IJETS@R

(ISSN: 2587-0238)

Haksever, Ü.G. & Taşpınar Şener, Z. & Kert, S.B. (2024). Development Of Computatıonal Thınkıng Skılls Through Mathematıcal Modellıng: A Self-Effıcacy Perspectıve, *International Journal of Education Technology and Scientific Researches, 9(28), 603-618.*
DOI: http://dx.doi.org/10.35826/ijetsar.750

Article Type (Makale Türü): Research Article

DEVELOPMENT OF COMPUTATIONAL THINKING SKILLS THROUGH MATHEMATICAL MODELLING: A SELF-EFFICACY PERSPECTIVE

Ümmi Gülsün HAKSEVER
Yıldız Tecnichal University, Istanbul, Turkiye, ugulsumhaksever@gmail.com
ORCID: 0000-0001-8641-4594

Zehra TAŞPINAR ŞENER
Assoc. Prof., Yıldız Tecnichal University, Istanbul, Turkiye, taspinar@yildiz.edu.tr
ORCID: 0000-0001-8914-784X

Serhat Bahadır KERT
Prof. Dr., Yıldız Tecnichal University, Istanbul, Turkiye, sbkert@yildiz.edu.tr
ORCID: 0000-0002-5534-8026

Received: 05.04.2024 *Accepted: 25.08.2024* *Published: 15.09.2024*

ABSTRACT

Computational thinking is of great importance in developing students' ability to analyse complex problems in a systematic way, to develop appropriate strategies for solving these problems, and to use technology effectively in applying these solutions. In this context, mathematical modelling activities, which involve solving complex real-life situations through mathematics, stand out as an effective method for developing these skills in students. The purpose of this study is to investigate the effect of mathematical modelling activities on students' computational thinking (CT) self-efficacy. The participants of the study were English preparatory students at a high academic level public school in Istanbul. The modelling activities used in the study were restructured by considering the steps of computational thinking skills. Among these steps, the Python software language was used in the 'algorithmic thinking' step, and the students translated their problem-solving systems into software language through Python. As a data collection tool, in addition to the activities, the CT self-efficacy scale was used as a pre- and post-test. The mathematical modelling activities were applied to the high school preparatory class students, who were the research group, for a period, and the change in the students' CT self-efficacy was evaluated in the process. The results showed that modelling activities led to a significant increase in students' CT self-efficacy. It was found that students felt more competent in sub-dimensions such as problem solving, analytical thinking and algorithmic thinking. These findings suggest that mathematical modelling activities are a powerful tool for developing computational thinking skills. In conclusion, this study suggests that mathematical modelling activities should be more widely incorporated into the secondary school curriculum to develop computational thinking skills and draws attention to the importance of professional support for teachers in this process.

Keywords: Computational thinking skills, mathematical modelling, self-efficacy.

INTRODUCTION

The 21st century is characterized by the rapid spread of digital technologies, easier access to information and increased global connectivity. To keep up with the demands of this era, individuals need to develop certain skills, known as 21st-century skills, to be effective in a technology-enabled world. The Organization for Economic Co-operation and Development [OECD] (2018) defines 21st century skills as the ability of individuals to solve complex problems, think creatively, collaborate, and communicate effectively using digital technologies.

These skills enable individuals to overcome the difficulties they face not only in the business world but also in their personal lives. The development of these skills contributes to individuals being more active and participatory in society. The Ministry of National Education (MEB, 2018), aims to equip students with competencies such as digital literacy, critical thinking, creative problem solving and collaboration by placing 21st century skills at the heart of the education curriculum. These competencies are recognized as essential skills for students to be successful in their future social and professional lives.

In this context, computational thinking stands out as a critical 21st century skill required in the digital age. Computational thinking refers to the ability to use computer science concepts such as algorithmic thinking, modelling and abstraction to solve complex problems that individuals face in the digital world (Grover & Pea, 2013; Wing, 2006). This skill enhances individuals' ability to understand, use and develop technology and supports their ability to adapt to ever-changing technological innovations (Grover & Pea, 2013)

However, effective tools and methods are needed to develop computational thinking skills. Research into how activities and strategies that can be used in education can support these skills is of great importance.

Computational thinking

This concept, first proposed by Papert, (1980), was defined in a broader framework by Wing (2006). Wing (2006) defines computational thinking as the process of solving problems, designing systems and understanding human behaviour by focusing on the fundamental concepts of computer science. This skill involves individuals solving problems through abstraction, creating algorithms and systematic thinking. According to Wing, computational thinking is a critical skill not only for programmers, but for all individuals who can exist in the digital age.

As technology has become more effective in education, as it has in other sectors, computational thinking skills have become increasingly important and have been defined by a number of researchers, but there is no generally accepted definition (Grover & Pea, 2013; Resnick et al., 2009). The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA), together with expert researchers in the field, have undertaken project work to develop a definition for use in schools (ISTE, 2011b; Lye & Koh, 2014). ISTE and CSTA also define computational thinking as the process of solving problems using computers or other digital tools (ISTE, 2011a) This definition emphasizes that computational thinking is not only a technical skill, but also an approach to problem solving. This process involves steps such as formulating problems, organizing and analysing data logically, developing algorithms and generalizing solutions. In this respect, computational thinking can contribute to solving not only technology-related problems, but also complex problems in other areas of life.

The development of the concept of computational thinking has followed a parallel course with the increasing place of computer science in education. Since the 1980s, the rapid spread of computing technologies has led to computational thinking becoming increasingly important in all areas of education (Grover & Pea, 2013). In the process, it has been recognized that computational thinking is not limited to the fields of technology and engineering, but is also an important tool in the sciences, social sciences and the arts (Lye & Koh, 2014). In STEM (Science, Technology, Engineering, and Mathematics) education, the teaching of computational thinking has increasingly found a place in curricula to improve students' problem-solving skills and equip them with the skills required in the 21st century (Barr & Stephenson, 2011).

When examining the definitions related to computational thinking skills, it is found that computational thinking skills have multiple components and different components have been proposed in each different definition (Yağcı, 2018). As a result of a study conducted by ISTE (2015a) together with CSTA, it was found that problem solving, algorithmic thinking, critical thinking, collaborative learning, divergent thinking and communication skills are the most used skills of computational thinking skills.

Problem solving

Problem solving is one of the most fundamental dimensions of computational thinking. Students analyse the complex problems they encounter, break them down into sub-problems and develop algorithmic methods to solve these sub-problems (Wing, 2006). The problem-solving process typically involves cognitive processes such as algorithm development, data organization and analysis. This dimension plays an important role in developing students' critical thinking and systematic problem-solving skills (Brennan & Resnick, 2012).

Algorithmic thinking

Algorithmic thinking involves the development of step-by-step sequences of instructions for solving problems and is one of the central components of computational thinking (Grover & Pea, 2013). This way of thinking allows individuals to develop systematic and logical solutions by breaking down complex problems into smaller, more manageable components (Wing, 2010). In this respect, algorithmic thinking stands out as an indispensable skill both in interdisciplinary problem-solving processes and in the context of lifelong learning.

Collaborative learning and critical thinking

Computational thinking involves collaborative learning and critical thinking skills that go beyond individual learning processes. Students' ability to evaluate and integrate different perspectives during group work is an important dimension of ICT (Kafai & Burke, 2014). In this process, students can approach problems from a broader perspective and produce creative solutions by working together.

Differently thinking

Divergent thinking involves students generating creative and innovative solutions using existing knowledge and skills. This dimension helps students to develop flexibility and adaptability skills in computational thinking processes (Lye & Koh, 2014). Divergent thinking provides students with creative ways of solving problems beyond the ordinary and encourages them to discover new approaches in the process.

Teaching computational thinking

As the place of computational thinking in education grows, so does the question of how to teach this skill. Educators and researchers have developed different strategies for teaching computational thinking to students. Weinberg (2013) suggests four basic ways to teach this skill: these methods, shown in Figure 1, are computer science without computers, programming tools, game and robot programming, and interdisciplinary programming methods (Kalelioğlu & Keskinkılıç, 2018)

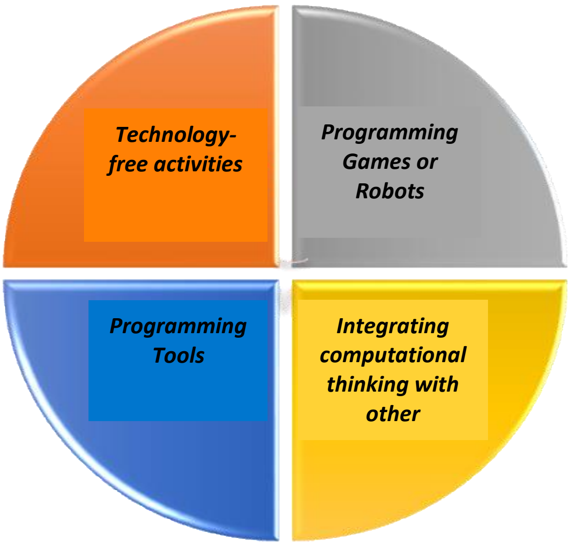

Figure 1. Teaching computational thinking skills by Weinberg (2013)

Technology-free activities

Technology-free activities are activities that do not use computers or digital devices to teach the basic concepts of computer science (Gülbahar & Kalelioğlu, 2017). This method is particularly suitable for children at the primary school level (6-10 years old). Students in this age group can learn how computers work, how algorithms work, and how data is organized by working with concrete materials (Bell et al., 2005) For example, algorithmic logic and problem-solving processes can be taught through simple card games, puzzles and physical activities. Such activities demonstrate that students can develop computational thinking skills without computers.

Programming Games or Robots

Programming Games or Robots allows students to develop their computational thinking skills in a fun and interactive way (Grover & Pea, 2013). This method is generally suitable for middle school (11-14 years) and high school (15-18 years) students (Lye & Koh, 2014). Game programming allows students to develop algorithms, problem solving and creative thinking skills (Resnick, 2007). Platforms such as Scratch or Alice make it easier for students in this age group to design and code games (Kafai & Burke, 2015). Robot programming involves students programming physical robots to solve real-world problems. Tools such as LEGO Mindstorms or VEX Robotics allow students to practice engineering, mathematical and computational thinking skills (Sullivan & Bers, 2016)

Programming Tools

One of the most widely used methods to provide students with computational thinking skills is programming instruction (Kalelioğlu & Keskinkılıç, 2018). This method is divided into two, block-based and text-based programming, depending on the level of the students. While block-based programming is generally used at the secondary school level, text-based programming is taught at the high school level. Block-based programming allows students to embody programming concepts in a visual and interactive environment through platforms such as Scratch (Resnick et al., 2009; Weintrop et al., 2016) Text-based programming involves writing code in languages such as Python, Java and C++ and requires more complex programming skills (Lye & Koh, 2014; Grover & Pea, 2013).

Integrating computational thinking with other disciplines

Integrating computational thinking with other disciplines allows students to use this skill in areas such as mathematics, science, social sciences, and even the arts (Grover & Pea, 2013; Kalelioğlu & Keskinkılıç, 2018). Interdisciplinary integration allows students to apply computational thinking to real-world problems. For example, data analysis using algorithms can be carried out in a math course, modelling and simulation techniques can be taught in a science course, or studies on data collection and analysis can be carried out in a social science course. This approach shows students that computational thinking can be used in a wide range of areas and how it can be applied in different fields.

As an example of teaching interdisciplinary computational thinking skills, mathematical modelling is an effective method that enables students to both understand mathematical concepts and apply these concepts to real-world problems (Doerr & English, 2006) Mathematical modelling stands out as an effective tool in this context (Blum & Nish, 1991). Mathematical modelling enables students to deal with real-world problems and improves their ability to apply mathematical concepts to solve these problems (Ferri, 2013) This process enriches students' learning experiences by applying abstract mathematical concepts to concrete situations and makes mathematical thinking meaningful in the context of everyday life (Lesh et al., 2003) Therefore, this study investigated the effects of mathematical modelling activities on computational thinking.

There are also different studies on how to assess computational thinking in educational settings. A general framework for the assessment of ICT is presented below.

Assessing computational thinking

Given the multiple dimensions and comprehensive nature of these skills, the assessment of computational thinking skills requires a rigorous and careful approach. Various instruments and scales have been developed in the literature to assess these skills.

The assessment of computational thinking skills has emerged as a rich area of research, both theoretically and practically. Brennan and Resnick (2012) provided a basis for the assessment of these skills with their 'Computational Thinking Framework' and identified three main components of computational thinking: problem solving, data representation and algorithmic thinking. This framework allows for a structured examination of computational thinking in educational settings. Grover and Pea (2013) provided a comprehensive review of research at the K-12 level and found that the most used methods for assessing computational thinking are generally problem-solving oriented. This is an important step in understanding and developing students' computational thinking processes. Shute, Sun, and Asbell-Clarke (2017) discussed the effectiveness of scales used to assess computational thinking and raised important questions about whether these scales accurately reflect students' cognitive processes. Their study highlighted that computational thinking skills should be addressed in a multidimensional way, rather than from a single perspective. In this context, the assessment of computational thinking skills stands out as an indispensable tool for understanding both student performance and the effectiveness of educational programs.

Self-efficacy refers to an individual's belief in his or her ability to successfully complete a specific task, and these beliefs have a significant impact on an individual's motivation, persistence, and academic achievement (Bandura, 1997). In complex cognitive processes such as computational thinking skills, students' self-efficacy perceptions play an important role in the development and application of these skills (Gülbahar et al., 2019) In this context, the Self-Efficacy Scale for Computational Thinking Skills developed by Yağcı (2018) is a valid and reliable tool designed to assess students' self-efficacy perceptions towards these skills. Yağcı's study aimed to measure students' self-efficacy perceptions that influence their participation in computational thinking processes and their success in these processes and made an important contribution to assessments in this area (Yağcı, 2018).

In teaching computational thinking, interdisciplinary approaches are used as an effective way to develop and integrate students' skills in different domains. In this context, mathematical modelling is a powerful tool that enables students both to understand mathematical concepts and to apply these concepts to real-world problems. While mathematical modelling allows students to develop computational thinking skills such as algorithmic thinking, data analysis and problem solving, it also strengthens their sense of self-efficacy in dealing with the difficulties they encounter in the process (Blum & Niss, 1991) Students' ability to solve the problems they encounter in mathematical modelling processes increases their confidence in their computational thinking skills, which in turn makes their learning processes more effective.

Self-efficacy reflects individuals' beliefs about their capacity to successfully complete a specific task and these beliefs play a critical role in learning processes (Bandura, 1997) When mathematical modelling provides students with the opportunity to solve complex problems and apply computational thinking skills in this process, students' confidence in these skills increases. Such interdisciplinary work not only provides students with technical skills, but also reinforces their self-confidence in the learning process. As a result, the use of mathematical modelling in teaching computational thinking supports both cognitive and affective development of students and contributes significantly to their self-efficacy.

Mathematical modelling

Mathematical modelling is defined as the process of mathematical representation and solution of real-world problems (Erbaş et al., 2014) This process involves representing a problem with a mathematical model, developing solutions through analysis of the model, and interpreting the results obtained (Blum, 2011) Mathematical modelling deepens students' mathematical thinking skills by allowing them to apply abstract mathematical concepts in concrete contexts. This process also supports higher order cognitive skills such as critical thinking, problem solving and creative thinking (Ferri, 2006).

Since the mid-twentieth century, mathematical modelling has been increasingly emphasized in educational research and practice, along with the growing importance of mathematical problem solving in education (Blum & Niss, 1991). Blum (2011) states that modelling has become central to mathematics education and an important tool for solving interdisciplinary problems. Ferri (2006) argues that mathematical modelling is a teachable and learnable process and stresses that this process can be taught to students through structured pedagogical approaches. In this context, modelling is seen as a process that is not only limited to the transfer of mathematical knowledge, but also deepens students' mathematical understanding and develops their creative problem-solving skills. In Turkey, the development of mathematical modelling has been supported by an increasing number of academic studies and research, especially in recent years. Aztekin and Taşpınar Şener (2015), in their meta-synthesis study of mathematical modelling research in Turkey, found that modelling is becoming increasingly important in mathematics education. According to this study, modelling is used as an effective tool in students' understanding and application of mathematical concepts, and the integration of mathematical modelling in educational programs in Turkey is considered an important step in providing students with problem-solving and critical thinking skills.

Effective teaching of mathematical modelling requires pedagogically rich and structured approaches aimed at providing students with conceptual depth and improving their problem-solving skills (Lesh & Doerr, 2003; Blum & Ferri, 2009). Erbaş et al. (2014) emphasize that different methods can be used to teach mathematical modelling, and pedagogical strategies should be developed to ensure students' active participation in this process. In addition to deepening students' mathematical thinking skills, modelling instruction significantly enhances their interdisciplinary problem-solving abilities (Blum & Leiß, 2007; Lesh et al., 2003). In Turkey, to effectively teach mathematical modelling, teachers should be equipped with the necessary knowledge and skills and the curriculum should be organized accordingly (Aztekin & Taşpınar Şener, 2015)

This study investigates the effect of mathematical modelling activities on the computational thinking self-efficacy of high school preparatory students. In the study, a pre-test and a post-test were administered to determine the students' level of computational thinking self-efficacy. Our research question is as follows:

What is the effect of mathematical modelling activities on the computational thinking self-efficacy of high school preparatory class students?

- H_0 = There is no significant difference between the students' pretest and post-test scores on the computational thinking self-efficacy scale.
- H_1 = There is a significant difference between students' pretest and post-test scores on the computational thinking self-efficacy scale.

METHOD

Research model

In this study, a one-group pretest-post-test quasi-experimental design was used to examine mathematical modelling and computational thinking self-efficacy in high school preparatory students. The one-group pretest-post-test model is an experimental arrangement in which measurements are taken before and after an intervention on a group to test the effect of an independent variable. In this model, the effect of the independent variable on the dependent variable is examined by comparing the measurements taken before (pretest) and after (post-test) the intervention (Yıldırım & Simsek, 2021). In the study, mathematical modelling activities were supported using the Python programming language. Students aimed to solve real-world problems by creating mathematical models, using Python as a tool. For example, algorithms were developed for specific problems, these algorithms were coded in Python and the results were simulated. Students developed problem solving skills by developing different strategies in the modelling process and were involved in algorithmic thinking processes. In this way, the students' development in both computational thinking and programming skills was observed. At the beginning of the intervention, students' self-efficacy in computational thinking skills was measured. After this assessment, a study with mathematical modelling applications was conducted for 6 weeks. At the end of the study, students' self-efficacy in computational thinking skills was measured again as a post-test and the development of their self-efficacy was examined.

Working Group

This study was conducted with 29 students who were studying in the preparatory class of an Anatolian high school with a preparatory class in Fatih district, Istanbul province. The study used the criterion sampling method, which is one of the purposive sampling methods based on certain criteria or criteria defined by Yıldırım and Şimşek (2021). The selection criteria were that the students should know the Python programming language and have basic knowledge of mathematics.

Data Collection Tools

In this study, the "Computational Thinking Self-Efficacy Scale" developed by Yağcı (2018) was used to collect pre-test and post-test data. This scale was developed to measure students' self-efficacy perceptions of computational thinking skills and includes sub-dimensions such as algorithmic thinking, problem solving, divergent thinking and collaborative working. The scale was administered at the beginning and at the end of the study to assess the impact of the mathematical modelling process on students' computational thinking skills. In this way, the change and development of the students' self-efficacy perceptions were comparatively analysed.

To assess students' computational thinking skills, the four-factor, five-point Likert-type 'Computational Thinking Skill Scale' developed by Yağcı (2018) consisting of 42 items was used. After factor analysis and reliability analysis, 11 items were removed from the original scale, which originally contained 53 items. The scale includes the following sub-dimensions: Problem Solving, Cooperative Learning and Critical Thinking, Divergent Thinking and Algorithmic Thinking. The internal consistency coefficients for the sub-dimensions of the scale were .962 for Problem Solving, .937 for Cooperative Learning and Critical Thinking, .937 for Divergent Thinking and .828 for Algorithmic Thinking. The calculated internal consistency coefficient (Cronbach Alpha) for the entire scale was .969. The items in the scale were scored as (1) strongly disagree, (2) disagree, (3) undecided, (4) agree, and (5) strongly agree.

FINDINGS

This section presents the descriptive statistical results obtained from the pre- and post-test data of the students using the Computational Thinking Self-Efficacy Scale. To compare the self-efficacy scores, it was first examined whether the pre-test and post-test scores were normally distributed. Coefficients of skewness and kurtosis, measures of central tendency and the Shapiro-Wilk normality test were used to assess the normality of the data. In cases where the sample size is less than 50, the Shapiro-Wilk test is one of the preferred methods for assessing normality (Büyüköztürk, 2020; Demir, 2020). Since our sample group was less than 50, the Shapiro-Wilk normality test was applied to determine whether the data were suitable for normal distribution. The data obtained are presented in Table 1.

Table 1. Pre-test normality distributions of data collection tools

					Shapiro-Wilk	
	$\bar{X}$	Ss	Distortion	Kurtosis	Statistics	p
Problem solving	54.379	1.10129	.154	.35	.981	.874
Critical thinking and cooperative learning	20.206	.55126	1,244	2.217	.935	.075
Differently thinking	20.517	.72196	-1,489	0.078	.965	.431
Algorithmic thinking	14.275	.35406	.396	.216	.961	.354
General Total Pre-test	109.37	1.63280	.730	-.071	.943	.117

In the Shapiro-Wilk test, a significance value (p) greater than .05 indicates that the data obtained has a normal distribution and a value less than .05 indicates that it has a non-normal distribution (Demir, 2020). Looking at Table 2, according to the data from the pre-test of self-efficacy, the overall test score and the sub scores have a normal distribution since p> .05. This evaluation was done in the same way for the self-efficacy post-test data and is shown in Table 2.

Table 2. Post-test normality distributions of data collection tools

					Shapiro-Wilk	
	$\bar{X}$	Ss	Distortion	Kurtosis	Statistics	p
Problem solving	73.0690	1.81668	-2.396	1.183	.933	.066
Critical thinking and cooperative learning	29.6207	1.10353	-1.126	-.448	.960	.320
Differently thinking	29.2414	.81497	.025	-.977	.921	.032
Algorithmic thinking	17.5862	.46162	-1.686	1.178	.949	.168
General Total Post-test	149.51	3.33906	-.082	-.473	.981	.853

When analysing Table 2, looking at the post-test results for self-efficacy, the sub-items other than General and Divergent Thinking have p>.05. This result shows that they have a normal distribution. However, as p=0.32 and p<.05 for the divergent thinking subheading, the skewness and kurtosis values were examined. Another method used for normal distribution is the skewness and kurtosis values. If the values of skewness and kurtosis are between -1 and +1, the data have a normal distribution (Büyüköztürk, 2020). If we look at the skewness and kurtosis values of the different thinking sub-headings, we can see that they are between +1 and -1 values and show a normal distribution.

The dependent t-test, one of the parametric tests, was used in the analysis of the data due to the normal distribution of the pre- and post-tests of computational thinking self-efficacy. The dependent t-test is a parametric test used to determine whether the mean difference between two dependent data sets in the same sample is statistically significant. This test is usually used to analyse data obtained before and after measurement. For the dependent t-test to be applicable, the data must be assumed to be normally distributed.

The results of the dependent t-test are given in Table 3 below.

Table 3. Dependent sample t test results of CT pre-test and post-test scores

			t-test		
	$\bar{X}$	Ss	t	Sd	p
Problem Solving Pre-test	54.3783	1.10129			
Problem Solving post-test	73.0690	1.81668	8.699	28	<.001
Critical Thinking and Cooperative Learning pre-test	20.2069	.55126	7.839	28	<.001
Critical Thinking and Cooperative Learning post-test	29.6207	1.10353			
Thinking Differently pretest	20.5172	.434			

Thinking Differently post-test	29.2414	.81497	7.409	28	<.001
Algorithmic Thinking pretest	14.2759	.35406			
Algorithmic Thinking post-test	17.5862	.46162	5.322	28	<.001
General pre-test	109.37	8.79			
General post-test	149.51	17.98	10.311	28	.000

The results are considered statistically significant, and it is concluded that the difference between the two measurements to which the dependent t-test is applied is significant if $p \leq 0.05$. When Table 3 is analysed, it is seen that $p<0.05$ in the overall and sub-headings of the pre-test and post-test. This result shows that there is a significant difference between the pre-test and post-test. This reveals that the applications made after the pre-test positively affected the students' self-efficacy in computational thinking skills.

Examining the subdimension results of the dependent samples t-test, the post-test means of all subdimensions increased compared to the pretest means. In the sub-dimension of problem solving, the post-test means ($\bar{X}$ = 73.0690) was significantly higher than the pre-test means ($\bar{X}$ = 54.3783); t (28) = 8.699, p < .001. This shows that mathematical modelling activities significantly improved students' problem-solving skills. In the sub-dimension of critical thinking and cooperative learning, the post-test means ($\bar{X}$ = 29.6207) was significantly higher than the pre-test means ($\bar{X}$ = 20.2069); t (28) = 7.839, p < .001. This result shows that the implementation also made a significant contribution to the students in this area. In the Thinking Differently sub-dimension, the post-test means ($\bar{X}$ = 29.2414) is significantly higher than the pre-test means ($\bar{X}$ = 20.5172); t (28) = 7.409, p < .001. This shows a significant improvement in the students' divergent thinking skills. In the Algorithmic Thinking sub-dimension, the post-test means ($\bar{X}$ = 17.5862) was significantly higher than the pre-test means ($\bar{X}$ = 14.2759); t (28) = 5.322, p < .001. This result shows that there is a significant improvement in the students' algorithmic thinking skills. In the overall BIDB scores, the post-test means ($\bar{X}$ = 149.51) was significantly higher than the pre-test means ($\bar{X}$ = 109.37); t (28) = 10.311, p = .000. Analysing the t-values between the sub-dimensions, the highest increase was in problem solving (t (28) = 8.699) and the lowest increase was in algorithmic thinking (t (28) = 5.322).

The dependent sample t-test results presented in Table 4 show that there are significant improvements in students' computational thinking skills at both the overall and sub-dimension levels. A p-value < .001 for each sub-dimension indicates that these differences are statistically significant. These results suggest that the training or intervention implemented significantly benefited students in areas such as problem solving, critical thinking, collaborative learning, divergent thinking and algorithmic thinking.

Table 4. Descriptive statistics results for students' CT scores

	N	The Smallest	The biggest	$\bar{X}$ (Pre-test)	$\bar{X}$ (post-test)
Problem solving	29	1.00	5.00	2.71	3.65
Critical thinking and collaborative learning	29	1.00	5.00	2.52	3.70
Differently thinking	29	1.00	5.00	2.27	3.24
Algorithmic thinking	29	1.00	5.00	2.85	3.51
General	29	1.00	5.00	2.60	3.55

Table 4 shows in detail the pre-test and post-test results of the mean scores of the CT scale (Determination of Knowledge Level). Looking at the table, the mean scores of the students in the post-test are higher than the mean scores in the pre-test. This shows that the students' level of knowledge has increased.

In the evaluation of the scores obtained from the CT scale, the scale scores were analysed by dividing them into three main knowledge levels. In this analysis, the difference (5-1) between the highest and lowest scores of the 5-point Likert-type scale was divided into three equal intervals. According to this calculation

‘Low level' was defined as scores between 1 and 2.33.
‘Medium level' was defined as scores between 2.34 and 3.67
‘Advanced level' was defined as scores between 3.68 and 5.00

According to these scoring ranges, the average score of the students in the sub-dimension Critical Thinking and Cooperative Learning ($\overline{X}$ =3.70) reached the knowledge level of 'Advanced Level'. This shows that the students have a very high level of knowledge in this sub-dimension.

When other sub-dimensions are analysed:

- The average score in the sub-dimension problem solving ($\overline{X}$ =3.65) is in the category 'moderate level'. This means that the students have a moderate level of knowledge in problem solving skills.
- The mean score ($\overline{X}$ =3.24) in the Divergent Thinking sub-dimension is also in the 'Moderate Level' category, which means that the students have a moderate level of knowledge and skills in this sub-dimension.
- In the sub-dimension of algorithmic thinking, the mean score ($\overline{X}$ =3.51) is also in the Intermediate level category, indicating that the students' skills in this sub-dimension are at an intermediate level.

The overall mean score ($\overline{X}$ =3.55) is in the 'medium' category, indicating that the students' general knowledge is also at a medium level. This situation shows that there is a positive development in the students' knowledge level in general, but this development remains at the 'medium level' in the sub-dimensions where it is not at an advanced level.

CONCLUSION and DISCUSSION

This study investigated the effect of modelling activities designed based on computational thinking skills on students' computational thinking self-efficacy. When the students who participated in the study were assessed for their ITS self-efficacy before the application, it can be said that they considered themselves to be moderately self-efficient in all sub-dimensions. Yağcı (2018), in his study examining the ITS self-efficacy of students in Anatolian and science high schools, found that the students considered themselves moderately competent in terms of computational thinking. This finding parallels the results of the current study. However, when the subdimensions were examined in Yağcı's (2018) study, it was found that students considered themselves highly competent in terms of problem solving and algorithmic thinking skills. Similarly, in İbili, Günbatar, and Sırakaya's

(2020) study, it was found that 50% of vocational high school students considered themselves at a high level in the problem-solving dimension when looking at the subdimensions. Therefore, the self-efficacy scores of the participants in this study are lower than in other studies. Although the participants in this study knew the Python programming language and had a high academic level, it is unexpected that their self-efficacy scores were lower. Therefore, there may be a mismatch between students' knowledge and self-efficacy. In other words, although they are academically successful and have programming knowledge, they may not feel sufficiently equipped to apply this knowledge to different problems. In addition, the fact that the participants were preparatory year students and had not yet started advanced courses may have contributed to this result. In this context, further research with a larger sample at different grade levels is needed.

As a result of the application, it was found that students' computational thinking self-efficacy improved with modelling activities. When all sub-dimensions are considered, these activities strengthen students' self-efficacy for problem solving, critical thinking, divergent thinking and algorithmic thinking skills. Although it is theoretically stated in the literature that the mathematical modelling process is related to the computational thinking process, no experimental study has been found that modelling activities improve students' computational thinking skills. In this respect, this study makes an important contribution to the literature as one of the few studies to experimentally investigate the effect of mathematical modelling activities on students' self-efficacy in computational thinking skills. The results show that these activities strengthen computational thinking self-efficacy and support the relationship proposed in the theoretical framework with experimental data. The development of computational thinking self-efficacy enables students to approach the problem-solving process in a more active and confident manner. This is consistent with Bandura's (1997) concept of self-efficacy within social cognitive theory; when individuals believe that they can achieve success in a particular task, this belief increases their performance. Particularly in the educational context, the positive relationship between self-efficacy and academic achievement has been confirmed in many studies (Zimmerman, 2000; Schunk & Pajares, 2009). The findings of this study support this literature; the students' experiences and achievements during the process increased their expectations of success in computational thinking tasks and strengthened their beliefs about achieving more complex tasks. In this regard, there is a need for more comprehensive studies that examine not only the effects of modelling activities on students' perceptions of self-efficacy, but also their contributions to the development of computational thinking skills. Comprehensive studies to be conducted at different educational levels on the level and development of the sub-dimensions of computational thinking skills (algorithmic thinking, problem solving, critical thinking, etc.) will further deepen the literature in this area. Such studies will help to analyse the needs of different groups of students for computational thinking skills.

SUGGESTIONS

In line with the findings of this study, it is recommended that more mathematical modelling activities be included in the high school curriculum to develop computational thinking skills. These activities will both deepen students' mathematical understanding and develop their computational thinking skills. In addition, professional development programs are needed for teachers to effectively teach computational thinking and mathematical modelling skills. These programs should aim to teach teachers how to integrate modelling processes into the classroom environment and how to guide students to develop these skills. It is also important to promote student-centred learning methods to support the development of computational thinking skills; in this context, preference should be given to activities that actively engage students and allow them to be directly involved in problem-solving processes. Given the role of technology in education, it is recommended to use computer-based modelling software and simulation tools in computational thinking and mathematical modelling activities. Such tools can help students to better understand abstract mathematical concepts by making them concrete.

For future research, it is important to conduct such studies across different educational levels and demographic groups to increase the generalizability of the findings. Similarly, research into the long-term effects of modelling activities can help us to understand the effects on the development of computational thinking skills more fully. In addition, a comparison of different pedagogical approaches to teaching computational thinking and mathematical modelling would be valuable in identifying the most effective methods. Finally, the role of interdisciplinary approaches in the development of computational thinking skills should be investigated and the integration of mathematical modelling activities with other disciplines such as science, engineering and computer science should be explored.

ETHICAL TEXT

This article has followed the journal writing rules, the publication principles, the research and publication ethics rules, and the journal ethics rules. The authors are responsible for any violations that may occur in relation to the article.

Ethical approval was granted by Yıldız Technical University Institute of Science and Technology Ethics Committee with the decision dated 03.05.2023 and number 2023.05.

In this study, the contribution rate of the first author is 35%, the contribution rate of the second author is 35% and the contribution rate of the last author is 30%.

REFERENCES

Aztekin, S., & Şener, Z. T. (2015). The Content Analysis of Mathematical Modelling Studies in Turkey: A Metasynthesis Study. *Education & Science/Egitim ve Bilim*, *40*(178). Bandura, A. (1997). *Self-efficacy: The exercise of control* (Vol. 604).

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM inroads*, *2*(1), 48-54.

Bell, T., Witten, I. H., Fellows, M., Adams, R., & McKenzie, J. (2005). *Computer Science Unplugged: An enrichment and extension programme for primary-aged children*.

Blum, W. (2011). Can modelling be taught and learnt? Some answers from empirical research. *Trends in teaching and learning of mathematical modelling: ICTMA14*, 15-30.

Blum, W., & Leiss, D. (2007). How do students and teachers deal with modelling problems. *Mathematical modelling (ICTMA 12): Education, engineering and economics*, 222-231.

Blum, W., & Niss, M. (1991). Applied mathematical problem solving, modelling, applications, and links to other subjects - State, trends and issues in mathematics instruction. *Educational Studies in Mathematics*, *22*(1), 37-68. https://doi.org/10.1007/BF00302716

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*(Vol. 1, p. 25).

Büyüköztürk, Ş. (2020). Sosyal Bilimler İçin Veri Analizi El Kitabı İstatistik, (28. Baskı). Ankara: Pegem Akademi.

Chen, P. P. (2003). Exploring the accuracy and predictability of the self-efficacy beliefs of seventh-grade mathematics students. *Learning and individual differences*, *14*(1), 77-90.

Demir, İ. (2020). SPSS ile istatistik rehberi. *Efe Akademi*.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. *Computers & Education*, *58*(1), 240-249.

Doerr, H. M., & English, L. D. (2006). Middle Grade Teachers' Learning through Students' Engagement with Modeling Tasks*. *Journal of Mathematics Teacher Education*, *9*(1), 5-32. https://doi.org/10.1007/s10857-006-9004-x

Erbas, A. K., Kertil, M., Çetinkaya, B., Cakiroglu, E., Alacaci, C., & Bas, S. (2014). Mathematical modeling in mathematics education: basic concepts and approaches. *Educational Sciences: Theory and Practice*, *14*(4), 1621-1627.

Ferri, R. B. (2013). mathematical modelling in european education. *Journal of Mathematics Education at Teachers College*, *4*(2). https://doi.org/https://doi.org/10.7916/jmetc.v4i2.624

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, *42*(1), 38-43.

Gülbahar, Y., & Kalelioğlu, F. (2017, November). Competencies of high school teachers and training needs for computer science education. In *Proceedings of the 6th Computer Science Education Research Conference* (pp. 26-31). https://doi.org/10.1145/3162087.3162092

Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısı ölçeği: Geçerlik ve güvenirlik çalışması. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *10*(1), 1-29. https://doi.org/10.16949/turkbilmat.385097

Ibılı, E., Günbatar, M. S., & Sırakaya, M. (2020). Bilgi-işlemsel düşünme becerilerinin incelenmesi: Meslek liseleri örneklemi. *Kastamonu Education Journal*, *28*(2), 1067-1078.

ISTE. (2011a). *Computational thinking leadership toolkit*.

ISTE. (2011b). *Computational_Thinking_Operational_Definition_ISTE*.

Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational psychologist*, *50*(4), 313-334.

Kalelioğlu, F., & Keskinkılıç, F. (2018). Bilgisayar bilimi eğitimi için öğretim yöntemleri. *Pegem Atıf İndeksi*, 155-182.

Lesh, R., Doerr, H. M., Carmona, G., & Hjalmarson, M. (2003). Beyond Constructivism. *Mathematical Thinking and Learning*, *5*(2-3), 211-233. https://doi.org/10.1080/10986065.2003.9680000

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

MEB. (2018), Milli Eğitim Bakanlığı Ortaöğretim Matematik Dersi (9, 10, 11 ve 12. Sınıflar) Öğretim Programı. Erişim adresi: http://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=343

Papert, S. (1980). Microworlds: Incubators for knowledge. *Mindstorms-Children, Computers and Powerful Ideas*, 120-134.

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, 1-6.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch. *Communications of the ACM*, *52*(11), 60-67. https://doi.org/10.1145/1592761.1592779

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *26*, 3-20.

Wang, J., Zhang, Y., Hung, C. Y., Wang, Q., & Zheng, Y. (2022). Exploring the characteristics of an optimal design of non-programming plugged learning for developing primary school students' computational thinking in mathematics. *Educational Technology Research and Development*. https://doi.org/10.1007/s11423-022-10093-0

Weinberg, A. E. (2013). *Computational thinking: An investigation of the existing scholarship and research* (Doctoral dissertation, Colorado State University).

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

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

Yağcı, M. (2018). A study on computational thinking and high school students' computational thinking skill levels. *International Online Journal of Educational Sciences*, *10*(2), 81-96.

Yıldırım, A., & Simsek, H. (2021). *Sosyal bilimlerde nitel araştırma yöntemleri. 12 baskı*. *Seçkin Yayıncılık*