Informatics in Education, 2024, Vol. 23, No. 3, 571–592

DOI: 10.15388/infedu.2024.20

Relationships between Middle School Students' Digital Literacy Skills, Computer Programming Self-efficacy, and Computational Thinking Self-efficacy

Muhammed Murat GÜMÜŞ*, Volkan KUKUL, Özgen KORKMAZ
Amasya University, Amasya, Türkiye.
e-mail: muhammedmuratg@gmail.com, volkan.kukul@amasya.edu.tr, ozgenkorkmaz@gmail.com

Received: October 2023

Abstract. This study aims to explain the relationships between secondary school students' digital literacy, computer programming self-efficacy and computational thinking self-efficacy. The study group consists of 204 secondary school students. A relational survey model was used in the research method and three different data collection tools were used to collect data. The structural equation model was used in data analysis to reveal a model that explains and predicts the relationships between variables. According to the results of the research, it was determined that digital literacy of secondary school students affected their computer programming self-efficacy, digital literacy affected their computational thinking self-efficacy, and computer programming self-efficacy affected their computational thinking self-efficacy. It was also found that digital literacy skills have an indirect effect on secondary students' computational thinking self-efficacy on computational thinking self-efficacy.

Keywords: digital literacy, computer programming, computational thinking.

1. Introduction

As in today's education system, technology is widely used in every field. Students use digital technologies in their education and in their daily lives. For example, they use digital technologies to solve the problems they face in the environments they live in and fulfill their duties and responsibilities in the schools where they study, especially in technology-supported courses. From this perspective, it has been stated that students need to develop digital literacy skills to use digital technologies and computational thinking skills to solve the problems they may encounter in their lives (Jun *et al.*, 2014; Zapata-

* Corresponding author

Ros, 2015; Shute *et al.*, 2017; Akiba, 2022). In particular, studies show that students have difficulties in using, interpreting and organising digital materials (Mudra, 2020). In addition, the difficulty of finding the right information among the mass of information in the digital environment and accessing the digital world also reveals problems related to digital literacy (Ervianti *et al.*, 2023). In fact, teachers' difficulty in combining digital skills with pedagogical methods, especially during and after the pandemic, shows that there are problems with digital literacy (Sánchez-Cruzado *et al.*, 2021). Therefore, it was predicted that students should have computer programming skills (Jun *et al.*, 2014; Demarle-Meusel *et al.*, 2017; Menon *et al.*, 2020; Akiba, 2022) and digital literacy skills (Akpınar and Altun, 2014; George-Reyes *et al.*, 2021; Kılıç, 2022) to develop computational thinking skills. From this point of view, it is thought that the computational thinking self-efficacy, computer programming self-efficacy, and digital literacy of middle school students, which are discussed in the study, are important in terms of solving the problems they face in their lives and increasing their academic achievement. For this reason, it can be stated that it is important to update educational curricula along with technology.

As a result of the development of digital technologies, countries in different parts of the world have updated or proposed to update their curricula (Mannila *et al.*, 2014; Bocconi *et al.*, 2016; Dagienė *et al.*, 2022; Petrovica *et al.*, 2022; Çayak and Erol, 2023; Timotheou *et al.*, 2023; Fernández *et al.*, 2023). While some countries focus on students' digital skills, others have integrated programming into their curricula. Some countries have tried to develop computational thinking skills by integrating them into different disciplines (Bocconi *et al.*, 2016; Dagienė *et al.*, 2022). Considering recent studies, Jawawi *et al.* (2022) showed that programming with educational robots plays an important role in developing computational thinking skills. Menon *et al.* (2020) stated that it is possible to develop computational thinking skills by using digital literacy skills. As in European countries, some other countries have also made efforts to integrate computational thinking skills into their curricula. For example, Shah (2019) developed a curriculum in India to develop computational thinking skills beyond using digital literacy and computer programming. Falkner, Vivian, and Falkner (2018) developed a broad curriculum in Australia that covers computational thinking concepts using digital literacy skills. At the same time, projects such as "Computer Science for All" in the USA show that computational thinking skills are highly valued (Bocconi *et al.*, 2016). It is seen that their common point is that they aim to increase individuals' computational thinking skills. In this context, modeling the relationship between digital literacy skills, programming self- efficacy, and computational thinking self-efficacy is important for the correct design of education to improve computational thinking skills. In this context, it is thought that by revealing the interaction between digital literacy, programming skills and computational thinking skills, the difficulties that students experience while developing their digital skills will be better understood. Therefore, it is seen that digital literacy can affect programming skills and programming skills can affect digital literacy skills (Edstrand and Sjöberg, 2023; Kılıç, 2022; Nouri *et al.*, 2020). However, the point taken as a basis in this study is that digital literacy is a broader and more fundamental concept and will affect programming skills since it is a skill that can be possessed by everyone. In other words, in this study, the direction of interaction is from general skills to specific skills. At the same time, it is known that the role of programming skills is important in the development of computational thinking skills (Caballero-Gonzalez, 2022; Ekinci *et al.*, 2023; Uslu, 2018; Ramazanoğlu, 2021; Yildiz *et al.*, 2017). Therefore, since programming skills are a more general concept than computational thinking skills, the relationship was established in the direction of computational thinking skills from programming skills. From this point of view, this study aimed to reveal the relationship between middle school students' digital literacy, programming self-efficacy, and computational thinking self-efficacy.

2. Literature Review

2.1. *Computational Thinking*

Although computational thinking is generally known as a problem-solving approach, it is defined as the knowledge, skills, and attitudes necessary to formulate problems by combining technology and thinking, produce solutions to problems, and use them in system design (Wing, 2006; ISTE, 2015; Curzon, 2015; Özden, 2015; Korkmaz *et al.*, 2017; Turhan, 2023). In addition, Peracaula-Bosch *et al.* (2024) revisited the computational thinking skill and based it on identifying the problem, defining the problem, developing the algorithm to solve the problem and implementing the solution. It is known that the concepts of 21st-century skills such as creative thinking, critical thinking, algorithmic thinking, and collaborative thinking constitute the sub-dimensions of computational thinking (Brichacek, 2014; ISTE, 2015; Korkmaz *et al.*, 2015; Turhan, 2023). In addition, Wing (2008) stated that these skills are prerequisites for computational thinking. Computational thinking self-efficacy can be defined as students' beliefs and perceptions about their computational thinking self-efficacy skills. With the increase in digitalization, students are expected to have computational thinking skills that reflect 21st-century skills (Wing, 2006; Aho, 2012; Akçay and Çoklar, 2016). For this purpose, computational thinking skills can be developed by using mathematics, physics, biology, programming, and other subject areas (Benaklı, *et al.*, 2017; Rubinstein and Chor, 2014; Weintrop *et al.*, 2016; Hsu, *et al.*, 2018; Ekinci *et al.*, 2023). It has been suggested that the development of problem-solving skills through algorithmic thinking, especially in the field of programming, enables the development of computational thinking skills (Akpınar and Altun, 2014; Lye and Koh, 2014; Lawanto *et al.*, 2017; Sun, *et al.*, 2021; Zhang *et al.*, 2023; Dağ *et al.*, 2023). In addition, middle school students' computational thinking skills have positive effects on metacognitive processes, executive functions, and working memory (Castro *et al.*, 2023).

2.2. *Programming*

Programming, which plays an important role in developing computational thinking skills and is used to enable the acquisition of 21st-century skills, can be defined as the digital organization, processing, and execution of commands for solving problems (Arabacıoğlu, *et al.*, 2007). Computer programming self-efficacy can be expressed as students' beliefs and perceptions about programming skills. Students' computer programming self-efficacy plays an important role in the easy learning of programming (Tsai, 2019). In addition, computer programming self-efficacy is an important factor in having information about students' achievements (Aşkar and Davenport, 2009). On the other hand, it has been determined in many studies that programming education influences 21st-century skills such as problem-solving and critical thinking and has a positive effect on computational thinking skills (Alsancak Sırakaya, 2019; Noh and Lee, 2020; Wu and Su, 2021; Wei, *et al.*, 2021; Yang and Lin, 2024). Computer programming education is known as a preferred approach to developing computational thinking, problem-solving, and algorithmic thinking skills (Shin, Park and Bae, 2013; Lye and Koh, 2014; Akçay and Çoklar, 2016; Zhang *et al.*, 2023). Therefore, it is possible to say that computer programming and 21st-century skills are interrelated concepts.

2.3. *Digital Literacy*

Digital literacy includes the knowledge, skills, and attitudes necessary to perform operations in digital environments more efficiently and effectively (Ferrari, 2012; Onursoy, 2018). In other words, it can be expressed as accessing, analyzing, and evaluating resources and information in the digital environment (Martin, 2005; Erdem *et al.*, 2023). Digital literacy plays an essential role in effective learning for students and allows them to use digital technologies (Belshaw, 2011; Vasile, 2012; Sağıroğlu *et al.*, 2020). It can also be said that 21st-century skills such as problem-solving and creativity are also included in digital literacy (Voogt and Roblin, 2012; Nouri *et al.*, 2020). Students with digital literacy skills can use information effectively and efficiently. In addition, the level of digital literacy required for design and production by using digital tools in online environments is associated with 21st-century skills (Vasile, 2012; Adiawaty *et al.*, 2023). Therefore, it is possible to say that digital literacy has an important role in enabling to acquisition and development of 21st-century skills such as computer programming skills and computational thinking skills.

2.4. *Research Problems*

This study will seek answers to the following research problems in order to determine the relationships among middle school students' digital literacy, computer programming self-efficacy and computational thinking self-efficacy:

1. Do digital literacy skills have a direct effect on middle school students' computer programming self-efficacy?
2. Do digital literacy skills have a direct effect on middle school students' computational thinking self-efficacy?

3. Does computer programming self-efficacy directly affect middle school students' computational thinking self-efficacy?
4. Do digital literacy skills indirectly affect computer programming self-efficacy on middle school students' computational thinking self-efficacy?

3. Research Hypotheses

3.1. *Digital Literacy and Programming Self-Efficacy*

Digital literacy skills are one of the skills that middle school students should have in the 21st century (ISTE, 2016). For example, being able to produce a solution to a problem using technology requires digital literacy skills. It is thought that having digital literacy skills and using digital technologies effectively within the framework of these skills will play an important role in the computer programming skills of students (Günüç *et al.*, 2013; Kılıç, 2022). It is possible to say that digital literacy skills such as 21st-century skills such as problem-solving, analyzing, creative thinking, and collaboration support computer programming skills. Therefore, digital literacy skills can also provide computer programming skills. Digital literacy topics in the curricula of most countries are addressed through computer programming (Wohl *et al.*, 2017). In addition, effort has been made to develop computer programming skills by increasing the digital literacy skills of middle school students (Burke, 2012). Although some have argued that digital literacy skills improve computer programming skills, it has been stated that computer programming skills can also help increase digital literacy skills (Burke, 2012; Akpınar and Altun, 2014). Therefore, it can be stated that these two concepts are intertwined concepts affecting each other. The research hypothesis related to these issues is given below.

H1: Digital literacy skills have a positive effect on middle school students' computer programming self-efficacy.

3.2. *Digital Literacy and Computational Thinking*

Digital literacy is not only about using digital environments effectively and efficiently but also about developing 21st-century skills such as problem-solving. It can also be expressed as a basic skill to be academically successful in the 21st century (Shute *et al.*, 2017). Topics such as programming and digital literacy can be taught in primary and secondary school curricula, usually through different activities and with clear objectives. However, teaching computational thinking skills to students is a more complex process and it is difficult to express the educational objectives clearly. At this point, it can be said that digital literacy skills have the potential to help students acquire and develop computational thinking skills in a more understandable way (Fagerlund *et al.*, 2021). Therefore, it can be suggested that 21st-century skills such as problem- solving and digital literacy may affect computational thinking skills. To be able to use technology in education, it can be stated that having digital literacy and computational thinking skills, which are stated as two intertwined concepts, can be academically effective for middle school students (Menon *et al.*, 2020; George-Reyes *et al.*, 2021; Akiba, 2022). Therefore, it is thought that digital literacy skills can support computational thinking skills. The research hypothesis related to these issues is given below.

H2: Digital literacy skills have a positive effect on middle school students' computational thinking self-efficacy.

3.3. *Programming Self-Efficacy and Computational Thinking*

Computer programming skills are known to include 21st-century skills such as problem- solving and algorithmic thinking. In addition, studies have been conducted with robotic coding and other coding methods or languages and their effects on computational thinking skills have been researched. Accordingly, computer programming education increases or supports students' computational thinking skills and reflective thinking skills (Witherspoon *et al.*, 2017; Yolcu, 2018; Kaya *et al.*, 2020; Fanchamps *et al.*, 2021; Laura-Ochoa *et al.*, 2022). Therefore, it has also been stated that computer programming tools are a mediator in developing computational thinking skills (Oluk *et al.*, 2018; Laura-Ochoa *et al.*, 2022). On the other hand, these two concepts also include psychological dimensions such as computer programming self-efficacy and computational thinking self-efficacy (Gülbahar *et al.*, 2019). Therefore, it is predicted that computer programming self-efficacy affects computational thinking self-efficacy. The research hypothesis related to this issue is given below.

H3: Computer programming self-efficacy has a positive effect on middle school students' computational thinking self-efficacy.

3.4. *The Relationship Between Variables*

Digital literacy skills and computer programming skills are intertwined concepts that affect each other (Zapata-Ros, 2015). At the same time, programming education increases or supports students' computational thinking skills (Yolcu, 2018; Oluk and Çakır, 2019). Digital literacy plays an important role not only in using technology effectively and efficiently but also in developing and supporting 21st-century skills such as computer programming skills, problem-solving, reflective thinking, algorithmic thinking, and computational thinking skills. Therefore, digital literacy is very important in terms of supporting or developing both programming skills and computational thinking skills (Kılıç, 2022). In addition, computational thinking skills are also described as a new digital literacy skill that can be applied to problem-solving processes (Jawawi *et al.*, 2022). Therefore, it is predicted that computer programming skills have an effect on the effect of digital literacy skills on computational thinking self-efficacy. The research hypothesis related to this issue is given below.

H4: Digital literacy skills have an indirect effect of computer programming self- efficacy on middle school students' computational thinking self-efficacy.

4. Method

In this research, the aim was to examine the relationships between middle school students' computational thinking self-efficacy and computer programming self-efficacy as well as digital literacy variables and design a structural equation model that predicts the relationships between variables by testing the fact that whether these variables predict the levels of computational thinking skills. Therefore, the correlational survey model, one of the quantitative research methods, was used in this research (Karasar, 2005).

4.1. *Research Model*

The research model was designed in accordance with the structure of the relevant literature. In this model, hypotheses are stated by drawing one-way arrows between the variables of the research. The hypotheses of the research are shown in Fig. 1.

According to Fig. 1, 4 hypotheses, namely H1, H2, H3 and H4, were formed in the study.

4.2. *Participants*

The study group of the research consisted of 204 middle school students studying at the 5th, 6th, 7th, and 8th-grade levels (age 10–13) in the 2020–2021 academic year. The study group was selected using the appropriate sampling method. Of the participants,

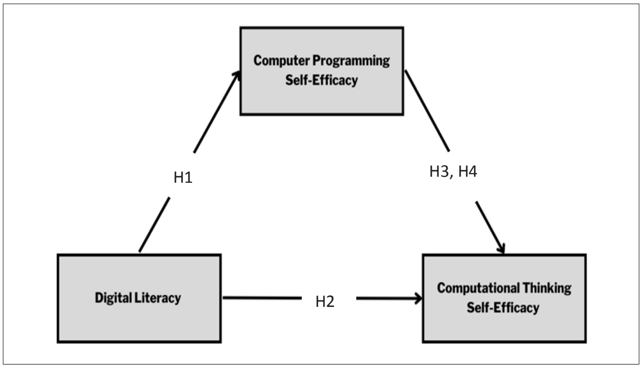

Fig. 1. Hypothesized Research Model.

60.6% were female and 39.4% were male; 20.7% were 5th-grade students, 38.9% were 6th-grade students, 21.2% were 7th-grade students and 19.2% were 8th-grade students.

4.3. *Data Collection Tools*

Before starting the data collection process, necessary permissions were obtained from the parents of the students participating in the research. In the study, data collection tools were applied to middle school students through a questionnaire created in the Google Forms (online) tool. The necessary explanations about the data collection tool were made to the students. Three different data collection tools were used in the research. The first one was the “Digital Literacy Scale”, the second one was the “Computer Programming Self-Efficacy Scale” and the third one was the “Computational Thinking Self-Efficacy Scale”.

4.3.1. *Digital Literacy Scale*

The first data collection tool used in the study was the “Digital Literacy Scale”. This scale was developed by Pala and Basibuyuk (2020). This scale is intended for middle school students. It consists of 21 items and four factors. In addition, the scale was developed in a five-point Likert-type design. In this scale which consists of four factors, the “information-processing” sub-scale consists of 5 items, the “communication” sub- scale consists of 5 items, the “security” sub-scale consists of 6 items, and the “problem- solving” sub-scale consists of 5 items. The Cronbach’ alpha consistency coefficient calculated for the scale in this research was high (α = .877).

4.3.2. *Computer Programming Self-Efficacy Scale*

The second data collection tool used in the study was the “Computer Programming Self- efficacy Scale”. This scale was developed by Kukul, Gökçearslan, and Günbatar (2017). This scale is intended for middle school students. It consists of 31 items and one factor. In addition, the scale was developed in a five-point Likert-type design. The Cronbach’ alpha consistency coefficient calculated for the scale in this research was high (α = .950). The explained variance of the scale is 41.15%. Sample items of the scale are given in the Table 1.

Table 1

Computer Programming Self-Efficacy Scale sample items

Factor	Item
Computer Programming Self-Efficacy	I can correct a programming problem whose solution steps are gi-ven wrong. I can solve complex programming problems by separating them into smaller sub-problems. I can use the cycle instead of repeating instructions. I can show the steps of solution by drawing figures on paper.

4.3.3. *Computational Thinking Self-Efficacy Scale*

The third data collection tool used in the study was the "Computational Thinking Self-Efficacy Scale". This scale was developed by Kukul and Karataş (2019). This scale is intended for middle school students. The scale consists of 18 items and four factors. In addition, the scale was developed in a five-point Likert-type design. In this scale which consists of four factors, the "Logical Inquiry" sub-scale consists of 5 items, the "Abstraction" sub-scale consists of 5 items, the "Discrimination" sub-scale consists of 4 items, and the "Generalization" sub-scale consists of 4 items. The Cronbach' alpha consistency coefficient calculated for the scale in this research is at a high level (α=.884). The explained variance of the scale is 41.15%. Sample items of the scale are given in the Table 2.

4.4. *Data Analysis*

In this research, Structural Equation Modeling was used to test the relationships between middle school students' digital literacy skills, programming self-efficacy, and computational thinking self-efficacy. The bootstrap method was used to calculate the mediation effects in the structural equation method. In the analysis made with the Bootstrap method, a sample of 2000 was determined at a confidence interval of 95%. Since the CI values in the 95% confidence interval do not include the value (0) with the Bootstrap method, the mediation effect is accepted as significant at the 5% level (Hayes, 2015:11).

In this study, a model that explains and predicts the relationships between variables was designed by using observed variables in the AMOS program. CMIN/DF (chi-square fit index test), RMSEA (root mean square error of approximation), GFI (goodness of fit index), CFI (comparative factor index), NFI (normed fit index), and SRMR (Standardized Root Mean Square Residual) fit index values were examined to determine the fit levels of the relationships in the suggested model.

Considering that the Computer Programming Self-Efficacy Scale used in the present study contains too many items (N = 31) due to the limited sample size, the balanced parceling method was used to reduce the number of items in the scale and to create a

Table 2

Computational Thinking Self-Efficacy Scale sample items

Factor	Item
Reasoning	I can decide whether the data to be used for the solution of the problem is adequate or not
Abstraction	I can make comments on the data used for the solution of the problem
Decomposition	If there are sub-problems in the problem, I can manage the solution processes of these subproblems
Generalization	I can make connections between the current problem and previously encountered problems

factor structure (Güler and Çetin, 2020). Due to the known item factor loadings in the scale used, items were grouped according to their factor loadings (Little *et al.* 2013). By applying the method suggested by Little *et al.* (2013), four parcels were created in the scale. Accordingly, in the scale consisting of 31 items, the items were listed according to their factor loadings; items 1-5-9-13-17-21-25, and 29 formed the first dimension; items 2-6-10-14-18-22-26, and 30 formed the second dimension; items 3-7-11-15-19-23-27, and 31 formed the third dimension; items 4-8-12-20-24, and 28 formed the fourth dimension. With the parceling method, the number of suggested model parameters was reduced, factor structures were created and the relationship between structural parameters was maintained (Little *et al.*, 2013; Güler and Çetin, 2020).

5. Findings

The structural equation model coefficients formed by the variables used in the study in line with the data obtained from middle school students are shown in Fig. 2.

Before testing the hypotheses formed in the study, the fit index values of the model were checked by analyses. The acceptable value ranges of the goodness of fit values and the values obtained in the study are given in Table 3.

When Table 1 is examined, the RMSEA value of the model created in the study was observed to be 0.78; NFI = 0.967; CFI = 0.978; SRMR = 0.0355. The value obtained was within the range of acceptable values (Kline, 2005; Meydan and Şeşen, 2011; Tabachnick and Fidell, 2013). Therefore, it can be suggested that the model provided a good fit. In addition, the fit indexes of the model were found to be significant ($p < 0.05$). Within the framework of these results, it was concluded that there was an acceptable fit in the examination of the predicted relationships between middle school students' digital

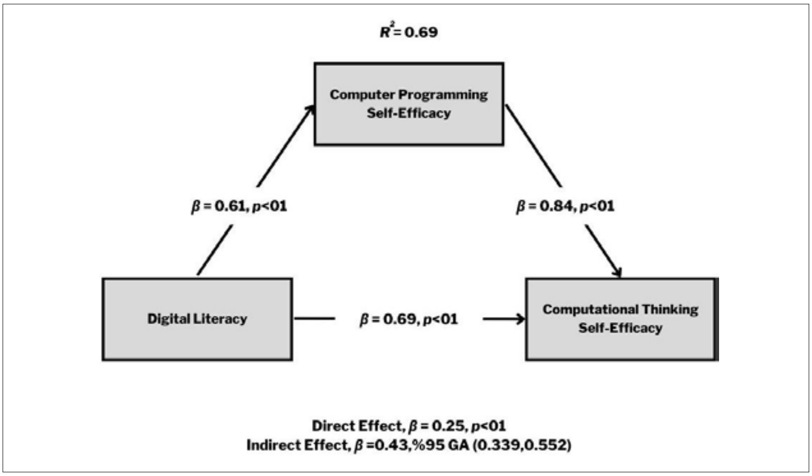

Fig. 2. Structural Equation Model Analysis Results.

Table 3

Results of Goodness of Fit Indexes

Indexes	Values Reached	Acceptable Values	Interpretation
X^2/DF	2.851	3 < (x^2/df) < 5	Acceptable
RMSEA	0.78	< .08	Acceptable
SRMR	0.0355	< .08	Acceptable
GFI	.926	> .90	Acceptable
CFI	.978	> .90	Acceptable
NFI	.967	> .90	Acceptable

literacy skills, computer programming self-efficacy, and computational thinking self- efficacy with structural equation modeling.

After the validation of the model, the research hypotheses were analyzed through the latent variable structural model. The results of the analysis are given in Table 4.

H1 hypothesis was accepted (β = 0.609; p < 0.05; t = 10.883). Accordingly, it can be said that there was a positive effect between middle school students' digital literacy skills and computer programming self-efficacy. It was also observed that the digital literacy skills of middle school students explained 37% (*R2* = 0.370) of the change in computer programming self-efficacy.

H2 hypothesis was accepted (β = 0.686; p < 0.05; t = 11.657). Accordingly, it can be said that there was a positive effect between the digital literacy skills of middle school students and computational thinking self-efficacy. Accordingly, it can be said that there was a positive effect between the digital literacy skills of middle school students and their computational thinking self-efficacy.

H3 hypothesis was accepted (β = 0.843; p < 0.05; t = 20.519). Accordingly, it can be said that there was a positive effect between middle school students' computer programming self-efficacy and computational thinking self-efficacy. It was also observed that the computer programming self-efficacy of middle school students explained 71% (*R2*=0.710) of the change in computational thinking self-efficacy.

H4 hypothesis was accepted (β = 0.438, *CI* [0.339–0.552]; p < 0.05; t = 5.428). Accordingly, it was determined that the indirect effect of digital literacy skills on computational thinking self-efficacy through computer programming self-efficacy of middle school students was significant.

Table 4

Results of Structural Model Analysis

Hypothesis	β	t-value	R^2	Status
H1: Digital Literacy – Computer Programming Self-Efficacy	.609	10.883	.370	Accepted
H2: Digital Literacy – Computational Thinking Self-Efficacy	.686	11.657	.471	Accepted
H3: Computer Programming Self-Efficacy – Computational Thinking Self-Efficacy	.843	20.519	.710	Accepted
H4: Digital Literacy – Computational Thinking Self-Efficacy – Computer Programming Self-Efficacy	.438 [0.339–0.552]	5.428	.777	Accepted

6. Discussion

In this study, the predicted relationships between middle school students' digital literacy skills, computer programming self-efficacy, and computational thinking self-efficacy were revealed by the structural equation modeling method. Accordingly, the four hypotheses determined in the study were confirmed.

According to the analysis results of the model, it was determined that the digital literacy skills of middle school students predicted computer programming self-efficacy at a good level. In other words, it was concluded that digital literacy skills directly affect computer programming self-efficacy. Although digital literacy skills and computer programming self-efficacy have not been discussed together in the literature, there are similar results when compared with different research findings. For example, Kılıç (2022) stated that increasing digital literacy skills is important for the development of computer programming skills. At the same time, students'' digital literacy skills are expected to play an important role in the development of computer programming skills (Günüç *et al.*, 2013). Therefore, it is also seen that in most countries, computer programming courses are given together with digital literacy courses in the curriculum to develop computer programming skills (Wohl *et al.*, 2017). Moreover, digital literacy skills affect computer programming skills, and computer programming skills increase digital literacy levels (Akpınar and Altun, 2014). Therefore, remarkable and fun activities that can increase digital literacy skills and that can be done with current technologies can be ensured to be used in the courses of middle school students in order to psychologically improve their' computer programming self-efficacy. For this purpose, it is necessary to integrate current digital technologies into the computer programming courses of middle school students. Therefore, training programs can be organized periodically for teachers with the help of university institutions so that teachers can develop their digital literacy skills and use these skills in computer programming courses.

As a result of the second analysis in the model, it was determined that the digital literacy skills of middle school students predicted computational thinking self-efficacy at a good level. In other words, it was concluded that digital literacy skills directly affected computational thinking self-efficacy.

Although this finding has not been discussed together with digital literacy skills and computational thinking self-efficacy in the literature, it was seen that there are similar results obtained when compared with different research findings. For example, Akiba (2022), George-Reyes *et al.* (2021) and Menon *et al.* (2020) found that 'having digital literacy skills affects the academic achievement of secondary school students together with their computational thinking skills. Therefore, digital literacy skills play an important role in computational thinking skills to achieve learning goals (Fagerlund *et al.*, 2021). From this point of view, digital literacy skills and computational thinking skills that will contribute to 21st-century skills should be considered together. In addition, as in computer programming courses, it is important that both middle school students and teachers receive training that can enable integration with other courses to provide guidance to students on this issue. Therefore, increasing the computational thinking self-efficacy of middle school students and providing support on where and how to use their digital literacy knowledge will contribute to the development of their skills.

As a result of the third analysis in the model, it was determined that computer programming self-efficacy predicted computational thinking self-efficacy at a high level. In other words, it was concluded that computer programming self-efficacy directly affected computational thinking self-efficacy. When this finding was compared with the research findings in the literature, the results obtained were similar to the studies conducted with middle school students and students at other levels. For example, Durak and Saritepeci (2018) stated in their study that the reason for the negative prediction of students' computational thinking skills was the lack of computer programming education. Therefore, for students to improve their computational thinking skills, it is necessary to emphasize activities in which they can improve their computer programming skills. Although the literature does not psychologically focus on the concepts of computer programming self-efficacy and computational thinking self- efficacy, it can be prescribed that it strongly and directly predicts computer programming self-efficacy, which includes 21st-century skills, and computational thinking self-efficacy, which also includes 21st-century skills.

As a result of the fourth analysis in the model, it was determined that the digital literacy skills of middle school students predicted computational thinking self-efficacy through computer programming self-efficacy at a low level. In other words, it was concluded that digital literacy skills indirectly affected computational thinking self-efficacy through computer programming self-efficacy. Although this finding was not discussed together with the variables addressed within the scope of the study, it was seen that the results obtained are similar to the findings of studies conducted in different contexts. For example, Zapata-Ros (2015) stated in his study that digital literacy skills and computer programming skills are intertwined concepts. At the same time, Yolcu (2018) and Oluk and Çakır (2019) stated in their study that computer programming skills contribute to computational thinking skills. Therefore, Kılıç (2022) stated that digital literacy skills support both programming skills and computational thinking skills. From this point of view, psychologically, it can be predicted that middle school students' digital literacy skills affect computational thinking self-efficacy through computer programming self-efficacy. Therefore, developing middle school students' digital literacy skills and computer programming self-efficacy together will prepare a solid ground for them to develop computational thinking self-efficacy. From this perspective, teachers should be able to prepare the necessary environments for providing the content that will enable secondary school students to develop both their digital literacy skills and their computer programming self-efficacy in their education. Therefore, it can be said that the creation of courses in which prospective teachers can develop their digital literacy and programming skills in their university education will contribute significantly to the development of students' information and computational thinking self-efficacy.

7. Conclusion and Recommendation

As a result, it was determined that there were significant relationships between middle school students' digital literacy skills, computer programming self-efficacy, and computational thinking self-efficacy. It was found that digital literacy skills and computer programming self-efficacy together had a great effect on the computational thinking self-efficacy of middle school students. Then, it was observed that computer programming self-efficacy alone predicted computational thinking self-efficacy, and that finally, digital literacy skills predicted computational thinking self-efficacy. In addition, when computer programming self-efficacy was included in the relationship between digital literacy skills and computational thinking self-efficacy of middle school students, it was determined that the effect of digital literacy skills on computational thinking self-efficacy decreased due to the addition of computer programming self- efficacy. Therefore, to design learning environments that will improve middle school students' computational thinking self-efficacy for future research, it is suggested that students' digital literacy skills and computer programming self-efficacy should be reviewed by considering the model revealed within the scope of the research. It is also suggested to research the effects of programs that will address these skills in the future and to compare them with similar and different studies. The study also showed that both digital literacy skills and computer programming skills directly affect computational thinking skills. However, when the effect size is examined, it is seen that the effect size of computer programming on computational thinking is higher. Therefore, countries should focus on activities that will improve computer programming skills to develop computational thinking skills in their education curricula. If necessary, digital literacy can be integrated into different courses to develop computational thinking skills (Jawawi *et al.*, 2022; Yeni *et al.*, 2022). Computer programming can be designed as a separate course within computer science courses to cover more time. From this perspective, the changes made in the programs of countries such as the USA, India, Australia, the UK, Italy, Finland, and Poland (Bocconi *et al.*, 2016; Falkner *et al.*, 2018; Shah, 2019) can be considered as more appropriate moves to increase computational thinking skills. Finally, to discover new variables that may affect computational thinking self-efficacy, it is suggested to examine the subdimensions affecting computational thinking skills, computer programming skills, and digital literacy skills.

8. Limitation

In this study, there are some limitations arising from data collection tools, participants and methodological preferences. The first of these is the data collection tools. In the data collection tools, "Digital Literacy Scale" was used to measure the concept of digital literacy, "Computer Programming Self-Efficacy Scale" for computer programming self-efficacy and "Computational Thinking Self-Efficacy Scale" for computational thinking self-efficacy. The analyses made with these scales may not objectively reflect the actual performance of the participants. Therefore, it may be useful to make more concrete experimental measurements of the participants. Secondly, quantitative research methods were used to analyze the hypotheses in this study. The exclusion of qualitative data in the study limited the opportunity for participants to fully express their understanding and experiences. Thirdly, the participants of the study were limited to secondary school students selected from a specific sample. And finally, fourthly, the model used in the study and the relationships identified between them are limited within a specific framework based on the literature.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability

The data that support the findings of this study are not openly available due to [reasons of sensitivity e.g., human data] and are available from the corresponding author upon reasonable request.

Statements and Declarations

Conflict of Interest: The authors declare that they have no competing interests.
Ethics Approval: The authors declare that the work is written with due consideration of ethical standards. The study was conducted in accordance with the ethical principles approved by the human experiments.
Informed Consent: All the participants have given their written informed consent.
Consent for Publication: All the participants have given their consent for the publication of the research results.

References

Adiawaty, S., Oci, M., Siminto, S., Ermindyawati, L. (2023). Adaptive Strategies for 21st Century Learning: Digital Literacy and Learning Outcomes in Contemporary Indonesian Higher Education. *International Journal of Teaching and Learning, 1*(2), 115–127.

Aho, A.V. (2012). Computation and computational thinking. *The Computer Journal,* 55(7), 832–835. doi:10.1093/comjnl/bxs074

Akçay, A., Çoklar, A.N. (2016). A proposal for the development of cognitive skills: Programming training. *Educational technology readings*, 121–139.

Akiba, D. (2022). Computational Thinking and Coding for Young Children: A Hybrid Approach to Link Unplugged and Plugged Activities. *Education Sciences*, 12(11), 793.

Akpınar, Y. ve Altun, A. (2014). Factors Effecting Information Technology Teacher Trainees' Programming Process. *Elemantary Education Online*, 13(1), 1–4.

Alsancak Sırakaya, D. (2019). The Effect Of Programming Teaching On Computational Thinking. *Turkish Journal of Social Studies*, 23 (2), 575–590.

Arabacıoğlu, T., Bülbül, H.İ., Filiz, A. (2007). A New Approach to Computer Programming Teaching. *Academic Informatics*, 193–197.

Aşkar, P., ve Davenport, D. (2009). An investigation of factorsrelatedto self-efficacyfor java Programming among engineering students. *The Turkish Online Journal of Educational Technology (TOJET)*, 8 (1), 26–32.

Belshaw, D. (2012). *What is Digitalliteracy? A Pragmatic investigation.* Doctoral dissertation. Durham University.

Benakli, N., Kostadinov, B., Satyanarayana, A., Singh, S. (2017). Introducing computational thinking through hands-on projects using R with applications to calculus, probability and data analysis. *International Journal of Mathematical Education in Science & Technology*, 48(3), 393–427.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). Developing computational thinking in compulsory education-Implications for policy and practice (No. JRC104188). *Joint Research Centre* (Seville site).

Brichacek, A. (2014). *Computational Thinking Boosts Students' Higher-Orderskills*. Retrieved from `https://www.iste.org on 03.02.2023`

Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *The Journal of Media Literacy Education*, 4(2), 121–135.

Büyüköztürk Ş, Kılıç Çakmak E, Akgün ÖE, Karadeniz Ş, Demirel F. (2012). *Scientific Research Methods*. Pegem Publications.

Caballero-Gonzalez, Y. (2022, October). Computational Thinking, a Discipline to Enhance Digital Skills Such as Programming. In: *2022 8th International Engineering, Sciences and Technology Conference (IESTEC)* (pp. 354–359). IEEE.

Castro, C.R., Ossa, L.F.C., Hederich-Martínez, C. (2023). Effects of a computational thinking intervention program on executive functions in children aged 10 to 11. *International Journal of Child-Computer Interaction*, 100563.

Çayak, S., Erol, İ. (2023). Digitalization of Education as a Modernization Trend in the Global World. *Journal Of History School*, 16(62), 353–380.

Curzon, P. (2015). *Computational Thinking: Searching to Speak*. Queen Mary, University of London.

Dağ, F., Şumuer, E., Durdu, L. (2023). The effect of an unplugged coding course on primary school students' improvement in their computational thinking skills. *Journal of Computer Assisted Learning*, 39(6), 1902–1918.

Dagienė, V., Jevsikova, T., Stupurienė, G., Juškevičienė, A. (2022). Teaching computational thinking in primary schools: Worldwide trends and teachers' attitudes. *Computer Science and Information Systems*, 19(1), 1–24.

Demarle-Meusel, H., Sabitzer, B., Sylle, J. (2017, April). The Teaching-Learning- Lab-Digital Literacy and Computational Thinking for Everyone. *In CSEDU* (2) (pp. 166–170).

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.

Edstrand, E., Sjöberg, J. (2023). Professional digital competence (PDC) in teacher education–teacher candidates reasoning about programming when involved in problem- solving activities with digital tools. *Journal of Digital Learning in Teacher Education*, 1–13.

Ekinci, M., Öztop, N.S., Kaymak, Z.D., Güngören, Ö.C. (2023). The relationship between secondary school students' attitudes towards coding and computational thinking skills. *International Journal of Düzce Educational Sciences*, 1(1), 81–90.

Ekinci, M., Öztop, N.S., Kaymak, Z.D., Güngören, Ö.C. (2023). The relationship between secondary school students' attitudes towards coding and computational thinking skills. *International Journal of Düzce Educational Sciences*, 1(1), 81–90.

Erdem, C., Oruç, E., Atar, C., Bağcı, H. (2023). The mediating effect of digital literacy in the relationship between media literacy and digital citizenship. *Education and Information Technologies*, 28(5), 4875–4891.

Ervianti, E., Sampelolo, R., Pratama, M.P. (2023). The Influence of Digital Literacy on Student Learning. *Klasikal: Journal of Education, Language Teaching and Science*, 5(2), 358–365.

Fagerlund, J., Häkkinen, P., Vesisenaho, M., Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12–28.

Falkner, K., Vivian, R., Falkner, N. (2018, April). Supporting computational thinking development in K-6. *In 2018 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)* (pp. 126–133). IEEE.

Fanchamps, N.L., Slangen, L., Specht, M., Hennissen, P. (2021). The impact of SRA-programming on computational thinking in a visual oriented programming environment. *Education and Information Technologies,* 26, 6479–6498.

Fernández, A., Gómez, B., Binjaku, K., Meçe, E.K. (2023). Digital transformation initiatives in higher education institutions: A multivocal literature review. *Education and Information Technologies,* 28, 12351–12382.

Ferrari, A. (2012). *Digital Competence in Practice: An Analysis of Frameworks*. JRC IPTS.

George-Reyes, C.E., Rocha Estrada, F.J., Glasserman-Morales, L.D. (2021, October). Interweaving digital literacy with computational thinking. In: *Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21*) (pp. 13–17).

Gülbahar, Y., Kert, S.B., Kalelioğlu, F. (2019). The Self-Efficacy Perception Scale for Computational Thinking Skill: Validity and Reliability Study. *Turkish Journal of Computer and Mathematics Education (TURCOMAT),* 10(1), 1–29.

Güler, M., Çetin, F. (2020). Item Parceling in Organizational Behavior Studies: Method and Application. *Academic Review of Economics and Administrative Sciences*, 13(1), 61–71.

Günüç, S., Odabaşı, H.F. ve Kuzu, A. (2013). The Defining Characteristics of Students of the 21st Century by Student Teachers: A Twitter Activity. *Journal of Theory and Practice in Education*, 9(4), 436–455.

Hayes, A.F. (2015). An index and test of linear moderated mediation. *Multivariate Behavioral Research*, 50(1), 1–22.

Hsu, T., Chang, S., Hung, Y. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education,* 126, 296–310. `https://doi.org/10.1016/j.compedu.2018.07.004`

ISTE. (2015). *CT leadership toolkit*. Retired from `https://cdn.iste.org/www-root/2020-10/ISTE_CT_Leadership_Toolkit_booklet.pdf` on 03.02.2023

ISTE. (2016). *2016 ISTE Standards for Students.* Retired from `https://www.iste.org/` on 03.02.2023.

Jawawi, D., Jamal, N., Halim, S.A., Sa'adon, N., Mamat, R., Isa, M., Hamed, H.A. (2022). Nurturing secondary school student computational thinking through educational robotics. *International Journal of Emerging Technologies in Learning (iJET),* 17(3), 117–128.

Jun, S., Han, S., Kim, H., Lee, W. (2014). Assessing the computational literacy of elementary students on a national level in Korea. *Educational Assessment, Evaluation and Accountability*, 26, 319–332.

Karasar, N. (2005). *Scientific Research Method*. Nobel Publishing, 37, 9–13.

Kaya M., Korkmaz, Ö., Çakır, R. (2020). The Effect of Gamified Robotics Activities on the Problem Solving and the Computational Thinking Skills of the Secondary School Students. *Ege Journal of Education,* 21(1), 54–70.

Kılıç, S. (2022). The Relationship Between Computational Thinking Skills, Digital Literacy and Programming Self-Efficacy of associate degree Students. *15th International Computer and Instructional Technologies Symposium, 10.*

Kline, R.B. (2005). *Principles and Practice of Structural Equation Modeling* (2. ed.). NY: The Guillford Press.

Korkmaz, Ö., Çakır, R., Özden, M., Oluk, A. ve Sarıoğlu, S. (2015). Investigation of Individuals' Computational Thinking Skills in terms of Different Variables. *Ondokuz Mayis University Journal of Faculty of Education,* 34(2), 68–87.

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

Korkmaz, Ö., Karaçaltı, C., Çakır, R. (2018). Examination of the Students' ComputationalCritical Thinking and Problem-Solving Skills on their Success of Programming Course. *Amasya Education Journal,* 7(2), 343–370.

Kukul, V., Karatas, S. (2019). Computational thinking self-efficacy scale: Development, validity, and reliability. *Informatics in Education*, 18(1), 151–164.

Kukul, V., Gökçearslan, Ş., &Günbatar, M.S. (2017). Computer programming self- efficacy scale (CPSES) for secondary school students: Development, validation, and reliability. *Educational Technology Theory and Practice,* 7(1), 158–179.

Laura-Ochoa, L., Bedregal-Alpaca, N., Vidal, E. (2022). Improving Computational Thinking in Nursing Students through Learning Computer Programming. *International Journal of Advanced Computer Science and Applications*, 13(5).

Lawanto, K., Close, K., Ames, C., Brasiel, S. (2017). Exploring Strengths and Weaknesses in Middle School Students' Computational Thinking in Scratch. *In Emerging Research, Practice, and Policy on Computational Thinking* (307–326).

Little, T.D., Rhemtulla, M., Gibson, K., Schoemann, A. (2013). Why the items versus parcels controversy needn't be one. *Psychological Methods*. 18(3), 285–300.

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

Martin, A. (2005). DigEuLit – a European framework for digital literacy: A progress report. *Journal of eLiteracy*, 2(2), 130–136.

Menon, D., Sowmya, B.P., Romero, M., Viéville, T. (2020). Going beyond digital literacy to develop computational thinking in K-12 education. *In Epistemological Approaches to Digital Learning in Educational Contexts* (17–34).

Meydan, C.H., ve Şeşen, H. (2011). *Structural Equation Modeling AMOS Applications*. Detay Publishing

Mudra, H. (2020). Digital literacy among young learners: How do EFL teachers and learners view its benefits and barriers?. *Teaching English with Technology,* 20(3), 3–24.

Noh, J., Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational technology research and development*, 68, 463–484.

Nouri, J., Zhang, L., Mannila, L., Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry,* 11(1), 1–17.

Nouri, J., Zhang, L., Mannila, L., Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry,* 11(1), 1–17.

Oluk A., Korkmaz, Ö., Oluk, H. A. (2018). Effect of Scratch on 5th Graders' Algorithm Development and Computational Thinking Skills. *Turkish Journal of Computer and Mathematics Education (TURCOMAT),* 9(1), 54–71.

Oluk, A., Çakır, R. (2019). Investigating University Students' Computational Thinking Skills in Terms of Logical Mathematical Intelligence Problem Solving Skills. *Journal of Theoretical Educational Science,* 12(2), 457–473.

Onursoy S., (2018). Digital Literacy Levels of University Youth: A Research on The Students of Anadolu University. *Gümüşhane University e-Journal of Faculty of Communication*, 6(2), 989–1013.

Özden, M.Y. (2015). *Computational Thinking*. Retrieved from `http://myozden.blogspot.com.tr/2015/06/computational-thinking-bilgisayarca.html` on 03.02.2023.

Pala, S.M., Basibuyuk, A. (2020). A Study of Developing Digital Literacy Scale for 10–12 Age Group Students. *Mediterranean Journal of Educational Research,* 14(33), 542–565.

Peracaula-Bosch, M., González-Martínez, J. (2024). Towards a Hermeneutics of Computational Thinking in Wing's Approximations. *Journal of Educational Computing Research,* 61(8), 139–158.

Petrovica, S., Anohina-Naumeca, A., Avanesova, J. (2022). The Impact of Digitalisation of Higher Education: The Case of Latvia and Nordic-Baltic Region. *Applied Computer Systems*, 27(1), 19–29.

Ramazanoğlu, M. (2021). The effect of robotic coding applications on attitudes of the secondary school students towards computers and their perceptions of self-efficacy regarding computational thinking skills. *The Journal of Turkish Social Research,* 25(1), 163–174.

Rubinstein, A., Chor, B. (2014). Computational thinking in life science education.

PLoS Computational Biology, 10(11).

Sağıroğlu Ş., Bülbül H. İ., Kılıç A., Küçükali M., (Ed.)., (2020). *Digital Literacy: Tools Methodologies Practices and Recommendations*. Şeref Sağıroğlu, Halil İbrahim Bülbül, Mustafa Küçükali, Ahmet Kılıç (Ed.). Nobel Publications

Sánchez-Cruzado, C., Santiago Campión, R., Sánchez-Compaña, M.T. (2021). Teacher digital literacy: The indisputable challenge after COVID19. *Sustainability,* 13(4), 1858.

Shah, V. (2019). CSpathshala: bringing computational thinking to schools. *Communications of the ACM,* 62(11), 54–55.

Shin, S., Park, P., Bae, Y. (2013). The effects of an information-technology gifted program on friendship using scratch programming language and clutter. *International Journal of Computer and Communication Engineering,* 2(3), 246.

Shute, V.J., Sun, C., Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research* *review,* 22, 142–158.

Sırakaya, D.A. (2019). The Effect of Programming Teaching on Computational Thinking. *The Journal of Turkish Social Research,* 23(2), 575–590.

Sun, L., Hu, L., Zhou, D. (2021). Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors. *Thinking Skills and Creativity,* 42, 100926.

Tabachnick, B.G., Fidell, L.S. (2013). *Using Multivariate Statistics* (6th ed.). Boston, MA: Pearson.

Timotheou, S., Miliou, O., Dimitriadis, Y., Sobrino, S.V., Giannoutsou, N., Cachia, R., ... Ioannou, A. (2023). Impacts of digital technologies on education and factors influencing schools' digital capacity and transformation: A literature review. *Education and Information Technologies,* 28(6), 6695–6726.

Tsai, C.-Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior,* 95, 224–232.

Turhan, E.M. (2023). Bilgi İşlemsel Düşünme. *International Journal of Social Humanities Sciences Research,* 10(92), 434–439.

Uslu, N.A. (2018). The Effect of Visual Programming Activities on Secondary School Students' Computational Thinking Skills. *Ege Journal of Educational Technologies,* 2(1), 19–31.

Voogt, J., Roblin, N.P. (2012). A comparative analysis of international frameworks for 21st century competences: Implications for national curriculum policies. *Journal of Curriculum Studies,* 44(3), 299–321.

Wei, X., Lin, L., Meng, N., Tan, W., Kong, S.C. (2021). The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy. *Computers & Education,* 160, 104023.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., *et al.* (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology,* 25(1), 127–147.

Wing, J.M. (2006). Computational thinking. *Communication of ACM,* 49, 33–35.

Wing, J.M. (2008). Computational thinking and thinking about computing. *Phil. Trans. R. Soc. A,* 366, 3717–3725.

Witherspoon, E.B., Higashi, R.M., Schunn, C.D., Baehr, E.C., Shoop, R. (2017). Developing computational thinking through a virtual robotics programming curriculum. *ACM Transactions on Computing Education (TOCE),* 18(1), 1–20.

Wohl, B.S., Beck, S., Blair, L. (2017). The Future of the Computing Curriculum: How the Computing Curriculum Instills Values and Subjectivity in Young People. *Online Submission,* 1(1).

Wu, S.Y., Su, Y.S. (2021). Visual programming environments and computational thinking performance of fifth-and sixth-grade students. *Journal of Educational Computing Research,* 59(6), 1075–1092.

Yang, T.C., Lin, Z.S. (2024). Enhancing elementary school students' computational thinking and programming learning with graphic organizers. *Computers & Education,* 209, 104962.

Yildiz, A.G.M., Çiftçi, E., Karal, H. (2017). Computational thinking and programming. *Educational Technologies Readings,* 75–86.

Yolcu, V. (2018). *The Effect Of Using Robotics On Academic Success, Computational Thinking Skills And Transfer Of Learning In Programming Education*. (Publication No. 509835) [Master dissertation, Suleyman Demirel University].

Zapata-Ros, M. (2015). *Computational Thinking: A New Digital Literacy*. Red-Revısta De Educacıon A Dıstancıa, (46).

Zhang, J.H., Meng, B., Zou, L.C., Zhu, Y., Hwang, G.J. (2023). Progressive flowchart development scaffolding to improve university students' computational thinking and programming self-efficacy. *Interactive Learning Environments,* 31(6), 3792–3809.

M.M. Gümüş is a PhD candidate in the field of educational technology. His main research interests are digital competences and computer science education. He is currently working as a lecturer at Amasya University Design Vocational School of Design, Department of Computer Technologies.

V. Kukul holds a PhD degree in educational technology and is an associate professor in the same field. His main research interests are teaching programming, teacher training, digital storytelling and distance education. He is currently working at Amasya University, Faculty of Education, Department of Computer Education and Instructional Technology.

Ö. Korkmaz holds a PhD in educational technology and is a professor in the same field. His primary research interests are computer thinking, computational education and computer science education. He is currently working at Amasya University, Faculty of Engineering, Department of Computer Engineering.

Appendix

Computer Programming Self-Efficacy Scale

Item No	Item	1- Strongly Disagree	2- Disagree	3- Undecided	4- Agree	5- Strongly Agree
I24	I can enable the program to produce accurate results.					
I6	I can solve the problem via different solutions.					
I16	I know how to use the programming variables.					
I22	I can operate the program I have developed.					
I27	I can record the program I have developed.					
I31	I can explain my idea of software project step by step.					
I30	Among the multiple software projects, I select the one that is the fittest for the criterion.					
I5	I select the fittest knowledge for solving the programming problem.					
I4	I investigate the knowledge that is required for solving the programming problem.					
I10	Among various steps of solution, I select the fittest one for the solution to the programming problem.					
I7	I can determine the fittest solution to a problem.					
I25	I can make changes on the program.					
I15	I can make preparations (like determining the variables and processes) required for solving the programming problem.					
I3	I can make an interpretation regarding whether or not a programming problem could be solved.					
I8	I can suggest different solutions in order to solve the programming problems.					
I26	I can correct the mistakes about the coding in the program.					
I19	I determine the solution to the programming problem step by step.					
I20	I know the stages of programming.					
I29	I can explain the process of developing a software project.					
I17	When necessary, I can change the order of the processes designed for solving a programming problem.					
I28	I can share my program with other people via the internet.					
I23	I can enable the perfect functioning of the program.					
I14	I can discuss the different steps being developed for solving the programming problem.					
I13	I can correct a programming problem whose solution steps are given wrong.					
I21	I know where to write the program codes.					
I12	I share the steps of solution to the programming problem with my friends.					
I2	I can solve complex programming problems by separating them into smaller sub-problems.					
I1	I can understand whether a problem is a programming problem or not.					
I19	I know what the operators +, -, *, /, >,<, = mean in a programming.					
I18	I can use the cycle instead of repeating instructions.					
I11	I can show the steps of solution by drawing figures on paper.					

Computational Thinking Self-Efficacy Scale

Item No	Item	1- Strongly Disagree	2- Disagree	3- Undecided	4- Agree	5- Strongly Agree
I1	I recognize repetitive structures in data or images.					
I2	I evaluate the steps necessary for solving the problem from different perspectives.					
I3	I carry out more than one task at the same time to solve a problem.					
I4	I distinguish whether a problem I encounter is similar to problems I have encountered before.					
I5	I analyze the data I collect to solve the problem.					
I6	I relate problems to real life.					
I7	I sort data according to their types (text, number, sequence, etc.).					
I8	I understand whether the problem consists of sub-problems.					
I9	I decide whether the data to be used to solve the problem is sufficient.					
I10	I comment on the data I use to solve the problem.					
I11	I make connections between the problem I encounter and the problems I have encountered before.					
I12	If the problem has sub-problems, I manage the solution processes of these sub-problems.					
I13	I find the fastest solution that works correctly among different process steps.					
I14	I understand how a problem I encounter differs from problems I have encountered before.					
I15	I organize the data I collect in a way that is more understandable for solving the problem.					
I16	I decide whether the problem solution I choose is appropriate for the purpose.					
I17	If the problem has sub-problems, I break it down into smaller sub-problems.					
I18	I develop different solutions for solving a problem.					