Journal of Educational Technology & Online Learning
Volume 8 | Issue 1 | 2025

JETOL Journal of Educational Technology & Online Learning

Unpacking teachers' value beliefs about computational thinking and programming

Filiz Kuşkaya Mumcu [a]*, Branko Andic [b], Mirjana Maricic [c], Mathias Tejera [b], Zsolt Lavicza [b]

[a] Manisa Celal Bayar University, Faculty of Education, Türkiye.
[b] Johannes Kepler University, Linz School of Education, Austria.
[c] University of Novi Sad, Faculty of Education, Serbia.

Suggested citation: Mumcu, F.K., Andic, B., Maricic, M., Tejera, M., & Lavicza, Z. (2025). Unpacking Teachers' Value Beliefs about Computational Thinking and Programming. *Journal of Educational Technology & Online Learning*, *8*(1), 41-63.

Highlights

- The study successfully adapted and validated the Turkish form of the TBaCCT Scale for measuring teachers' value beliefs about computational thinking and programming.
- Significant gender differences were found in programming self-efficacy and teaching programming efficacy, with male teachers scoring higher, but no differences were noted in value beliefs and computational thinking self-efficacy.
- Teachers' value beliefs about computational thinking and programming varied significantly across subjects, with computer science teachers scoring the highest and social sciences, native language, and foreign language teachers scoring the lowest.
- The study emphasizes the need for professional development programs for social sciences teachers to enhance their beliefs and knowledge about computational thinking.

Article Info: Research Article

Keywords: Computational thinking, teacher, teaching programming, self-efficacy, value beliefs

Abstract

Many education policy strategy documents at the European Union level, as well as national strategies of various countries, recommend including computational thinking as a fundamental skill in curricula. The professional development of teachers should be supported to disseminate computational thinking in K12 education. Teachers' value beliefs about computer science and programming should be first known when designing professional development programs. This study aims twofold. The first is to adapt the Teacher Beliefs about Coding and Computational Thinking (TBaCCT) Scale into Turkish. The second is to explore Turkish primary and secondary school teachers' value beliefs about computational thinking and programming. The study involved 417 teachers. Confirmatory factor analysis was used for the validity studies of the scale. Independent samples t-test, one-way ANOVA, and MANOVA analysis were used to examine whether the scores differed according to gender and subject, respectively. The findings show that the Turkish form of the TBaCCT Scale is valid and reliable. For programming self-efficacy and teaching programming efficacy, there is a significant difference between male and female teachers, computer science teachers and other subjects, and elementary mathematics, class and science teachers and other teachers. Teachers working in social sciences especially need professional development programs that will transform their beliefs and knowledge about computational thinking.

* Corresponding author. Computer Education and Instructional Technology Department, Faculty of Education, Manisa Celal Bayar University, Türkiye.
e-mail addresses: filizkuskaya@gmail.com

Doi: http://doi.org/10.31681/jetol.1497284
Received 10 Jun 2024; Accepted 2 Oct 2024

1. Introduction

Computational thinking (CT) research has been proliferating since 2013 and is fed by the knowledge produced in education, informatics, and social sciences (Tekdal, 2021). The International Society for Technology in Education (ISTE, 2016) recognizes CT as one of the skills that students should have. According to ISTE (2016), students should utilize the components of CT to solve problems, formulate, facilitate decision-making, develop models, and create automated solutions. However, it is widely recognized that CT is a fundamental skill that should be encouraged from childhood and is educationally relevant to 21st-century skills (Tran, 2019; Tsarava et al., 2018).

In reviewing the literature on CT, Lodi (2020) highlights that CT involves a specific way of approaching problems, focusing on the thought process behind problem-solving. The problem and its solution should be structured so that an external entity, whether human or machine, can process and execute it effectively. Wing (2006) suggests that CT should be included in curricula as a fundamental skill for students to grasp abstract, algorithmic, and logical thinking and solve challenging, open-ended problems. Guzdial (2008) emphasizes that CT can be made more accessible to students by integrating computer science into different disciplines. Underlying this suggestion of Guzdial (2008) is making computer science education accessible to everyone. Today, computer science education is still not a compulsory part of school education worldwide. As a promising development, educational reforms focusing on computer science education and CT have been evident worldwide since 2014, especially in developed countries (Bocconi et al., 2018; Guizdal, 2016; Seow et al., 2019). Computer science education is crucial apart from CT and should be part of compulsory education. In this respect, efforts to teach CT can be considered an essential tool for disseminating computer science education.

While how to integrate CT into the curriculum is an essential topic of discussion, the lack of readiness of teachers and teacher training programs is another. As the designer of the learning and teaching process (Mumcu et al., 2022a), the teacher needs to be ready for that integration. We should support the professional development of teachers from various disciplines to promote CT at each level of education (Kong et al., 2023). Many studies are focusing on the development of CT skills of pre and in-service teachers (e.g., Aminger et al., 2020; Chandra & Lloyd, 2020; Dağlı & Tokmak, 2021; Gabriele et al., 2019; Ketelhut et al., 2020; Umutlu, 2021; Yadav et al., 2019; Zha et al., 2020). The trend in these studies shifts from viewing pre and in-service teachers as CT learners to identifying them as equal collaborators who design CT-enhanced curricula within their subjects (Haşlaman et al., 2024; Hershkovitz et al., 2023). This perspective seems to contribute to teachers' belief that technology and CT can transform their classrooms (Hershkovitz et al., 2023). Similarly, Cabrera (2019) states that teachers' value judgments about computer science and programming should be first known when designing professional development programs to integrate CT into their teaching and learning processes. However, the question of how CT-related concepts can be measured remains current (Kalelioğlu et al., 2016; Lee & Malyn-Smith, 2020).

Perception scales toward CT and related topics such as computing, computer science, programming, and digital literacy are mainly used to assess CT (Román-González, Moreno-León, & Robles, 2019). The most used ones in the literature are scales developed by Korkmaz et al. (2017), Durak and Sarıtepeci (2018), Kukul, Gökçearslan, and Günbatar (2017), Gülbahar, Kert, and Kalelioğlu (2019), and Yağcı (2019). While various scales exist that assess teachers' self-efficacy regarding programming or beliefs about CT, there has been a distinct gap in the literature in evaluating teachers' value beliefs about teaching CT (Rich, Larsen, & Mason, 2021). Many of the widely used scales, such as those developed by Korkmaz et al. (2017) and Durak and Sarıtepeci (2018), focus primarily on programming self-efficacy or attitudes toward CT, but they do not provide a direct measure of teachers' value beliefs, which are crucial for understanding how teachers perceive the importance of these skills in education. Moreover, most of the measurement tools that claim to measure teachers' or pre-service teachers' self-efficacy, attitudes, or interest in CT measure programming rather than CT (Wang et al., 2022). This gap is addressed by the Teacher Beliefs about Coding and Computational Thinking (TBaCCT) scale developed by Rich Larsen and Mason (2021), which explicitly measures teachers' value beliefs regarding teaching CT. Unlike other tools that primarily evaluate self-efficacy or interest, the TBaCCT scale focuses on how teachers value CT within their teaching practices. This focus on value beliefs is critical because it directly influences how teachers prioritize and implement CT in their classrooms. Thus, the TBaCCT scale was chosen for this study to fill this identified gap and provide a more comprehensive understanding of teachers' beliefs.

This study aims twofold: first is to adapt the TBaCCT scale developed by Rich, Larsen, and Mason (2021) to Turkish to measure teachers' value beliefs about CT and programming, and second, to explore Turkish primary and secondary school teachers' value beliefs about CT and programming regarding gender and subject. A study by Günbatar and Bakırcı (2019) highlighted the critical role that subject matter plays in influencing teachers' attitudes toward integrating CT in their classrooms. Their findings suggest that teachers from science, technology, engineering, and mathematics (STEM) departments exhibit a stronger intention to incorporate CT into their teaching compared to those from non-STEM departments, indicating that the subject a teacher specializes in can significantly shape their perception of CT's relevance and applicability to their practice. Additionally, it is essential for all teachers, regardless of their subject, to integrate computer science concepts into their teaching to enable students to develop CT skills (Bocconi et al., 2022). Hence, analyzing the subject variable is crucial to developing effective professional development programs. Furthermore, a systematic review by Espino and González (2016) indicates that relatively few studies have explored CT from a gender perspective, despite its growing importance in education. Research by Fagerlund et al. (2022) found that male teachers were more intrinsically motivated toward programming and teaching CT than their female counterparts. Other studies have shown that the gender variable plays an important role in developing CT skills in teacher education, often favouring male teachers (Villalustre & Cueli, 2023; Yadav et al., 2014). Given these findings, it is essential to examine the role of gender in this study to better understand how it influences teachers' value beliefs about CT and programming, as addressing gender disparities is crucial for fostering equitable development of CT skills in education.

2. Literature

2.1. Teachers' Self-efficacy in CT and Programming

Teachers' self-efficacy is pivotal in determining teaching effectiveness and student outcomes. Tschannen-Moran and Woolfolk Hoy (2001) define teachers' self-efficacy as their belief in their capability to plan, organize, and execute activities necessary to achieve educational goals, even amidst challenges. Rooted in Bandura's (1977) broader theory, this concept emphasizes a teacher's belief in their capacity to achieve specific performance outcomes, extending to their ability to effectively manage the classroom, engage students, and facilitate learning. This is particularly relevant in complex areas like CT, where teachers must navigate constantly evolving technological landscapes.

Klassen and Tze (2014) explore how teachers' self-efficacy affects their motivational strategies and classroom management, directly influencing student engagement and learning outcomes. High self-efficacy is linked to dynamic, responsive teaching methods and more remarkable persistence in overcoming educational challenges. Bandura (2006) further notes that teachers with solid self-efficacy are inclined to experiment with new teaching strategies, adapting to the changing educational needs of their students. Such adaptability is crucial in CT, where educators must continuously refine their methods to align with technological advancements and pedagogical shifts.

The concept of self-efficacy is especially significant in the teaching of CT. Webb et al. (2017) demonstrate that teachers with high self-efficacy in this field can integrate CT into their curricula, making complex concepts more accessible to students. Their confidence is crucial in guiding students through logical problem-solving and systematic thinking. The impact of self-efficacy on teaching effectiveness is substantial. Teachers' confidence in their abilities creates more engaging and interactive classes, which is essential for fostering CT skills. Yadav et al. (2017) found that such teachers often employ innovative teaching methods tailored to diverse student learning styles. This adaptability elevates their teaching methods and significantly boosts student engagement and the practical application of CT skills, preparing students for various problem-solving scenarios in an increasingly digital world.

2.2. Teachers' Value Beliefs in CT and Programming

Examining teachers' perceptions (opinions, attitudes, and beliefs) about CT and programming values has intensified in the last few years. The research results so far provide different information about teachers' perceptions of the values of CT and programming. So, for example, Li et al. (2023) and Laime-Choque et al. (2022) indicate that CT can help students in developing creativity, critical thinking, and problem-solving skills. These results are also supported by the results of Avcı and Deniz (2022), which indicate that CT contributes to logical thinking, problem-solving, using algorithms, programming, doing mathematics, and high school students' technology usage. The research results by Prado et al. (2022) and Kafai and Proctor (2022) show that teachers believe that CT can help adapt tasks to different student abilities and achieve the principle of inclusiveness in teaching. After completing training in applying these approaches in education, Surahman et al. (2022) examined teachers' understandings and opinions about CT and STEM education. The research results indicate that teachers believe that CT can contribute to the realization of STEM principles in teaching. The results of Dimos et al. (2023) are interesting, as they indicate that teachers believe that CT can be used not only to achieve learning outcomes but also to evaluate students. In that research, descriptive assessment of students was used with the implementation of thematically related criteria (rubrics) and CT. Dimos et al. (2023) indicate that teachers can successfully evaluate students' achievements with the application of CT but they need professional development opportunities for it. Fessakis and Prantsoudi (2019) indicate that if teachers do not have adequate training and support for applying CT in teaching, they may develop misconceptions and negative opinion about this approach. Yılmaz et al. (2019) and Yadav, Hong, and Stephenson (2017) also indicate that if they do not have appropriate teacher training, they may have a superficial perception of computer thinking and completely identify it with problem-solving skills. All of the above research suggests that further research is needed to understand how teachers perceive the values that CT entails. This was the inspiration for our research us well.

3. Methodology

3.1. Research Design

This research follows a two-step approach. First, a scale adaptation study was conducted to adapt and validate the Teacher Beliefs about Coding and Computational Thinking (TBaCCT) scale developed by Rich Larsen and Mason (2021) for use with Turkish primary and secondary school teachers. The adaptation process involved confirmatory factor analysis (CFA) and reliability analyses to ensure the validity and reliability of the scale in the new context. Descriptive statistical methods were also used during this phase to evaluate the psychometric properties of the adapted scale.

Following the scale adaptation, a causal-comparative study was conducted to explore the value beliefs of Turkish primary and secondary school teachers regarding CT and programming. According to Fraenkel, Wallen, and Hyun (2012, p. 366), causal-comparative research *"aims to find reasons or outcomes for existing differences among groups, making it a form of associational research, similar to correlational research.*" This research observes existing differences, such as teaching styles or educational backgrounds, and investigates their potential causes or effects. As ex post facto research, it examines these differences retrospectively. In this study, the causal-comparative method was employed to analyze how different variables, such as gender and subject matter, might influence teachers' value beliefs about CT and programming. By exploring these associations, the research aims to identify potential factors that impact teachers' attitudes toward integrating these subjects into their teaching practices.

3.2. Participants and Context

Ethical approval was obtained from Manisa Celal Bayar University Scientific Research and Publication Ethics Committee to implement the study (Decision No. 20 taken at the meeting numbered 2022/10 dated 07.12.2022). 417 teachers working in primary and secondary schools in Turkiye participated in the study voluntarily. Demographic information about the teachers participating in the study is given in Table 1.

Table 1.

Demographic information of the study group.

Variable		**f**	**%**
Gender	Female	252	60.4
	Male	165	39.6
Teaching experience	<1	5	1.2
	1-5	27	6.5
	6-10	105	25.2
	11-15	102	24.5
	16-20	91	21.8
	21>	87	20.9
Subject	Computer science	155	37.2
	Mathematics	78	18.7
	Science	62	14.9
	Social sciences	24	5.8
	Visual arts	11	2.6
	Primary education	50	12.0
	Native language	19	4.6
	Foreign language	18	4.3
Total		417	100

Accordingly, the majority of the participants were female teachers. More than half of the teachers have between 6 and 15 years of professional experience, most of whom work at the secondary school level. In Turkiye, grades 5, 6 (middle schools), and grade 9 (science high schools) students can get computer science education. It is optional at other levels and institutions. The curriculum published in 2018 aims for students to "acquire and develop problem-solving and CT skills" (Ministry of National Education, 2018). CT is included in the "Problem Solving and Programming" unit of the "Information Technologies and Software" course curriculum. It covers the learning outcomes of problem-solving, algorithm design, programming components, block-based programs, and logic. In the grades where computer science education is compulsory, there is a noticeable effort to enhance students' CT skills (Yılmaz & İzmirli, 2023). However, integrating CT into school education has yet to be.

3.3. Data Collection Tool and Procedure

3.3.1. The TBaCCT Scale

The TBaCCT scale was developed by Rich, Larsen, and Mason (2021). The scale consists of 33 items in total with four sub-factors: "coding" (8 items), "teaching coding" (11 items), "value belief" (10 items), and "CT self-efficacy" (4 items). The scale was developed on a 6-point Likert scale (1: strongly disagree - 6: strongly agree). Rich, Larsen, and Mason (2021) state that the models created for each sub-factor show a good fit for the model.

3.3.2. The Adaptation Process of the TBaCCT Scale into Turkish

Regarding the aim of the study, the first step was to adapt the Teacher Beliefs about Coding and CT (TBaCCT) Scale into Turkish. Prior to the adaptation, we obtained permission from the original developers of the scale via email. The first author translated the scale items into Turkish, adhering to terminology commonly used in the Turkish literature. The draft Turkish version of the scale was then reviewed by five Turkish researchers from different universities, all of whom hold doctoral degrees in computer education and instructional technologies and have expertise in this field.

These experts were asked to evaluate the translation for each item by marking its appropriateness and providing suggestions for improvement where necessary. Based on their feedback, the Turkish version of the scale was finalized. Special attention was given to the experts' suggestions regarding the translation of terms like "computing" and "coding," as well as the translation of negatively worded items. "Computing" was translated as "computer science," and "coding" as "programming" where applicable. This distinction was made because, for computer science experts, "coding" refers to the physical act of writing a program (as it refers to the representation of information by a code), while "programming" encompasses a broader intellectual and cognitive process (Lodi & Martini, 2021, p. 886).

Given this, the sub-factor "coding" was renamed "programming self-efficacy" to better reflect the intellectual aspect of programming. Additionally, eight negatively worded items in the scale were changed to positive ones based on the experts' feedback. Further adjustments were made in response to the other issues raised by the experts, ensuring that the final version of the scale was linguistically and conceptually appropriate for Turkish teachers. In addition, the following were made regarding the other issues that the experts drew attention to:

- The experts found these two items unclear: "*I can find uses for computer programming that are relevant for students*." and "*I can* develop *and plan effective computing lessons*." under the "teaching programming efficacy" factor. Regarding their recommendations, the items were adapted as;
  - "*I can find uses for computer programming that are relevant for students*." → "I can find appropriate programming activities for students." and "I can select programming tools that are appropriate for students' programming skills." and
  - "*I can develop and plan effective computing lessons.*" →"I can develop effective computing lessons." and "I can plan effective computing lessons." respectively.

  Rich, Larsen, and Mason (2021) also state the reviewers' emphasis on the uncertainty about these items in the "teaching programming efficacy" factor as a limitation in the evaluation process of the original study.
- The item "*Knowledge of computer programming is NOT needed in most careers*." under the "value belief" factor was adapted as "Knowledge of computer science is necessary for making a career." and "Knowledge of computer science is NOT needed in most careers." since the word "career" in this item has both career and profession meanings in Turkish.
- The experts recommended adding three new items under the CT self-efficacy factor to measure the algorithm design, abstraction, and data collection components of CT. The original form of the scale consists of decomposition, generalization, problem-solving, and coding concepts. These are, respectively, "I can generalize by identifying common features of data or events in solving a problem.", "I can divide the solution of a problem into steps and order these steps logically.", "I can determine the requirements for solving a problem."

After the Turkish form of the scale was finalized, it was sent to a different field expert fluent in Turkish and English. The Turkish form was translated back into English by this expert, and the re-translated version of the scale was compared with the original version and revised with final corrections. The final version of the scale was administered to 20 teachers to assess its usefulness, and it was determined that no revisions were needed. The final version of the Turkish version of the scale is presented in Appendix 1.

3.4. Data Analysis

The data were collected through Google Forms, and the participation of teachers was entirely voluntary. To mitigate potential issues such as cyber faking, which can affect the validity of responses, we implemented rigorous data-cleaning procedures to identify and exclude outliers, thereby ensuring the robustness of our analyses. The data collected from 417 teachers were first analysed for outlier data using Mahalanobis distance. Accordingly, 43 teachers' data were identified as outliers and removed from the data set. With a sample size of 374 participants, the study achieved a 95% confidence level and a margin of error of ±5%. Skewness and kurtosis coefficients were calculated on the remaining 374 teachers' data set to examine normality assumptions. In the literature, ±3 for skewness and ±10 for kurtosis are taken as references (Kline, 2005). The skewness values of the data set vary between -1.852 and 0.051, and the kurtosis values vary between -1.538 and 2.881. Kaiser-Meyer-Olkin (KMO) and Bartlett's Test of Sphericity tests were used to determine the suitability of the data for confirmatory factor analysis.

Accordingly, the KMO value (0.971) and Bartlett's Test of Sphericity (25497.18) values were found statistically significant ($p < .000$). The obtained values were accepted as an indication that CFA could be performed for the validity studies of the TBaCCT. Independent samples t-test, one-way ANOVA, and MANOVA analysis were used to examine whether the scores obtained from the scale for programming self-efficacy, teaching programming efficacy, value belief, and CT self-efficacy differed according to gender and subject, respectively. SmartPLS 4 was used for CFA, and Jamovi (2022) statistical analysis program was used for descriptive and inferential statistics.

4. Findings

In this section, the results of the adaptation of the TBaCCT scale and the examination of teachers' value beliefs about CT and programming according to gender and subject are presented.

4.1. Validity and Reliability Studies of the TBaCCT Turkish Form

The four-factor model was examined using CFA. In CFA, various fit indices were used to evaluate the model's fit to the data. The fit indices CFI and TLI vary between 0 and 1. Values between 0.90 and 0.95 are acceptable, and above 0.95 indicate a good fit (Bentler, 1990; Hu & Bentler, 1999; Marsh et al., 2004). SRMR ≤ 0.08 indicates consistency (Karaman, 2023). RMSEA ≤ 0.08 is widely accepted, and values < 0.05 are considered good, 0.05-0.08 acceptable, 0.08-0.1 borderline, and > 0.1 bad (Fabrigar et al., 1999; Schermelleh-Engel et al., 2003; Sümer, 2000). The 39 items were analysed, and the factor loadings and the model are given in Figure 1. The RMSEA value of the model was calculated as 0.077. When the values obtained from the analysis [chi-square (N = 374) = 2262, $p < 0.001$, RMSEA = 0.079, SRMR = 0.075, CFI = 0.940, TLI = 0.934] were analysed, the fit criteria were calculated within acceptable ranges. The factor loadings of items VB4 (0.686) and VB10 (0.483) under the value belief factor in the scale are below 0.70. It is a common problem to obtain weak indicator loadings in measurements in social science studies (Hulland, 1999). However, since indicators with very low loadings (below 0.40) should be permanently eliminated from the measurement model, these two items were not removed from the model (Hair et al., 2022). Except for these two items, the factor loadings ranged between 0.752 and 0.977.

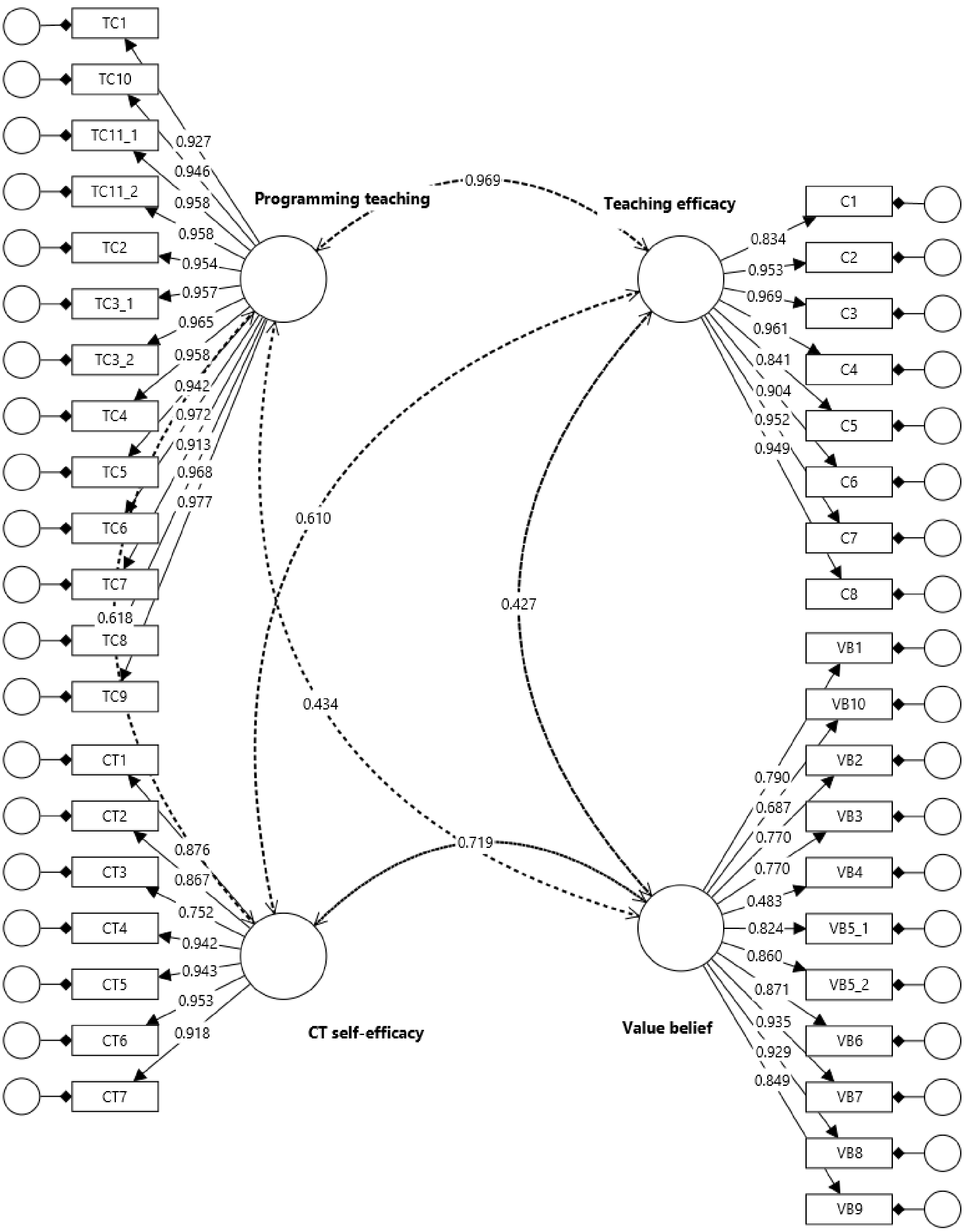

Fig. 1. Confirmatory factor analysis model of the TBaCCT Turkish Form.

Fig. 1 Alt Text: *The diagram shows the confirmatory factor analysis model of the Turkish version of the TBaCCT, consisting of 39 items and four factors. The diagram also shows the factor loadings of the items in the model and the reliability coefficients of the factors.*

Cronbach Alpha is widely used to determine internal consistency. For the reliability study of the scale, the Cronbach Alpha coefficient was calculated as 0.984. According to the factor scores, Cronbach's Alpha coefficient was calculated as α = 0.978 for programming self-efficacy, α = 0.992 for teaching programming efficacy, α = 0.952 for value belief, α = 0.965 for CT self-efficacy, and according to these results, the reliability coefficients of the scale show excellent values (Sarmento & Costa, 2017). In addition, composite reliability (rho_c) was used to determine the reliability of each item in the model. The composite reliability values were calculated as programming self-efficacy: 0.979; teaching programming efficacy: 0.992; value belief: 0.946; and CT self-efficacy: 0.966, respectively, which is above the critical threshold of 0.70 (Hair et al., 1998; Thompson et al., 1995). Convergent validity was assessed using average variance extracted (AVE) (Bagozzi & Yi, 1988; Hair et al., 2010). The convergent validity value AVE is expected to exceed the lower limit of 0.50. The values for all factors of the scale are above 0.50 (programming self-efficacy: 0.850; teaching programming efficacy: 0.909; value belief: 0.650; and CT self-efficacy: 0.802). The validity and reliability evidence obtained for the test scores indicate that the Turkish form of the TBaCCT Scale can be used to measure teachers' value beliefs about CT and programming.

4.2. Teachers' Value Beliefs about CT and Programming According to Gender and Subject

Teachers' value beliefs about CT and programming were examined regarding gender and subject. First, the scores given by the teachers for each factor were analysed through descriptive statistics based on items (see Appendix 1). Descriptive statistics for total scores based on factors are given in Table 2.

Table 2.

Descriptive statistics for scores obtained from the scale for programming self-efficacy, teaching programming efficacy, value belief, and CT self-efficacy.

	N	x̄	SD	Min	Max
Programming self-efficacy	374	3.83	1.74	1	6
Teaching programming efficacy	374	3.79	1.80	1	6
Value belief	374	4.80	1.13	1	6
CT self-efficacy	374	4.79	1.20	1	6

When the density distribution of the scores obtained from the scale for programming self-efficacy, teaching programming efficacy, value belief, and CT self-efficacy is examined (Figure 2), teachers' value belief and CT self-efficacy scores are higher than programming self-efficacy and teaching programming efficacy scores.

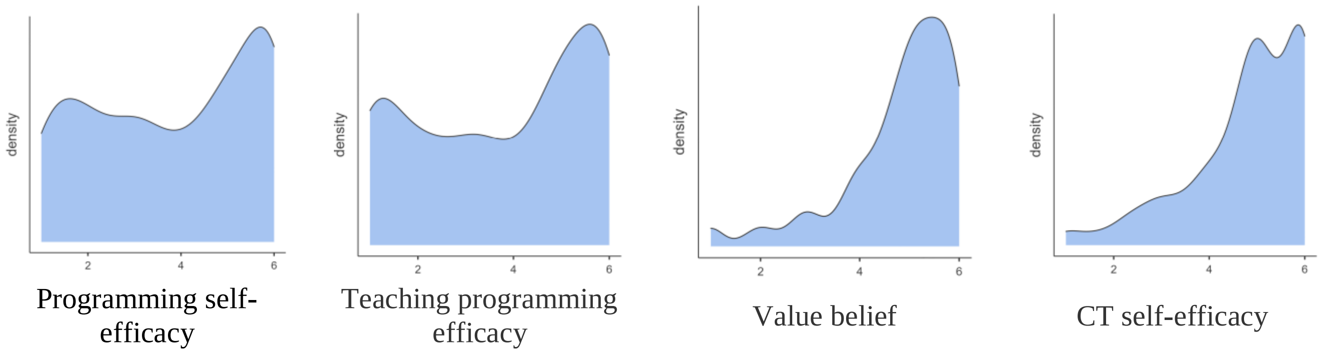

Fig. 2. The density distribution of scores obtained from the scale for programming self-efficacy, teaching programming efficacy, value belief, and CT self-efficacy.

Fig. 2 Alt Text: *Graphs displaying the density distribution of scores acquired from the scale for programming self-efficacy, teaching programming efficacy, value belief, and CT self-efficacy, illustrating varying distributions for each measured attribute.*

When the distribution of the scores obtained from the scale for programming self-efficacy, teaching programming efficacy, value belief, and CT self-efficacy according to gender is examined (Figure 3), male teachers have higher mean scores than female teachers in all factors. A one-way MANOVA was conducted to examine the effect of gender on four dependent variables: programming self-efficacy (PSE), teaching efficacy (TE), value beliefs (VB), and computational thinking self-efficacy (CTSE). Box's test of equality of covariance matrices was significant, $\chi^2(10) = 46.9$, $p < .001$, indicating a violation of the assumption of homogeneity of covariance matrices. Therefore, Pillai's Trace was used as the multivariate test statistic. The MANOVA revealed a statistically significant multivariate effect of gender on the combined dependent variables, Pillai's Trace = 0.0404, $F(4, 369) = 3.88$, $p = .004$.

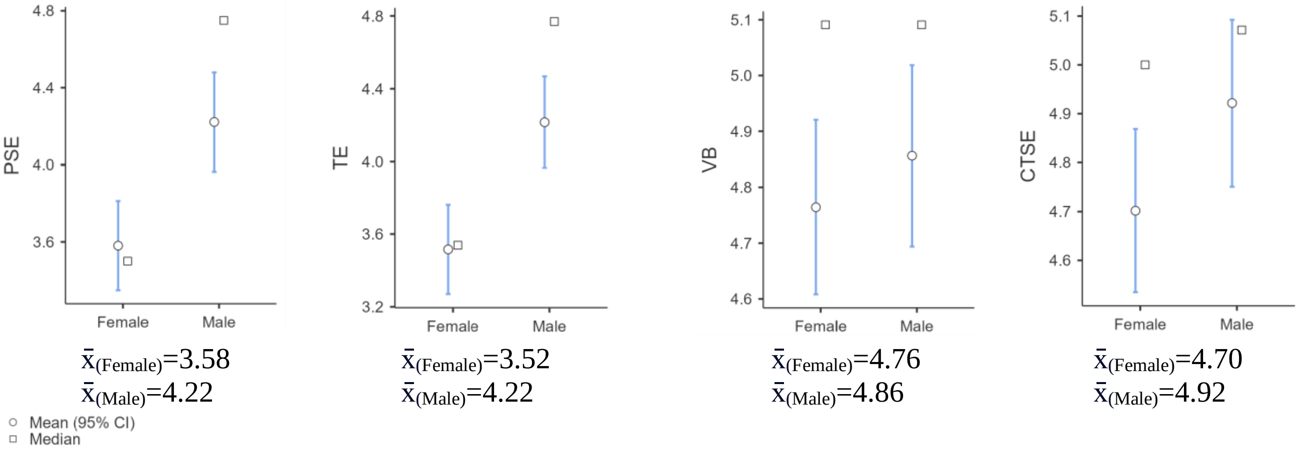

Fig. 3. Distribution of teacher beliefs about CT and programming by gender.
Fig. 3 Alt Text: *Graphs showing the distribution of teachers' beliefs about CT and programming by gender reveal the comparative distribution of the mean values of beliefs between different genders.*

An independent samples t-test was conducted to examine whether the difference between the averages was significant. Table 3 shows independent sample t-tests comparing male and female participants across four factors: PSE, TE, VB, and CTSE. Since the variances were not homogeneous, the results were interpreted with Welch's t-test instead of F statistics (Kohr & Games, 1974). Accordingly, there is a significant difference between male and female teachers for programming self-efficacy and teaching programming efficacy (t = -3,626, DF = 335, p <.001 for programming self-efficacy; t = -3,906, DF= 349, p <.001 for teaching programming efficacy). This difference favours male teachers. Male teachers consider themselves more competent in programming and using programming in their teaching. However, there is no significant difference between male and female teachers regarding value belief and CT self-efficacy factors. The Cohen's d effect size of the difference was calculated as 0.37 for programming self-efficacy and 0.40 for teaching programming efficacy. These differences have a moderate effect.

Table 3.

Independent samples t-test.

		Statistic	df	p	Mean difference	SE difference		Effect Size
PSE	Student's t	-3.533 [a]	372	< .001	-0.6412	0.181	Cohen's d	0.3744
	Welch's t	-3.626	335	< .001	-0.6412	0.177	Cohen's d	0.3793
TE	Student's t	-3.744 [a]	372	< .001	-0.7005	0.187	Cohen's d	0.3968
	Welch's t	-3.906	349	< .001	-0.7005	0.179	Cohen's d	0.4052
VB	Student's t	-0.767	372	0.444	-0.0918	0.120	Cohen's d	0.0813
	Welch's t	-0.799	348	0.425	-0.0918	0.115	Cohen's d	0.0829
CTSE	Student's t	-1.729 [a]	372	0.085	-0.2200	0.127	Cohen's d	0.1833
	Welch's t	-1.806	350	0.072	-0.2200	0.122	Cohen's d	0.1872

Note. H_a $\mu_{Female} \neq \mu_{Male}$

A MANOVA was conducted to examine the effect of the independent variable (subject) on four dependent variables: PSE, TE, VB, and CTSE. Box's test of equality of covariance matrices was significant, $\chi^2(70)$ = 209, p < .001, indicating a violation of the assumption of homogeneity of covariance matrices. As a result, Pillai's Trace was used as the preferred multivariate test statistic. The MANOVA revealed a statistically significant multivariate effect of gender on the combined dependent variables, Pillai's Trace = 0.591, F(28, 1464) = 9.06, p < .001.

One-way ANOVA analysis was conducted to examine whether the scores obtained from the scale for PSE, TE, VB, and CTSE differed significantly according to the subject. Since the variances were not homogeneous, the results were interpreted with Welch's t-test instead of F statistics (Kohr & Games, 1974). The results show that teachers' scores obtained from the scale for programming PSE, TE, VB, and CTSE differed significantly according to the subject (Welch's F (7, 374) = 45.51, p = < .001 for programming self-efficacy; Welch's F (7, 374) = 39.37, p = < .001 for teaching programming efficacy; Welch's F (7, 374) = 5.38, p = < .001 for value belief; Welch's F (7, 374) = 8.39, p = < .001 for CT self-efficacy). Since the variances were not homogeneous, the Games-Howell post-hoc test was used to examine between which groups the difference thus exists. According to this test;

- For programming self-efficacy, there is a significant difference between computer science teachers ($\bar{x}$ = 41.1) and all other subjects. In addition, the scores of elementary mathematics ($\bar{x}$ = 27.2) and science ($\bar{x}$ = 25.6) teachers are significantly higher than those of social studies ($\bar{x}$ = 16.6), native language ($\bar{x}$ = 16.2) and foreign language ($\bar{x}$ = 13.7) teachers. Finally, the scores of primary education teachers ($\bar{x}$ = 25.3) are significantly higher than the scores of native language and foreign language teachers.
- There is a significant difference between computer science teachers ($\bar{x}$ = 66.1) and all other subjects for teaching programming efficacy. In addition, the scores of elementary mathematics teachers ($\bar{x}$ = 42.4) are significantly higher than those of social studies ($\bar{x}$ = 26.3) and native language ($\bar{x}$ = 25.1) teachers. Finally, there is a significant difference between the scores of science teachers ($\bar{x}$ = 40.1), primary education teachers ($\bar{x}$ = 43.9) and native language teachers.
- After computer science teachers ($\bar{x}$ = 56.8), the highest mean for value belief belongs to elementary mathematics ($\bar{x}$ = 53.6) and primary education ($\bar{x}$ = 50.2) teachers. There is a significant difference between the average scores of computer science teachers on this factor and the scores of science teachers ($\bar{x}$ = 49.5) and social studies teachers ($\bar{x}$ = 46.5). Apart from this, there is no significant difference between other subjects.
- For CT self-efficacy, after computer science teachers ($\bar{x}$ = 36.4), the highest mean belongs to elementary mathematics ($\bar{x}$ = 35.1) and primary education ($\bar{x}$ = 32.7) teachers. There is a significant difference between the average scores of computer science teachers on this factor and the scores of science teachers ($\bar{x}$ = 31.8) and social studies teachers ($\bar{x}$ = 27.6). Apart from this, there is no significant difference between other subjects.

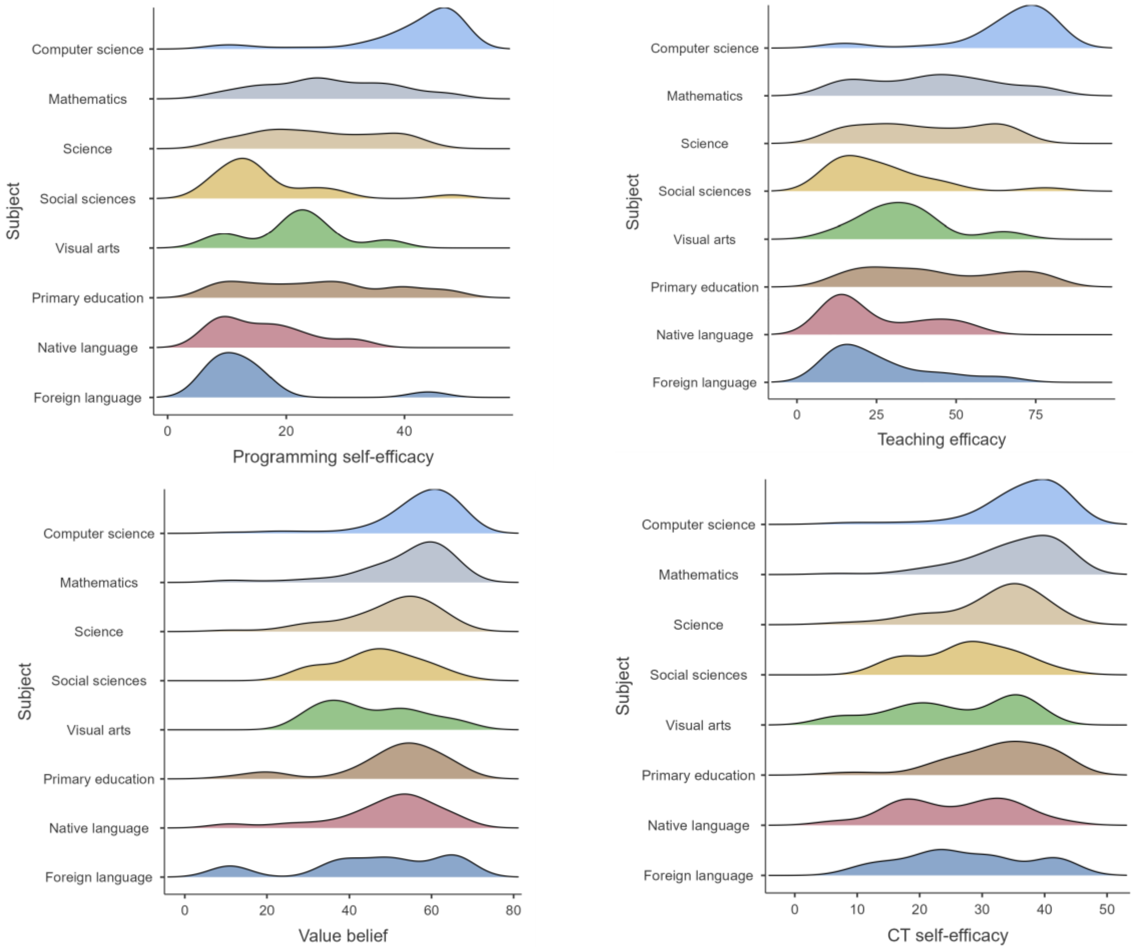

Fig. 4. The density distribution of scores for programming self-efficacy, teaching programming efficacy, value belief, and CT self-efficacy according to the subject.

Fig. 4 Alt Text: *Graphs illustrating the density distribution of scores related to programming self-efficacy, teaching programming efficacy, value belief, and CT self-efficacy categorized by subject, showcasing the spread and variation of scores across different subject categories.*

When the distribution of the scores obtained from the scale for programming self-efficacy, teaching programming efficacy, value belief, and CT self-efficacy is examined according to the subject, computer science teachers have the highest scores in all sub-factors. This is followed by the scores of elementary mathematics teachers, primary education teachers, and science teachers, respectively. Native language and foreign language teachers have the lowest scores in all sub-factors. In addition to computer science teachers, elementary mathematics, primary education, and science teachers consider themselves more competent in CT and programming than other subject teachers.

5. Discussion

Teachers' professional development plays a crucial role in integrating CT into their teaching (Mumcu et al., 2023; Wang et al., 2022). Drawing from self-efficacy theory (Bandura, 1977), we argue that teachers' value beliefs and self-efficacy are key drivers in how successfully they implement CT and programming in their classrooms. The expectancy-value theory (Boström & Plam, 2020) further highlights the importance of teachers' beliefs in their ability to achieve positive outcomes in their teaching. Effective professional development programs must consider teachers' value judgments and beliefs about CT and programming (Cabrera, 2019). In this study, we adapted the TBaCCT scale developed by Rich, Larsen, and Mason (2021) to measure these beliefs among Turkish teachers and explored how factors such as gender and subject impacted their perceptions of CT and programming.

Our findings indicate that teachers generally hold strong value beliefs about CT and possess CT self-efficacy. However, their programming self-efficacy and teaching programming efficacy scores were lower.

This suggests that while teachers recognize the importance of CT and feel capable of applying its principles to problem-solving, they are less confident in their programming abilities and teaching programming. This gap between value beliefs and self-efficacy aligns with expectancy-value theory, which suggests that while teachers may value CT, they may still require additional support to build confidence in programming teaching. Programming training is a commonly used method to improve teachers' CT skills and their ability to teach CT (e.g., Broza et al., 2023; Özdinç et al., 2022). Despite programming being essential for developing CT, beginning teachers often face challenges in designing and implementing programming activities (Sung & Jeong, 2019). Studies have shown positive outcomes when CT and programming are included in teacher education. For instance, Alqahtani et al. (2022) found that a CT training program incorporating physical programming activities positively influenced pre-service mathematics teachers' perceptions, attitudes, and intentions to use programming in their teaching. Given that CT and programming education are still in the early stages and are not yet a mandatory component of teacher education, these results are understandable. Nonetheless, it underscores the need for enhanced professional development to support teachers in building both their CT and programming teaching capabilities.

While there is a significant difference between the beliefs of male and female teachers regarding programming self-efficacy and teaching programming efficacy sub-factors, there is no significant difference between value belief and CT self-efficacy. It is difficult to interpret this result since studies emphasizing gender differences in programming self-efficacy or teaching programming efficacy for teachers are limited. Yadav et al. (2014) and Villalustre and Cueli (2023) found that male pre-service teachers were more interested in programming than female pre-service teachers. However, Villalustre and Cueli (2023) also discovered that female pre-service teachers who have used programming languages before and had experience have better CT skills than male pre-service teachers. Similarly, Sun et al. (2022) revealed that students' programming attitudes significantly predict their CT skills, and female students who received programming education had better CT skills than male students. Despite findings that suggest results against women, we see that those with proper education and experience surpass men in these areas. Yadav et al. (2014)'s assertion that these discrepancies stem from gender imbalances in computer science is valid. To address this, we must provide more opportunities for women to receive training in this field. The gender variable remains critical in developing programming and CT skills, highlighting the importance of expanding educational opportunities for women in this domain.

According to the subject taught, there are differences in teachers' value beliefs regarding programming self-efficacy, teaching programming efficacy, value beliefs, and CT self-efficacy. Computer science teachers score the highest across all factors, followed by elementary mathematics teachers, primary education teachers, and science teachers. In contrast, native language and foreign language teachers have the lowest scores in all sub-factors. There is a significant difference between computer science teachers and all other subjects in terms of programming self-efficacy and teaching programming efficacy. Furthermore, the programming self-efficacy scores of elementary mathematics and science teachers are significantly higher than those of social studies, native language, and foreign language teachers. Additionally, elementary mathematics teachers' teaching programming efficacy scores are significantly higher than those of social studies and native language teachers. Computer science teachers, elementary mathematics teachers, primary education teachers, and science teachers perceive themselves as more competent in programming and teaching programming than other subject teachers. These subjects are primarily the focus of STEM education research, reflecting one of the trends in CT research—its integration into STEM education (Tekdal, 2021). CT is a crucial skill for problem-solving and has a strong impact on STEM fields. Günbatar and Bakırcı (2019) revealed that CT had the most significant effect on teachers' STEM teaching intentions, further emphasizing its importance in shaping educational practices. Moreover, the subjects that teachers teach play a significant role in their motivation to teach CT (Fagerlund et al., 2022), with teachers of STEM-related subjects being the most motivated to integrate CT into their teaching. This alignment is due to both areas emphasizing problem-solving skills (Mumcu et al., 2023; Zhou et al., 2020). Additionally, it is possible that this shared focus has become so deeply ingrained in teaching practices that educators have internalized it, taking for granted the mutual relevance of these skills in their disciplines.

Although teaching CT can be seen as the responsibility of computer science teachers, it is not a compulsory part of K12 school education, and these teachers often lack familiarity with CT concepts, necessitating professional development (Good et al., 2017; Ni et al., 2023). Still, they have better scores than other subject teachers and mathematics and science teachers follow them. Programming education, STEM education, or out-of-school learning are utilized to develop CT (Kong et al., 2019). In addition, computer science education at the primary level typically integrates CT into core subjects such as mathematics and science (Luo et al., 2023). Considering the visibility of teacher education projects (Interdisciplinary Teacher Academy [ITA], 2023) and examples of ready-to-implement activities (Mumcu et al., 2022b), as well as studies focusing on teachers' professional development (Mumcu et al., 2023), it is not surprising that mathematics and science teachers have higher programming self-efficacy and teaching programming efficacy than other subjects. Rich et al. (2019) found that mathematics and science teachers see more robust connections between CT and mathematics teaching and science teaching than with other subjects. They also found that teachers drew on their existing knowledge of terminology related to CT to make connections to mathematics and science teaching. In this context, it can be interpreted as an expected result that mathematics and science teachers' value beliefs about CT and programming are higher than other subjects. However, teachers working in social sciences need professional development programs that will transform their beliefs and knowledge about CT. Given that all disciplines are undergoing a computational transformation today (e.g., computational linguistics), this topic needs to be studied in the future.

While the results obtained for computer science teachers can be taken for granted, Alfayez and Lambert (2019), in their study with computer science teachers, found that most teachers had a low conceptual level of CT, and some teachers had misconceptions about the exact nature of CT. At the same time, Alfayez and Lambert (2019) state that computer science teachers need more training on what CT means and how to teach it. Similarly, Yadav et al. (2016) state that computer science teachers face several challenges, such as a lack of sufficient computer science background and limited professional development opportunities. Computer science teachers in Turkiye face similar problems (Sadık et al., 2016). In conclusion, although computer science teachers in this study have higher beliefs about CT and programming than other subjects, the future of computer science education in Turkiye, the training of these teachers, and even the course content are issues that need to be discussed and revised. Future studies must examine computer science teachers' current practices and needs in this context.

According to Brennan and Resnick (2012), the CT framework has three dimensions: *concepts, practices*, and *perspectives*, and should be considered in assessing programming and CT skills. These dimensions are closely related to programming education, which is part of computer science education. In recent years, programming and CT have been seen as an integral part of each other. Considering recent studies that assume that programming and CT are inseparable (e.g., Tamborg et al., 2022), professional development programs should be designed to consider the value beliefs of teachers. This adapted scale can measure the impact of professional development programs for CT, including teaching efficacy, or it can be used to design professional development programs by determining teachers' value beliefs about CT and programming. However, this scale's constructs related to CT focus more on the affective domain. It is crucial to develop measurement tools that focus on the cognitive domain, such as CT competence as well as the affective domain (Wang et al., 2022).

The integration of CT and programming in teaching is a central focus of this research, as reflected in the adaptation of the TBaCCT scale. CT plays a critical role in today's education, particularly in STEM fields, where problem-solving and computational thinking are essential skills. By developing a tool that measures teachers' beliefs about CT and programming, this study provides a foundation for future educational programs that aim to enhance teachers' confidence and competence in these areas. The findings underscore the importance of professional development that not only improves teachers' CT skills but also strengthens their ability to teach programming effectively.

6. Implications, Limitations, and Future Studies

6.1. Implications

Based on the discussed results, the following implications for practice emerge from this research:

- Teachers' professional development should be supported in the field of CT and teaching programming through professional development programs (workshops, training, seminars), given that they are not yet a compulsory part of teacher education;
- Special support in the form of professional development programs should be given to female teachers and those who teach social sciences in order to transform and improve their value beliefs about CT and programming;
- The current teaching practice and the needs of teachers should be checked and addressed, given that the future of computer science education in Turkiye, the training of teachers, and the course content are issues that should be further considered and revised;
- Considering recent studies that assume that CT and programming are inseparable, professional training programs should be re-designed in that way to include programming teaching so that teachers can integrate CT into their practice.

6.2. Limitations and Future Studies

This study shows that items Value Belief 4 (VB4) and Value Belief 10 (VB10) items predict the value belief sub-scale at a relatively critical level compared to the others. Reviewing or revising these two items may be among the subjects of future studies. Since the research emphasizing gender differences and subjects in programming self-efficacy or teaching efficacy for teachers is limited, it is also recommended to intensify and repeat them on other samples. Through the intensification and repetition of these researches, it should be explained why the differences in teachers' value beliefs within these aspects are caused by gender and subject. In addition to the above, it is recommended that this type of research be strengthened with an interview to gain deeper insights into the value beliefs of teachers (especially those from social sciences) about CT and programming, which would allow them freedom and breadth in their explanations. These data would guide researchers in interpreting and relating to previous claims in this area. Given that the affective domain within this issue has been researched the most in comparison to other domains (cognitive and psychomotor), developing new measuring instruments that would support other domains in education is recommended as a topic for future studies. For example, it is recommended to develop an instrument that would include the measurement of variables from the cognitive domain, such as teacher competencies in CT and programming, as well as the psychomotor domain, such as developed teacher's skills in this field. Given that all disciplines are undergoing computational transformation today, it is recommended that the research on this issue be intensified in future studies by examining variables from all domains of education. Bearing this in mind, for the subject of future studies, it is also recommended to examine the current practice of all teachers and their needs within the field of CT and programming, considering the current state of IT education in schools, its future, teacher training programs as well as the content of the courses themselves. These are critical issues for the improvement and sustainability of this area that should be further considered and revised if necessary.

Acknowledgements

This study is part of the first author's postdoctoral research, supported by The Scientific and Technological Research Council of Turkey (TUBITAK) between 2022 and 2023 at Linz Johannes Kepler University.

References

Alfayez, A. A., & Lambert, J. (2019). Exploring Saudi Computer Science Teachers' Conceptual Mastery Level of Computational Thinking Skills. *Computers in the Schools*, *36*(3), 143–166. https://doi.org/10.1080/07380569.2019.1639593

Alqahtani, M. M., Hall, J. A., Leventhal, M., & Argila, A. N. (2022). Programming in Mathematics Classrooms: Changes in Pre-service Teachers' Intentions to Integrate Robots in Teaching. *Digital Experiences in Mathematics Education*, *8*(1), 70–98. https://doi.org/10.1007/s40751-021-00096-6

Aminger, W., Hough, S., Roberts, S. A., Meier, V., Spina, A. D., Pajela, H., ... & Bianchini, J. A. (2020). Preservice Secondary Science Teachers' Implementation of an NGSS Practice: Using Mathematics and Computational Thinking. *Journal of Science Teacher Education*, 1-22.

Avcı, C., & Deniz, M. N. (2022). Computational thinking: early childhood teachers' and prospective teachers' preconceptions and self-efficacy. *Education and Information Technologies*, *27*(8), 11689-11713.

Bagozzi, R. P., & Y. Yi. (1988). On the Evaluation of Structural Equation Models. *Journal of the Academy of Marketing Science*, *16*(1), 74–94. https://doi.org/10.1177/009207038801600107

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, *84*(2), 191–215. https://doi.org/10.1037/0033-295X.84.2.191

Bandura, A. (2006). Guide for constructing self-efficacy scales. In F. Pajares & T. Urdan (Eds.), *Self-efficacy beliefs of adolescents* (Vol. 5, pp. 307-337). Information Age Publishing.

Bentler P. M. (1990). Comparative fit indexes in structural models. *Psychological Bulletin*, *107*(2), 238–246. https://psycnet.apa.org/doi/10.1037/0033-2909.107.2.238

Bocconi, S., Chioccariello, A., & Earp, J. (2018). The Nordic approach to introducing Computational Thinking and programming in compulsory education. *Report prepared for the Nordic@ BETT2018 Steering Group*, *42*.

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M.A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V. and Stupurienė, G. (2022). *Reviewing Computational Thinking in Compulsory Education*. Inamorato dos Santos, A., Cachia, R., Giannoutsou, N. and Punie, Y. editor(s). Publications Office of the European Union, Luxembourg. ISBN 978-92- 76-47208-7, doi:10.2760/126955, JRC128347

Boström, E., & Palm, T. (2020). Expectancy-value theory as an explanatory theory for the effect of professional development programmes in formative assessment on teacher practice. *Teacher Development*, *24*(4), 539-558.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the *2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada (pp. 1–25).

Broza, O., Biberman-Shalev, L., & Chamo, N. (2023). "Start from scratch": Integrating computational thinking skills in teacher education program. *Thinking Skills and Creativity*, *48*, 101285.

Cabrera, L. (2019). Teacher Preconceptions of Computational Thinking: A Systematic Literature Review. *Journal of* Technology *and Teacher Education*, *27*(3), 305–333.

Chandra, V., & Lloyd, M. (2020). Lessons in persistence: Investigating the challenges faced by preservice teachers in teaching coding and computational thinking in an unfamiliar context. *Australian Journal of Teacher Education*, *45*(9), http://dx.doi.org/10.14221/ajte.2020v45n9.1.

Dagli, Z., & Sancar Tokmak, H. (2021). Exploring high school computer science course teachers' instructional design processes for improving students' "computational thinking" skills. *Journal of Research on Technology in Education*, 1-26.

Dimos, I., Velaora, C., Louvaris, K., Kakarountas, A., & Antonarakou, A. (2023). How a Rubric Score Application Empowers Teachers' Attitudes over Computational Thinking Leverage. *Information*, *14*(2), 118.

Durak, H. Y., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education, 116*, 191-202.

Espino, E. E. E., & González, C. G. (2016, September). Gender and computational thinking: Review of the literature and applications. In *Proceedings of the XVII International Conference on Human Computer Interaction* (pp. 1-2).

Fabrigar L. R., MacCallum R. C., Wegener D. T., Strahan E. J. (1999). Evaluating the use of exploratory factor analysis in psychological research. *Psychological Methods, 4*(3), 272–299. https://psycnet.apa.org/doi/10.1037/1082-989X.4.3.272

Fagerlund, J., Leino, K., Kiuru, N., & Niilo-Rämä, M. (2022, November). Finnish teachers' and students' programming motivation and their role in teaching and learning computational thinking. In *Frontiers in Education* (Vol. 7, p. 948783). Frontiers Media SA.

Fessakis, G., & Prantsoudi, S. (2019). Computer Science Teachers' Perceptions, Beliefs and Attitudes on Computational Thinking in Greece. *Informatics in Education*, *18*(2), 227-258.

Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2012). *How to design and evaluate research in education*. 8th Ed. McGraw-Hill. ISBN: 978-0-07-809785-0

Gabrielle, L., Bertacchini,F., Tavernise, A., Vaca-Cardenas, L., Pantano, P., & Bilotta, E. (2019). Lesson Planning by computational thinking skills in Italian pre-service teachers. *Informatics in Education, 18*(1), 69-104.

Good, J., Yadav, A., & Mishra, P. (2017). Computational thinking in computer science classrooms: Viewpoints from CS educators. In *Society for Information Technology & Teacher Education International Conference* (pp. 51-59). Association for the Advancement of Computing in Education (AACE).

Gülbahar, Y., Kert, S. B. ve Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısı ölçeği: geçerlik ve güvenirlik çalışması. *Türk Bilgisayar ve Matematik Eğitimi Dergisi*, *10*(1), 1-29.

Günbatar, M. S., & Bakırcı, H. (2019). STEM teaching intention and computational thinking skills of pre-service teachers. *Education and information technologies*, *24*, 1615-1629.

Guzdial, M. (2008). Education Paving the way for computational thinking. *Communications of the ACM*, *51*(8), 25-27.

Guzdial, M. (2016). *What Does Computing for Everyone Mean?*. In: Learner-Centered Design of Computing Education. Synthesis Lectures on Human-Centered Informatics. Springer, Cham. https://doi.org/10.1007/978-3-031-02216-6_1

Hair, J. F., Hult, T., Ringle, C. M., & Sarstedt, M. (2022). *A primer on partial least squares structural equation modeling (PLS-SEM)* (3rd ed.). Thousand Oaks: Sage.

Hair, J. F., Jr., R. E. Anderson, R. L. Tatham, & W. C. Black (1998). *Multivariate Data Analysis with Readings*, 5th Edition. Englewood Cliffs, NJ: Prentice Hall.

Hair, J. F., W. C. Black, B. J. Babin, & R. E. Anderson (2010). *Multivariate Data Analysis: A Global Perspective*. 7th ed. Upper Saddle Riwer, NJ: Prentice Hall.

Haşlaman, T., Mumcu, F., & Atman Uslu, N. (2024). Fostering Computational Thinking through Digital Storytelling: A Distinctive Approach to Promoting Computational Thinking Skills of Pre-Service Teachers. *Education and Information Technologies*. https://doi.org/10.1007/s10639-024-12583-5

Hershkovitz, A., Bain, C., Kelter, J., Peel, A., Wu, S., Horn, M.S. & Wilensky, U. (2023). Contribution of Computational Thinking to STEM Education: High School Teachers' Perceptions after a Professional Development Program. *Journal of Computers in Mathematics and Science Teaching, 42*(1), 35-65.

Hu, L. T., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: conventional criteria versus new alternatives. *Structural Equation Modeling*, *6*(1), 1-55. https://doi.org/10.1080/10705519909540118

Hulland, J. (1999). Use of partial least squares (PLS) in strategic management research: A review of four recent studies. *Strategic Management Journal*, *20*(2), 195-204.

Interdisciplinary Teacher Academy [ITA] (2023). http://interteacher.org/yayinlarimiz/.

International Society for Teacher Education [ISTE] (2016). *ISTE standards for students*. https://www.iste.org/standards/for-students

Kafai, Y. B., & Proctor, C. (2022). A revaluation of computational thinking in K–12 education: Moving toward computational literacies. *Educational Researcher*, *51*(2), 146-151.

Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic J. Modern Computing*, *4*(3), 583–596.

Karaman, M. (2023). Keşfedici ve doğrulayici faktör analizi: Kavramsal bir çalışma. *Uluslararası İktisadi ve İdari Bilimler Dergisi, 9*(1), 47-63.

Ketelhut, D. J., Mills, K., Hestness, H., Cabrera, L., Plane, J., McGinnis, J. R. (2020). Teacher change following a professional development experience in integrating computational thinking into elementary science. *Journal of Science Education and Technology, 29*, 173–187.

Klassen, R. M., & Tze, V. M. (2014). Teachers' self-efficacy, personality, and teaching effectiveness: A meta-analysis. *Educational Research Review*, *12*, 59-76.

Kline, R. B. (2005). Principles and practice of structural equation *modeling* (2nd Ed.). New York, NY: Guilford Press.

Kohr, R. L., & Games, P. A. (1974). Robustness of the analysis of variance, the Welch procedure and a Box procedure to heterogeneous variances. *The Journal of Experimental Education*, *43*(1), 61-69. https://doi.org/10.1080/00220973.1974.10806305

Kong, S.C., Abelson, H., Lai, M. (2019). *Introduction to Computational Thinking Education. In: Kong, S.C., Abelson, H. (eds) Computational Thinking Education*. Springer, Singapore. https://doi.org/10.1007/978-981-13-6528-7_1

Kong, S.-C., Lai, M., & Li, Y. (2023). Scaling up a teacher development programme for sustainable computational thinking education: TPACK surveys, concept tests and primary school visits. *Computers & Education*, *194*, 104707. https://doi.org/10.1016/j.compedu.2022.104707

Korkmaz, Ö., Çakır, R., & Özden, Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior* 72, 558-569.

Kukul, V., Gökçearslan, Ş., & Günbatar, M. S. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. *Eğitim Teknolojisi Kuram ve Uygulama*, *7*(1), 158-179.

Laime-Choque, A. M., Mamani-Calcina, J. G., Cardona-Reyes, H., Ponce-Aranibar, M. D. P., Vera-Vasquez, C. G., & Espinoza-Suarez, S. (2022, September). Attitude towards Computational Thinking of in-service teachers. In *2022 XII International Conference on Virtual Campus (JICV)* (pp. 1-5). IEEE.

Lee, I., & Malyn-Smith, J. (2020). Computational Thinking Integration Patterns Along the Framework Defining Computational Thinking from a Disciplinary Perspective. *Journal of Science Education and Technology*, *29*(1), 9–18. https://doi.org/10.1007/s10956-019-09802-x

Li, Q., McNary, S. W., & Boyd, T. (2023). Assessment of Computational Thinking: A Study of Preservice Teachers' Knowledge and Beliefs. Athens Journal of Sciences, 10, 65-82.

Lodi, M. (2020). Informatical Thinking. *Olympiads In Informatics*, 113–132. https://doi.org/10.15388/ioi.2020.09

Lodi, M., & Martini, S. (2021). Computational Thinking, Between Papert and Wing. *Science & Education*, *30*(4), 883–908. https://doi.org/10.1007/s11191-021-00202-5

Luo, F., Ijeluola, S. A., Westerlund, J., Walker, A., Denham, A., Walker, J., & Young, C. (2023). Supporting Elementary Teachers' Technological, Pedagogical, and Content Knowledge in Computational Thinking Integration. *Journal of Science Education and Technology*. https://doi.org/10.1007/s10956-023-10045-0

Marsh, H. W., Hau, K. T., & Wen, Z. (2004). In search of golden rules: Comment on hypothesis-testing approaches to setting cutoff values for fit indexes and dangers in overgeneralizing Hu and Bentler's (1999) findings. *Structural equation modeling*, *11*(3), 320-341. https://doi.org/10.1207/s15328007sem1103_2

Ministry of National Education (2018). *Information Technologies and Software Course Curriculum*. Ankara. https://mufredat.meb.gov.tr/Dosyalar/202312491032309-Bili%C5%9Fim%20Teknolojileri%20ve%20Yaz%C4%B1l%C4%B1m.pdf#page=5.07 (in Turkish)

Mumcu, F., Uslu, N. A., Özdinç, F., & Yıldız, B. (2022a). Exploring teacher development courses in the lens of integrated STEM education: A holistic multiple case study. *International Journal of Contemporary Educational Research*, *9*(3), 476-491. https://doi.org/10.33200/ijcer.1035464

Mumcu, F., Yıldız, B., Özdinç, F. ve Kaya, G. (2022b). *Disiplinlerarası Eğitim Örnekleri III – Disiplinlerarası Öğretmen Akademisi Etkinlikleri*. (Ocak 2022). ISBN: 978-625-00-9466-2.

Ni, L., Bausch, G., & Benjamin, R. (2023). Computer science teacher professional development and professional learning communities: A review of the research literature. *Computer Science Education*, *33*(1), 29–60. https://doi.org/10.1080/08993408.2021.1993666

Özdinç, F., Kaya, G., Mumcu, F., & Yıldız, B. (2022). Integration of computational thinking into STEM activities: An example of an interdisciplinary unplugged programming activity. *Science Activities – Projects and Curriculum Ideas in STEM Classrooms, 59*(3), 151-159. DOI: 10.1080/00368121.2022.2071817

Prado, Y., Jacob, S., & Warschauer, M. (2022). Teaching computational thinking to exceptional learners: lessons from two inclusive classrooms. *Computer Science Education*, *32*(2), 188-21

Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). *Computational Thinking, Mathematics, and Science: Elementary Teachers' Perspectives on Integration*. *27*(2), 165–205.

Rich, P. J., Larsen, R. A., & Mason, S. L. (2021). Measuring teacher beliefs about coding and computational thinking. *Journal of Research on Technology in Education*, *53*(3), 296-316.

Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions. In S.-C. Kong & H. Abelson (Ed.), Computational Thinking Education (pp. 79–98). Springer Singapore.

Sadık, O., Kıdıman, E., Işık, A., & Topcu, A. E. (2016). Bilgisayar bilimi müfredatı oluşturmaya yönelik bilişim teknolojileri öğretmenlerinin görüşleri. *10. Uluslararası Bilgisayar ve Öğretim Teknolojileri Eğitimi Sempozyumu – ICITS 2016*, 16 – 18 Mayıs, Rize, Türkiye.

Saritepeci, M., & Durak, A. (2022). Adaptation of T-STEM CT scale to Turkish: Teacher self-efficacy and outcome expectancy for teaching computational thinking. *Research on Education and Psychology*, *6*(Special Issue), 47-56.

Sarmento, R. P., & Costa, V. (2019). Confirmatory factor analysis--a case study. *arXiv preprint* arXiv:1905.05598.

Schermelleh-Engel, K., Moosbrugger, H., & Müller, H. (2003). Evaluating the fit of structural equation models: Test of significance and descriptive goodness-of-fit measures. Methods of Psychological Research Online, 8(2), 23-74. University of Koblenz-Landau.

Seow, P., Looi, CK., How, ML., Wadhwa, B., Wu, LK. (2019). *Educational Policy and Implementation of Computational Thinking and Programming: Case Study of Singapore*. In: Kong, SC., Abelson, H. (eds) Computational Thinking Education. Springer, Singapore. https://doi.org/10.1007/978-981-13-6528-7_19

Sun, L., Hu, L., & Zhou, D. (2022). Programming attitudes predict computational thinking: Analysis of differences in gender and programming experience. *Computers & Education*, *181*, 104457.

Sung, Y.-H., & Jeong, Y.-S. (2019). Development and Application of Programming Education Model Based on Visual Thinking Strategy for Pre-service Teachers. *Universal Journal of Educational Research*, *7*(5A), 42–53. https://doi.org/10.13189/ujer.2019.071507

Surahman, E., Wang, T. H., Lam, K. F. T., Pratama, U. N., & Aisyah, E. N. (2022, January). Investigating Elementary Teachers' Understanding, Beliefs, and Intentions toward STEM and Computational Thinking in Education. In *2022 2nd International Conference on Information Technology and Education (ICIT&E)* (pp. 175-180). IEEE.

Sümer, N. (2000). *Yapısal eşitlik modelleri: Temel kavramlar ve örnek uygulamalar*. Türk Psikoloji Yazıları, 3(6), 49 -74.

Tamborg, A. L., Elicer, R., & Spikol, D. (2022). Programming and Computational Thinking in Mathematics Education: An Integration Towards AI Awareness. *KI - Künstliche Intelligenz*, *36*(1), 73–81. https://doi.org/10.1007/s13218-021-00753-3

Tekdal, M. (2021). Trends and development in research on computational thinking. *Education and Information Technologies*, *26*(5), 6499–6529. https://doi.org/10.1007/s10639-021-10617-w

The jamovi project (2022). *jamovi*. (Version 2.3) [Computer Software]. Retrieved from https://www.jamovi.org.

Thompson, R., D. W. Barclay, and C. A. Higgins (1995) "The Partial Least Squares Approach to Causal Modeling: Personal Computer Adoption and Use as an Illustration," Technology Studies: Special Issue on Research Methodology (2) 2 (Fall), pp. 284-324.

Tran, Y. (2019). Computational Thinking Equity in Elementary Classrooms: What Third-Grade Students Know and Can Do. *Journal of Educational Computing Research*, *57*(1), 3–31. https://doi.org/10.1177/0735633117743918

Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training Computational Thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games*, *5*(2), Article 2. https://doi.org/10.17083/ijsg.v5i2.248

Tschannen-Moran, M., & Woolfolk Hoy, A. (2001). Teacher efficacy: Capturing an elusive construct. Teaching and Teacher Education, 17(7), 783-805.

Umutlu, D. (2021). An exploratory study of pre-service teachers' computational thinking and programming skills. *Journal of Research on Technology in Education*, 1-15.

Villalustre, L., & Cueli, M. (2023). Assessing the Computational Thinking of Pre-Service Teachers: A Gender and Robotics Programming Experience Analysis. *Education Sciences*, *13*(10), 1032.

Wang, C., Shen, J., & Chao, J. (2022). Integrating Computational Thinking in STEM Education: A Literature Review. *International Journal of Science and Mathematics Education*, *20*(8), 1949–1972. https://doi.org/10.1007/s10763-021-10227-5

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer Science in K-12 school curricula of the 21st century: Why, what and when? Education and Information Technologies, 22(2), 445-468.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer Science Education*, *26*(4), 235–254. https://doi.org/10.1080/08993408.2016.1257418

Yadav, A., Hong, H., & Stephenson, C. (2017). Computational Thinking for Teacher Education. Communications of the ACM, 60(4), 55-62.

Yadav, A., Larimore, R., Rich, K., & Schwarz, C. (2019, March). Integrating computational thinking in elementary classrooms: Introducing a toolkit to support teachers. In *Society for Information Technology & Teacher Education International Conference* (pp. 347-350). Association for the Advancement of Computing in Education (AACE).

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational Thinking in Elementary and Secondary Teacher Education. *ACM Transactions on Computing Education*, *14*(1), 5:1-5:16. https://doi.org/10.1145/2576872

Yagci, M. (2019). A valid and reliable tool for examining computational thinking skills. *Education and Information Technologies*, *24*(1), 929–951.

Yılmaz, T., & İzmirli, S. (2023). Effect of unplugged and plugged coding activities on secondary school students' computational thinking skills. *Journal of Educational Technology and Online Learning*, *6*(4), 1180-1193.

Yılmaz, F. G. K., Yılmaz, R., & Durak, H. Y. (2018). A review on the opinions of teachers about the development of computational thinking skills in K-12. *Teaching computational thinking in primary education*, 157-181.

Zha, S., Jin, Y., & Moore, P., & Gaston, J. (2020). Hopscotch into coding: Introducing pre-service teachers computational thinking. *Techtrends, 64,* 17-28.

Zhou, D., Gomez, R., Wright, N., Rittenbruch, M., & Davis, J. (2020). A design-led conceptual framework for developing school integrated STEM programs: the Australian context. *International Journal of Technology and Design Education*, 1-29. https://doi.org/10.1007/s10798-020-09619-5

Appendix 1: Turkish Form of the TBaCCT Scale

Factor	English	*Turkish*	$\bar{x}$	sd
Programming self-efficacy	I can describe fundamental computing concepts (e.g., loops, variables, algorithms, conditional logic).	*Temel programlama kavramlarını (örneğin döngü, değişken, algoritma, mantıksal ifade) tanımlayabilirim.*	4.33	1.76
	I can correct mistakes in the coding of a computer program on my own.	*Bir bilgisayar programının kodlarındaki hataları düzeltebilirim.*	3.58	1.94
	I can suggest different solutions in order to solve coding problems.	*Programlama yaparken karşılaştığım kodlama problemleri için farklı çözümler geliştirebilirim.*	3.60	1.87
	I can look at a computer program and explain the purpose of each command.	*Bir bilgisayar programına bakıp her bir komutun amacını açıklayabilirim.*	3.63	1.88
	I am good at finding patterns in data.	*Bir veri setindeki örüntüleri bulabilirim. (Örüntü, tekrar eden veya belirli bir düzene sahip olan yapılardır. Örneğin haftanın günleri bir örüntüdür.)*	4.18	1.76
	I can apply Boolean logic (e.g., IF, AND, NOT, OR) to solve problems with multiple conditions.	*Birden fazla koşul içeren bir problemi çözmek için Boolean mantığını (örneğin EĞER, VE, DEĞİL, VEYA mantıksal operatörleri) uygulayabilirim.*	4.01	1.88
	I can read a formula (e.g., algorithm, equation, input/output process) and explain what it should do.	*Bir kod bloğunu (örneğin prosedür, fonksiyon) okuyup hangi görev için yazıldığını açıklayabilirim.*	3.76	1.89
	I can plan out the logic for a computer program even if I don't know the specific programming language.	*Bir programlama dilinden bağımsız olarak bir bilgisayar programını mantıksal olarak tasarlayabilirim.*	3.56	1.93
Teaching programming efficacy	I can explain basic computing concepts to children (e.g., algorithms, loops, conditionals, functions, variables, debugging, pattern-finding).	*Öğrencilere temel programlama kavramlarını (örneğin, algoritma, döngü koşullu ifade, fonksiyon, değişken, hata ayıklama, örüntü bulma) açıklayabilirim.*	3.94	1.93
	I can help students debug their computer programs.	*Öğrencilerin yazdıkları kodlardaki hatalarını ayıklamalarına yardımcı olabilirim.*	3.74	1.95
	I can find uses for computer programming that are relevant for students.	*Öğrenciler için uygun programlama etkinlikleri bulabilirim.*	3.92	1.86
	I can find uses for computer programming that are relevant for students.	*Öğrencilerin programlama becerilerine uygun programlama araçlarını seçebilirim.*	3.91	1.89
	I can integrate computer programming into my current curriculum.	*Programlamayı mevcut öğretim uygulamalarıma entegre edebilirim.*	3.85	1.86
	I know where to find the resources to help students learn to code.	*Öğrencilerin kod yazmayı öğrenmelerine yardımcı olacak kaynakları bulabilirim.*	4.01	1.89
	I believe that I have the requisite computer programming skills to integrate computing content into my class lessons.	*Programlamayı derslerime entegre etmek için gerekli programlama becerilerine sahibim.*	3.74	1.89
	I can recognize and appreciate computing concepts in all subject areas.	*Tüm branşlarda programlama kavramlarını kullanabilirim.*	3.51	1.85
	I can create computing activities at the appropriate level for my students.	*Öğrencilerimin seviyelerine uygun programlama aktiviteleri oluşturabilirim.*	3.76	1.86
	I can explain computing concepts well enough to be effective in teaching computing.	*Programlama kavramlarını, programlama öğretimini etkili kılacak kadar iyi açıklayabilirim.*	3.70	1.87
	I can explain how computing concepts are connected to daily life.	*Programlama kavramlarının günlük yaşamla nasıl bağlantılı olduğunu açıklayabilirim.*	3.91	1.87
	I can develop and plan effective computing lessons.	*Etkili programlama dersleri geliştirebilirim.*	3.64	1.85
	I can develop and plan effective computing lessons.	*Etkili programlama dersleri planlayabilirim.*	3.63	1.84
Value	Computing should be taught in elementary/primary school.	*Bilgisayar bilimi ilkokuldan itibaren öğretilmelidir.*	4.98	1.48
	Learning about computing can help elementary students become more engaged in school.	*Bilgisayar bilimini öğrenmek, öğrencilerin okulla daha fazla ilgilenmelerine yardımcı olabilir.*	4.72	1.48

Factor	English	*Turkish*	x̄	sd
	Computing content and principles CAN be understood by elementary school children.	*Bilgisayar bilimi içeriği ve ilkeleri ilkokul çocukları tarafından anlaşılabilir.*	4.62	1.45
	My current teaching situation does NOT lend itself to teaching computing concepts to my students.	*Mevcut öğretim sürecim, öğrencilerime bilgisayar bilimi kavramlarını öğretmeye uygundur.*	3.92	1.68
	Knowledge of computer programming is NOT needed in most careers.	*Kariyer yapmak için bilgisayar bilimi bilgisi gereklidir.*	4.82	1.38
	Knowledge of computer programming is NOT needed in most careers.	*Çoğu meslekte bilgisayar bilimi bilgisi gereklidir.*	4.97	1.32
	Providing more computing activities is NOT necessary to enrich my students' overall learning.	*Öğrencilerimin öğrenme süreçlerini zenginleştirmek için daha fazla bilgisayar bilimi etkinliği sağlamalıyım.*	4.83	1.30
	Computing is an important 21st-century literacy.	*Bilgi işlemsel düşünme önemli bir 21. yüzyıl okuryazarlığıdır.*	5.22	1.21
	Computational thinking is an important part of today's science standards.	*Bilgi işlemsel düşünme günümüz bilim standartlarının önemli bir parçasıdır.*	5.19	1.19
	My current students are going to need to know how to code to remain competitive for jobs by the time they are adults.	*Öğrencilerimin geleceğin iş dünyasında rekabet edebilmeleri için nasıl kod yazılacağını bilmeleri gerekir.*	4.95	1.28
	Computing is NOT something that should be taught to special needs students.	*Bilgisayar bilimi, özel ihtiyaçları olan öğrencilere öğretilmelidir.*	4.58	1.44
Computational thinking self-efficacy	When I'm presented with a problem, I have difficulty breaking it down into smaller steps.	*Bir problemi, çözülebilir daha küçük problemlere bölebilirim.*	4.80	1.40
	I struggle to generalize solutions that can be applied to many different problems.	*Bir çözümü, birçok farklı probleme uygulanabilecek biçimde genelleştirebilirim.*	4.64	1.46
	I am NOT good at solving puzzles.	*Bulmaca çözmekte iyiyimdir.*	4.81	1.19
	I struggle to identify where and how to use variables in the solution of a problem.	*Bir problemin çözümünde değişkenlerin nerede ve nasıl kullanılacağını belirleyebilirim.*	4.78	1.34
	-	*Bir problemin çözümünde verinin veya olayların ortak özelliklerini tanımlayabilirim.*	4.82	1.27
	-	*Bir problemin çözümünü adımlara bölerek bu adımları mantıklı bir şekilde sıralayabilirim.*	4.85	1.29
	-	*Bir problemin çözümü için gereksinimleri belirleyebilirim.*	4.82	1.30

* Items 5, 6, and 7 in the computational thinking self-efficacy factor were added to the scale.