competencies and their development in primary and lower-secondary schools in the Czech Republic. *Procedia - Social and Behavioral Sciences*, 171, 535–542.
- Resnick, M. (2017). *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT Press.
- Rich, P. J., Mason S. L., & O'Leary J. (2021). Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers & Education*, 168, Article 104196. <https://doi.org/10.1016/j.compedu.2021.104196>
- Sun, L., & Zhou, D. (2023). K-12 teachers' programming attitudes among different disciplines: Analysis of influential factors. *Journal of Computer Assisted Learning*, 40(2), 538–556.
- Syslo, M., & Kwiatkowska, A. (2015). Introducing a new computer science curriculum for all school levels in Poland. In A. Brodnik (Ed.), *Informatics in schools. Curricula, competencies, and competitions* (pp. 141–154). Springer.
- The Royal Society. (2012). *Shut down or restart? The way forward for computing in UK schools*. <https://royalsociety.org/~media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>
- Tucker, A. (Ed.). (2003). *A model curriculum for K-12 computer science: Final report of the ACM K-12 task force curriculum committee*. Association for Computing Machinery. <https://people.cs.vt.edu/~kafura/CS6604/Papers/K-12ModelCurr2ndEd.pdf>
- UMB. (2024). *DidInfo Homepage*. Fakulta přírodních věd, Univerzita Mateja Bela. <http://www.didinfo.net/en>
- UNESCO. (2002). *Information and communication technology in education – A curriculum for schools and programme for teacher development*. UNESCO.
- Vaniček, J. (2021). Towards a compulsory computing curriculum at primary and lower-secondary schools: the case of Czechia. In E. Barendsen, & C. Chytas (Eds.), *Informatics in schools. Rethinking computing education ISSEP 2021* (pp. 109–120). https://doi.org/10.1007/978-3-030-90228-5_9
- Vaniček, J., & Šimandl, V. (2020). Participants' perception of tasks in an informatics contest. In K. Kori, & M. Laanpere (Eds.), *Informatics in Schools. Engaging Learners in Computational Thinking* (pp. 55–65). Springer. https://doi.org/10.1007/978-3-030-63212-0_5
- Wangenheim, A. von, Wangenheim, C. G. von, Pacheco, F. S., Hauck, J. C. R., & Ferreira, M. N. F. (2017). Motivating teachers to teach computing in middle school: A case study of a physical computing taster workshop for K-12 teachers. *International Journal of Computer Science Education in Schools*, 1(4), 35–49. <https://doi.org/10.21585/ijcses.v1i4.17>
- Witherspoon, E. B., Schunn, C. D., Higashi, R. M., & Shoop, R. (2018). Attending to structural programming features predicts differences in learning and motivation. *Journal of Computer Assisted*

Learning, 34(2), 115–128.

Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational thinking as an emerging competence domain. In M. Mulder (Ed.), *Competence-based vocational and professional education: Bridging the worlds of work and education*, 23 (pp. 1051–1067). Springer International Publishing.

https://doi.org/10.1007/978-3-319-41713-4_49

Zapata-Cáceres, M., Marcelino, P., El-Hamamsy, L., & Martín-Barosso, E. (2024). A Bebras Computational Thinking (ABC-Thinking) program for primary school: Evaluation using the competent computational thinking test. *Education and Information Technologies*, 29(12), 14969–14998.

Biographical note

JIŘÍ VANÍČEK, PhD, is an associate professor and head of the Department of Computing at the Faculty of Education, University of South Bohemia in České Budějovice, Czech Republic. His research interests are informatics education in primary and lower-secondary schools and programming at an early age. He has published over 70 scientific papers and over 100 methodological works. He has written eight computing textbooks for primary and lower-secondary schools. Since 2014 to 2020, he was a member of an expert group at the National Institute for Education which created innovated state educational framework for computing. In 2017 – 2020 he was the head of the strategic ministerial project PRIM – Support for the Development of Computational Thinking, which developed a new informatics curriculum for all levels of schools and a system for training teachers in this field. In 2025, he participated in creation of the Competency Framework for Computing Teachers. For 18 years he has been organizing the Bebras informatics contest in the Czech Republic. He is the representative of the Czech Republic in the International Bebras Association.

JAN PRŠALA is an assistant in the field of computer science education at the Faculty of Education at the University of Hradec Králové, Czech Republic. His research interests include programming education and computational thinking in primary and lower-secondary schools, the use of tools such as Scratch and micro:bit for developing computational thinking, as well as teachers' attitudes toward teaching programming and computing and teacher professional development in computing. He is the author of a programming textbook for lower-secondary schools. For the past five years, he has served as the chief editor of the Bebras informatics contest in Czechia.