Received: 27 November 2021 | Revised: 5 June 2023 | Accepted: 29 June 2023
DOI: 10.1111/jcal.12850

ORIGINAL ARTICLE

Journal of Computer Assisted Learning WILEY

The effect of an unplugged coding course on primary school students' improvement in their computational thinking skills

Funda Dağ[1,2] | Evren Şumuer[1,2] | Levent Durdu[1,2]

[1]Education Faculty, Computer Education and Instructional Technologies Department, Kocaeli University, Kocaeli, Turkey

[2]Computer Education and Instructional Technologies, Kocaeli University, Kocaeli, Turkey

Correspondence
Funda Dağ, Education Faculty, Computer Education and Instructional Technologies Department, Kocaeli University, 41000, Umuttepe/İzmit, Kocaeli, Turkey
Email: fundadag@kocaeli.edu.tr

Funding information
Scientific Research Project Unit of Kocaeli University, Grant/Award Number: 2019/015

Abstract

Background: Today, computational thinking (CT), which is considered to be a form of literacy, has taken its place in the ICT curriculum of many countries at the K-12 level. Therefore, there is a need for more evidence with regards to a theoretical and practical understanding of CT skills' development of K-12 students.

Objectives: The purpose of the study was to investigate the effect of an unplugged coding course on primary school students' development of CT skills, differences in their CT skills in terms of socio-demographics such as gender, computer ownership, daily computer use, and home Internet access, and the relationship between their CT and 21st century skills.

Methods: The research was based a quasi-experimental design with one-group pretest-posttest with follow-up. The CT skills of 212 third and fourth grade students at a public primary school of Turkey were measured with a CT Skills Test before, after, and about ten weeks following having attended an unplugged coding course.

Results and Conclusions: The results of the study showed that the unplugged coding course statistically significantly improved the participants' CT skills, specifically in algorithmic design, abstraction, evaluation, decomposition, and generalization. Moreover, the findings indicated that primary school students' CT skills were not associated with their socio-demographics. In addition, the students' CT skills were found to be positively and statistically significantly correlated with their collaboration and communication skills.

Implications: This study contributes to understanding of the effects of unplugged activities on the development of primary school students' CT skills, which is beneficial to teaching practices for CT skills in the primary education.

KEYWORDS
21st-century skills, Computational thinking skills, primary education, unplugged coding

1 | INTRODUCTION

Technology has changed the way in which we live, learn, and work. It has also changed our style of communication, has affected our learning habits, and has even replaced certain professions. Many traditional

Project website: http://www.dusunencocuklar.com
Course Materials: http://www.dusunencocuklar.com/bilgisayarsiz-etkinliklerle-algoritma/

business areas have become out-of-date or obsolete, while new professions, most of which are somehow related to technology, have emerged, and therefore require new and varied qualifications. Changes in professional qualifications require children and young people to acquire more skills than ever before, which necessitates educational systems that focus on the education of individuals who can think critically as well as the ability to develop different approaches to problem-solving. Therefore, computational thinking (CT) skills are essential for all individuals in today's digital society (Kaila et al., 2018). CT is defined as a form of thinking and behaviour that can be applied to real-world problems (Kong, 2016; Snow et al., 2019). Different disciplines and strategies can be used to improve the students' CT skills. Mostly, computer programming is offered as a means to support and develop their CT skills (Grover & Pea, 2013; Selby & Woollard, 2014).

Computer science education from an early age has become an important issue for many countries to develop individuals who have the required skills in CT. In this context, CT has become a component of information and communication technologies (ICT) curricula in many countries (Kong, 2016; Lockwood & Mooney, 2018; Yadav et al., 2017). The United States (K-12 Computer Science Framework, 2016), England (Department for Education, 2013), Hong Kong (Coolthink, n.d.) and Turkey (Turkish Ministry of National Education, [MoNE], 2018) are just some of the countries in which ICT curricula have been updated for CT, or where some CT-based practices have been started at the primary or secondary school level.

CT is defined as a mental and behavioural process which is combining problem-solving and design to create useful solutions, by being aware of the possibilities in computing science (Kong & Wang, 2020; Millwood et al., 2018). Computing provides a rich context to enhance the learners' ideas. An awareness of the potential of information systems and computer algorithms can help learners tackle with personal, societal and value challenges. Developing an understanding of algorithms and writing a computer program through coding are helpful for learners to support their capacity for self-directed learning, and to prepare themselves for their future by empathy, inquiry, imagination and persistence (Millwood et al., 2018). Learning environments that support students' skills necessary for the 21st-century information society such as problem-solving, project creation and presentation, creativity, analytical and critical thinking are emphasized in curriculums (K-12 Computer Science Framework, 2016; MoNE, 2018). The International Society for Technology in Education (ISTE, 2011) states that CT includes creative thinking, algorithmic thinking, critical thinking, problem-solving, cooperative learning and communication skills, and it is a problem-solving approach that combines technology and thinking. Studies on CT in primary schools show that cooperation has a positive influence on CT skills (Kong et al., 2018; Kuo & Hsu, 2020), CT is not related to creative thinking (Hershkovitz et al., 2019), CT does not have a significant influence on problem-solving skills (Kalelioğlu, 2015), or on the contrary, it has a positive influence on problem-solving skills (Asad et al., 2016). In addition, research indicates that programming education, which includes learning objectives for the improvement of CT skills, is positively correlated with the students' creative thinking, mathematical skills, metacognition skills as well as in areas such as spatial skills and reasoning (Scherer et al., 2019). On the other hand, a meta-analysis of studies on CT education between 2006 and 2017 stated that most studies were descriptive and focused more on CT performance, but research on information society skills was very limited (Hsu et al., 2018). Therefore, more studies are needed on CT skills and their relation to 21st-century skills (Wong & Cheung, 2020).

In order to teach CT, there is a need for effective approaches for all learners, but different from ones aimed for computer science professionals. However, there has been no consensus on how to teach CT skills in an efficient and effective way, which environments and methods to use for it and how to measure it (Guzdial, 2008; Kale et al., 2018; Román-González et al., 2017). Non-technology-based approaches, such as unplugged activities (Saxena et al., 2020), and technology-based approaches such as block-based programming (Park, 2019), and basic physical programming (Angeli & Valanides, 2020) are used when teaching CT at primary and early childhood education. For CT education, the main focus is not on teaching the use of technology or tools, but on using CT with a full understanding of them (Kim et al., 2013). In addition, because of the inclusion of CT objectives in ICT curricula, CT skills or understanding can be thought to be related with technology usage or competence. However, there remains a need for more evidence on the effect of CT skills on the students' learning, on how teacher training should be directed and improved for CT skills, and on how effective ICT are on CT skills (Snow et al., 2019; Webb et al., 2017). In addition, there is a need for more evidence on issues such as identifying effective methods and techniques that teachers can use to develop the students' CT at different development stages, how to assess their improvement in CT, the contribution of teaching process into their CT skills development and performance at different education levels (Alves et al., 2019; Ching et al., 2018; Grover & Pea, 2013). Hence, there is a need for experimental studies that examine the students' cognitive skills as well as their programming skills (Scherer et al., 2019). Also, research is needed to provide more studies on the transfer effects of CT-related teaching processes (Webb et al., 2017). These studies can provide a theoretical and practical understanding of CT skills' development for students and K-12 educators.

The current study aimed to investigate the effect of unplugged activities on students' CT skills, differences in their CT skills in terms of socio-demographic factors as well as the relationship between their CT skills and 21st-century skills. To this end, a one group pretest–post-test study with a follow-up test after a 10 weeks was conducted. This research is expected to contribute to the relevant literature because it provides evidence on the effect of approaches on students' CT skills, and the factors related to CT skills, especially at the primary school education.

2 | LITERATURE REVIEW

2.1 | CT as a set of multidimensional skills

When computational thinking was first addressed in the 1960s, Alan Perlis defended that university students from all disciplines needed knowledge of both programming and computational theory (Perlis, 1964). In the 1980s, Seymour Papert pioneered the idea of children developing procedural thinking skills through LOGO programming in K-12 classes (Papert, 1980). The CT concept was later highlighted by Wing (2006), expressed as using a set of thinking skills, processes and approaches in the field of computer science to solve complex problems.

CT is a multidimensional thinking skill. Wing (2011) stated that CT is primarily concerned with making complex tasks abstract and dividing them into smaller component tasks and emphasized the most important and highest-level thinking skill for CT as the abstraction process. Subdimensions of CT have been discussed in a number of studies, such as Angeli et al. (2016), Barr and Stephenson (2011), Bocconi et al. (2016, p. 16-19), Lee et al. (2011), Selby and Woollard (2013) and Wing (2008). Although various dimensions of CT were emphasized in these studies, the most common were algorithmic thinking, decomposition, generalization, automation and abstraction. The computer science field has a considerable influence on the definition and interpretation of CT dimensions as a multidimensional thinking skill.

Selby et al. (2014) suggested the assessment of learning according to a CT taxonomy based on Benjamin Bloom's cognitive domain taxonomy and the SOLO (Structure of the Observed Learning Outcome) taxonomy. This proposed taxonomy includes the relationships between programming pedagogy and the perceived difficulty level of CT skills (Selby et al., 2014). Although it bears similarities with other taxonomies for CT skills, it is aimed to facilitate the implementation and evaluation of CT within a classroom setting.

CT taxonomy consists of five components, namely 'evaluation', 'algorithm design', 'generalization', 'abstraction' (firstly for functionality, then for data) and 'decomposition' according to the perceived difficulty levels from easy to difficult (Selby, 2014, 2015b). Evaluation is the assessment of the extent to which a solution (algorithmic) is suitable to its intended purpose. Algorithm design is the way to reach a result by clearly defining the steps required to perform a certain task or in solving a specific problem. Generalization is a way to quickly solve new problems based on the existing problem-solving experiences. Abstraction involves hiding details and eliminating unnecessary complexity. This skill is related to determining and focusing on the right details in order to reveal the problem. Decomposition is the ability to break down a problem, especially more complex ones. CT components specified in the taxonomy can be used for CT learning objectives and concrete thinking process can be analysed in accordance with the taxonomy (Csizmadia et al., 2015, p.14; Selby, 2015a, 2015b).

2.2 | Approaches and methods to teach CT skills in primary schools

There is no consensus yet for the K-12 level on what approaches and methods should be used in developing CT skills or how to analyse CT skills (Caeli & Yadav, 2020; Guzdial, 2008; Kale et al., 2018; Román-González et al., 2017; Webb et al., 2017). The main purpose of including CT skills in instructional processes is not for students to progress in the field of computer science, but for them to acquire the habit of applying computational thinking to different courses, as well as in life in general (ISTE, 2011).

It is difficult to find a common solution to the problem of how best to develop CT skills. Can CT be taught by teaching children problem-solving? Or is learning how to write a computer program sufficient as a means to gaining skills in CT? While CT relies on problem-solving, it cannot, however, be said that problem-solving fully covers CT (Caeli & Yadav, 2020). Problem-solving is a broad term that covers different strategies in various fields. In this context, CT can be expressed as a specialized form of problem-solving. In addition, computer programming can be considered as a special form of CT. Therefore, while computer programming can support the development of CT, it cannot ensure CT by itself (Selby, 2014).

Research on teaching CT at the primary and early childhood education indicated that mostly unplugged activities (Faber et al., 2017; Saxena et al., 2020) and rarely simple plugged activities (e.g. block-based programming like Scratch, code.org) are utilized (Asad et al., 2016; Kalelioğlu, 2015; Park, 2019). In addition, some studies have adopted a basic physical programming approach and used tangible objects (e.g. Bee-bot robots) in activities (Angeli & Valanides, 2020; Muñoz-Repiso & Caballero-González, 2019). At the primary education, CT can be also developed at the entry level without the need of plugged events (Looi et al., 2018). Unplugged Computer Science may be considered a suitable approach for this purpose (CS Unplugged, n.d.). It enables learners to get to know and understand the basis of computer science by materials such as paper, pencils, paint, rope, card and balls, and mostly through game-based activities (Bell et al., 2009). Considering that primary school students are likely to encounter CT for the first time, it may be the appropriate approach to teach firstly with unplugged activities in order to facilitate the learners' awareness of basic computer science concepts (Gaio, 2017). As plugged activities entail learning the features of computer programming tools and increase learners' cognitive load, unplugged activities may contribute more to the development of CT skills than plugged ones (Hermans & Aivaloglou, 2017). In the literature, the use of unplugged and plugged activities together in the classroom environment is also recommended (Caeli & Yadav, 2020; Del Olmo-Muñoz et al., 2020; Millwood et al., 2018).

2.3 | Approaches and methods to evaluate CT skills in primary schools

One of the most significant challenges in teaching CT skills is the evaluation the effectiveness of the instructional processes designed to improve these skills, since there is no comprehensive evaluation framework for CT practices in computer science at the K-12 level (Alves et al., 2019; Grover & Pea, 2013; Snow et al., 2019). In the K-12 computer science framework curriculum, performance tasks (such as project-based and portfolio-based assessment methods) are suggested for the evaluation of the students' achievements (K-12 Computer Science Framework, 2016). However, the CT development levels of both primary and secondary school students are evaluated using different techniques, including product and portfolio evaluation (Lin, 2012), computer screen recording (Kim et al., 2018), reflection reports (Choi, 2013) and approaches that provide quantitative data such as course activity evaluations and questionnaires (Kaila et al., 2018). In addition to these techniques, CT skills or course achievement tests (Tran, 2019; Zapata-Cáceres et al., 2020), certain scales relating to CT skills (Gülbahar, Kert, & Kalelioğlu, 2019; Korkmaz et al., 2015), and psychometric-based CT scales (Román-González et al., 2017; Tsai et al., 2019) are also available to measure the CT perceptions of students.

Although different techniques can be used to evaluate CT development (Brennan & Resnick, 2012), tests are one of the most useful measurement tools for analysing CT development of the students who encounter CT education at the primary education for the first time. For example, Tran (2019) developed and applied a knowledge test in order to analyse elementary school students' CT development (at 13 years of age). Zapata-Cáceres et al. (2020) proposed an assessment tool in the test structure for CT training for young children (aged 5–10 and 10–12 years old). In addition, the international Bebras model (Dagienė & Sentance, 2016) is another approach used to evaluate CT at the knowledge and skill level. The Bebras model was developed with tasks that involve computer science principles as a means of promoting CT skills, as well as the algorithmic, logical and operational thinking skills of students of all ages. However, some Bebras questions have been reported as being somewhat very complex, especially at the primary education level (Del Olmo-Muñoz et al., 2020). Empirical analyses on Bebras task categories related to CT skills are ongoing (Dagienė et al., 2017; Ternik et al., 2020).

2.4 | Factors influencing CT Skills

In addition to research on how CT can be developed, it is also necessary to identify and understand which factors can influence the students' CT skills' acquisition (So et al., 2020). Factors such as self-efficacy, interest in computing, and prior computing experience should be considered for learning CT (Ketenci et al., 2019). A study conducted by Del Olmo-Muñoz et al. (2020) underlined that unplugged activities were more effective in the development of CT skills, and the motivation of children at early ages. In addition, it found that there was no significant relationship between students' CT skills and their gender. However, Angeli and Valanides (2020) reported a significant CT development in students (aged 5–6 years old) and differences between female and male students in terms of activities that they benefited from. They reported that female students were more successful in collaborative activities, while male students were more successful in individual activities, which contrasts with the findings of Kong et al. (2018). Kalelioğlu (2015) stated that training fourth-grade students using plugged CT activities did not reveal any difference in primary school students' reflective thinking skills in problem-solving, but that small increases were observed in questioning and evaluation factors. Kalelioğlu (2015) also emphasized that a slight change was seen in the students' self-confidence in their problem-solving abilities, with male and female students having performed equally. In conclusion, in the literature, there are inconclusive results regarding the role of gender in the CT development of the students, especially at early ages.

Despite some studies on factors associated with CT at the primary education, there are not enough studies to clearly reveal important factors related to CT education. Since computational thinking has overlapping concepts with ICT education, the students' computers and the Internet usage is one of the important factors influencing CT. Studies in the literature revealed different findings on the influence of the students' computer ownership or use on their CT skills. For example, Oluk and Korkmaz (2016) reported that the duration of daily computer use did not affect the CT skills of students who were taught plugged activities. Similarly, Alsancak Sarıkaya (2020) showed that the level of computer experience has no impact on CT skills. Yildiz Durak and Sarisepetci (2018) investigated factors influencing CT skills in their study with students from grades 5 to 12. They reported that the use of information technology and the duration of daily internet use had no statistically significant influence on CT skills. On the other hand, there are some studies indicating that computer ownership (Paf & Dinçer, 2021) and Internet use (Xing & Lu, 2022) influence CT skills. In conclusion, considering inconclusive findings in the literature, it is important to investigate how technology-related variables such as prior computing experience, computer ownership, or Internet access influence students' CT skills, especially at primary school level.

Some curricula have emphasized learning environments that support students' skills' acquisition deemed necessary for the 21st-century information society such as problem-solving, project creation and presentation, creativity, as well as analytical and critical thinking (K-12 Computer Science Framework, 2016; MoNE, 2018). On the other hand, due to different findings on the relationship of CT skills with information society skills (Angeli & Valanides, 2020; Asad et al., 2016; Hershkovitz et al., 2019; Kalelioğlu, 2015; Kong et al., 2018; Kuo & Hsu, 2020) there is a need for further investigation on this relationship.

2.5 | Course design process

In the current research, as an intervention, an unplugged coding course was designed by considering the learning outcomes in the Turkish curriculum for an 'Information Technologies and Software' primary education course (MoNE, 2018), and also the taxonomy of computational thinking skills (Selby, 2014). The curriculum began to be implemented for the first time in the academic year in which the research was conducted. Level 1 and Level 2 of the 'Problem Solving and Programming' theme of the curriculum was referenced in the course design. In Turkey, the CT-related learning objectives at the first and second levels of the primary school ICT curriculum, of which there are four levels, are mostly unplugged activities and designed for the acquisition of basic CT skills (Gülbahar & Kalelioğlu, 2018). The third and fourth levels support the transition to unplugged and plugged activities. The ICT primary school curriculum in Turkey is compatible with the Computer Science Teachers Association's (CSTA) K-12 Computer Science Standards (CSTA, 2017), and has an emphasis on student-centered teaching methods, adopting a design that supports high-level participation in learning. So, the course is considered to be suitable for a collaborative learning approach.

In the course design process (see Figure 1), firstly, CT skills were ordered from simple to complex (Selby, 2014), and then the learning outcomes of the curriculum were combined into suitable CT skills. Afterwards, the re-organization/design of learning activities/tasks for each learning outcome group was realized. The activities/tasks were either adopted from unplugged coding activities included in the Information Technologies and Software (Level 1 and Level 2) coursebooks, developed in collaboration with Google and the Turkish Ministry of National Education (Gülbahar, Kalelioglu, et al., 2019a, 2019b), or developed independently by the researchers of the current study. During the development of the activities, behaviour and activity recommendations for CT skills in the curriculum were taken into account (Csizmadia et al., 2015; Dorling & Walker, 2015).

Since the students had not taken any other course related to either information technology or coding prior to taking part in the current study and the curriculum emphasizes student-centered teaching methods, the whole teaching process was implemented in a classroom environment where students can work in groups (three persons).

One activity for each of 14 weeks of the course was designed as semi-structured or unstructured. After the students were informed about which task/s they were going to do before the activity, they undertook the tasks in accordance with their own thinking styles with techniques such as cut-combine, find-fix, find-draw and think-write. The last activity (Make and Tell your solution) of the course was organized as an unstructured activity. In the first 2 weeks of the course, introductory activities related to basic information on information technologies and coding were implemented (at the knowledge acquisition level for CT skills) then the others were in order (see Table 4).

In the study, a test was used to evaluate CT skills development of students at the cognitive level. The 'Computational Thinking Skills Test (CTST)' was based on the CT taxonomy (Selby, 2014) in order to evaluate the development of CT in different dimensions (algorithmic design, evaluation, abstraction, decomposition, generalization). Although it is emphasized that various evaluation tools such as product portfolio, (Lin, 2012), reflection reports (Choi, 2013), course activity evaluations (Kaila et al., 2018) could be useful for comprehensive analysis of CT skills at the affective or behavioural level, easy-to-use evaluation tools could be beneficial for the teachers to analysis of CT skills.

While the 'Information Technologies and Software' course is not compulsory, it is offered during free activity lesson periods depending on the primary school teachers' preferences, and they have limited knowledge and awareness of CT in Turkey as similarly with many countries. Primary school teachers have been reported to have diverse conceptualization of CT and struggle with how to teach and how to evaluate (Caeli & Bundsgaard, 2020; Garvin et al., 2019). In this context, it is important to provide teachers with teaching resources for CT skills. All materials of the course were designed to ensure that students could integrate CT skills with different courses (ISTE, 2011; Tang et al., 2020), so that all of them are compatible with primary school education programs such as mathematics, science and life studies. Therefore, activities of the course and test items were associated with algorithmic design skills in different lessons at primary school level, such as performing the sequence of operations in a science experiment and using of simple mathematical operation skills (See Figure 2).

2.6 | Purpose of the study

The main purpose of the current study was to examine the effect of unplugged activities on the CT development of students, as well as the difference in their CT skills in terms of sociodemographic characteristics such as gender, computer ownership, daily computer use and the Internet access at home. In addition, any relation between 21st-century skills and CT skills was also investigated. Specifically, answers were sought to the following research questions:

1. Is there a significant effect of the unplugged coding course on primary school students' computational thinking skills?

a. Is there a statistically significant difference among primary school students' pretest, posttest, and follow-up test mean scores on computational thinking skills?

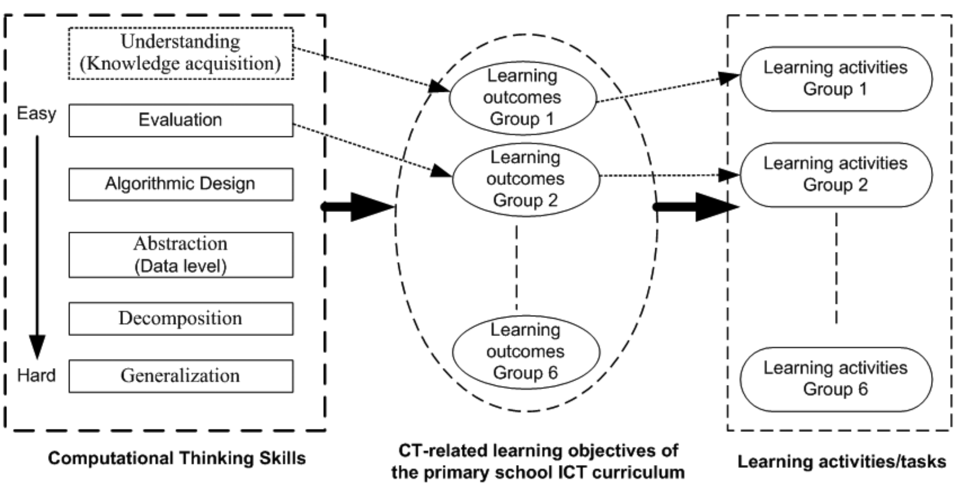

FIGURE 1 The course design process.

FIGURE 2 A photo showing the classroom environment.

TABLE 1 Research design of the study.

Study group	O_1	X	O_2	O_3
Experimental	Pretest on computational thinking skills	14-week unplugged coding course	Posttest on computational thinking skills	Follow-up test on computational thinking skills

b. Is there a statistically significant difference between primary school students' pretest and posttest mean scores in terms of the dimensions of computational thinking skills (algorithmic design, evaluation, generalization, abstraction, and decomposition)?

2. Do primary school students' posttest mean scores on computational thinking skills statistically significantly differ in terms of gender, computer ownership, home Internet access and duration of daily computer usage after controlling for their pretest mean scores?
3. Is there a statistically significant relationship between primary school students' posttest mean scores on computational thinking skills and their 21st-century learning and innovation skills (i.e. creativity and innovation skills, critical thinking and problem-solving skills, and collaboration and communication skills)?

3 | METHOD

3.1 | Research design

The current study was based on a one group pretest–posttest design with follow-up at 10 weeks after the intervention. In the one-group pretest–posttest design, there is only a single group, which is measured before and after being exposed to an intervention (Cohen et al., 2018; Fraenkel et al., 2012). In the current study, primary school students' computational thinking skills were measured before and after a 14-week unplugged coding course. In addition, a follow-up test on the participants' computational thinking skills was then administered after about ten weeks (i.e. the summer vacation) the unplugged coding course. As the school required all students to take this course, the study did not include a control group. The research design of the study is illustrated in Table 1.

3.2 | Participants

In the current study, data were collected from 212 students at a single public primary school in Turkey. Of the participants, 54.2% ($n = 115$) were third-grade students, while 45.8% ($n = 97$) were in their fourth grade. Of the students, 51.9% ($n = 110$) were male, while 48.1% ($n = 102$) were female. Their ages ranged from 8 to 11 years old, with a mean of 9.30 years (SD = 0.05). Majority of the students (62.3%, $n = 132$) did not own a computer at home and only 17.5% ($n = 37$) of them had internet access at home. The average daily computer use of the students was mainly less than 1 h (39.2%, $n = 83$) and 1–2 h (33.5%, $n = 71$). Of the students, 10.8% ($n = 23$) stated that they never used a computer during the day and 16.5% ($n = 35$) of the students stated that they used it for more than 2 h. The demographics of the participant students are summarized by grade level in Table 2.

3.3 | Data collection tools

In the current study, data were collected using the Computational Thinking Skills Test, the 21st-Century Learning and Innovation Skills Scale, and a demographics form. The following describes each of the employed tools.

3.3.1 | Computational thinking skills test (CTST)

The students' computational thinking skills were measured using the Computational Thinking Skills Test (CTST), which was developed by the researchers. At the beginning of the test development process, the Bebras tasks (Bebras, n.d.) and CT assessment tools used at the primary school level (Tran, 2019) were examined. In the literature,

TABLE 2 Primary school students' demographics by grade.

	3rd grade		4th grade		Total	
Demographics	*n*	%	*n*	%	*n*	%
Gender						
Male	57	49.6	53	54.6	110	51.9
Female	58	50.4	44	45.4	102	48.1
Computer ownership						
Yes	45	60.9	35	63.9	80	37.7
No	70	39.1	62	36.1	132	62.3
Internet access at home						
Yes	93	80.9	82	84.5	175	82.5
No	22	19.1	15	15.5	37	17.5
Duration of daily computer use						
None	10	8.7	13	13.4	23	10.8
Less than 1 h	50	43.5	33	34.0	83	39.2
1–2 h	34	29.6	37	38.1	71	33.5
More than 2 h	21	18.3	14	14.4	35	16.5

Note: $N = 212$.

Bebras tasks are suggested for the assessment of CT, but it has also been stated that the complexity of some Bebras questions may be too high for primary school students (Del Olmo-Muñoz et al., 2020). Bebras tasks are stated as being a skills transference tool and are suggested to be applied at some point following an intervention so as to evaluate retention or transfer effect (Román-González et al., 2019). For this reason, an original measurement tool deemed appropriate to the curriculum and participants' age level was developed for students encountering CT for the first time. The aim was for CT skills to be analysed at the cognitive level using the CTST.

The test consists of 15 multiple-choice items, of which three items were based on algorithmic design, three items on abstraction, five items on evaluation, two items on decomposition, and two items on generalization. Scores on computational thinking skills are computed by summing the correct responses in the test, giving a possible range in scores from 0 (low) to 15 (high). In addition, scores for each computational thinking skills' dimension are computed by aggregating correct responses to the related items. Achieving a higher score in the test indicates higher computational thinking skills.

The multiple-choice items in the CTST were developed based on the learning outcomes in the 'Problem Solving and Programming' theme of the Turkish primary education curriculum for the Information Technologies and Software course (MoNE, 2018) and the taxonomy of CT skills (Selby, 2014). The items in the CTST were developed by the researchers. Then, the content validity of these items was reviewed independently by an expert in the field of computer science, an expert in the field of computer education and instructional technology, two computer teachers and two primary school teachers. Based on their comments on the items of the CTST, the researchers eliminated redundant items from the test and then made necessary revisions on the remaining items to improve their clarity.

A pilot study was conducted with 239 primary school students who had previously taken a coding course (89 third grade; 150 fourth grade) in order to evaluate the quality and usefulness of the CTST's 15 items. The results of an item analysis (see Table 3) showed that nine of the items considered to be 'good' (Item 2, 4–10, 12), whereas six of the items (Item 1, 3, 11, 13–15) required some form of revision as were considered too easy (diff. index >0.7), too difficult (diff. index <0.3) or poorly discriminating item (disc. Index <0.40) (Crocker & Aigina, 2008; Haladyna, 2016). The reliability of the test was found to be moderate (KR-20 $= 0.61$) (Salvucci et al., 1997). Considering these results, the researchers applied the necessary revisions to the six aforementioned CTST items. An example item from the CTST is shown in Figure 3.

3.3.2 | 21st-Century Learning and Innovation Skills Scale

In the current study, the students' 21st-century learning and innovation skills were measured using the 21st-Century Learning and Innovation Skills scale, which was developed in the Turkish language by Boyaci and Atalay (2016) to assess the skills of primary school students. It contains 39 items on a three-point, Likert-type scale (i.e. never, sometimes and always), measuring three factors, namely, 'creativity and innovation' (20 items), 'critical thinking and problem solving' (12 items), and 'collaboration and communication' (seven items). Boyaci and Atalay (2016) found the reliability of the total scale ($\alpha = 0.96$) and the subscales (creativity and innovation $\alpha = 0.96$; critical thinking and problem-solving $\alpha = 0.94$; and collaboration and communication $\alpha = 0.89$) to be satisfactorily high. In the current study, the Cronbach alpha coefficient of the total scale was found to be 0.90, indicating a satisfactorily high level of reliability (Field, 2009). In addition, the Cronbach alpha coefficient values of the subscales ranged from 0.71 to 0.82, which indicated a sufficient level of reliability. The students' scores for 21st-century learning and innovation skills were computed by averaging their responses to the items in each subscale.

3.3.3 | Demographics form

A demographics form was used to collect demographic information about the participant students such as their grade, age, gender, computer ownership, home Internet access and their duration of daily computer use.

3.4 | Research setting of the study

In the current study, the unplugged coding course covered a period of 14 weeks (i.e. one full semester). During each week of the course, the students received a 40-minute class in which they were required to collaboratively work on completing unplugged coding activities in

TABLE 3 Item analysis results of CTST.

Item	Taxonomy level	Item responses (%) A	B	C	D	Omit	Diff. (*p*)	Index disc. (*D*)	Point biserial corr.
1[a]	Algorithmic design	5	90[b]	3	2	0	0.90	0.19	0.25
2[c]	Algorithmic design	8	8	20	59[b]	6	0.59	0.49	0.34
3[a]	Algorithmic design	16	9	3	72[b]	1	0.72	0.13	0.01
4[c]	Abstraction	8	43[b]	11	34	5	0.43	0.46	0.28
5[c]	Abstraction	41[b]	31	13	11	3	0.41	0.42	0.26
6[c]	Evaluation	29	39[b]	16	11	5	0.39	0.51	0.30
7[c]	Evaluation	8	15	55[b]	15	8	0.55	0.50	0.31
8[c]	Abstraction	19	19	39[b]	13	11	0.39	0.40	0.20
9[c]	Evaluation	21	13	45[b]	10	12	0.45	0.43	0.25
10[c]	Evaluation	11	50[b]	16	13	10	0.50	0.50	0.30
11[d]	Evaluation	13	44	27[b]	8	10	0.27	0.20	0.04
12[c]	Decomposition	41[b]	19	13	13	15	0.41	0.57	0.37
13[e]	Decomposition	25	18	15	15[b]	17	0.15	0.26	0.23
14[e]	Generalization	17	15	34	18[b]	17	0.18	0.20	0.29
15[e]	Generalization	28[b]	30	11	14	17	0.28	0.33	0.24

Note: KR-20 = .61.
[a]Very easy item.
[b]Keyed response.
[c]Effective discriminating item.
[d]Poorly discriminating item.
[e]very difficult item.

In order to calculate Zeynep's mathematics achievement score, a calculation robot follows the steps given below. When Zeynep tells the calculation robot the Math scores in the table, what does it find the Zeynep's math achievement score to be after performing the operation steps?

A) 75 B) 80 C) 85 D) 90

Operation steps:
Begin
Enter PerformanceScore
Enter ProjectScore
Enter ExamScore
Score = (PerformanceScore + ProjectScore) / 2
AchievementScore = (Score + ExamScore) / 2
End

Zeynep's Math Scores		
Performance Score	**Project Score**	**Exam Score**
60	80	100

FIGURE 3 An item from the CTST.

groups of three. The students took the course in multiple sessions, and the number of students in each session ranged from 20 to 24 students, as it was given in a 24-seated classroom with cluster seating arrangement. The course was delivered by four instructors who had each received training on how to teach information technologies and coding. The taxonomy level, learning outcomes and the course activities are presented in Table 4.

During the 14-week course, the activities were organized as semi-structured or unstructured. After the students were informed about the task(s) which they were going to do, they completed the activity based on their preference for tasks such as cut-combine, find-fix, find-draw and think-write. The last activity of the course was designed as an unstructured activity presented (i.e. the course activity of 'Make & Tell your solution').

3.5 | Data collection procedures

The primary school students voluntarily participated in the current study. Human subject protection was ensured whereby; prior to the study, the students and their parents were each informed about the purpose and procedures of the research study. In addition, parental informed consent was obtained for each student who was to take part in the study. This form explained the confidentiality of the data collection process and their voluntary participation, meaning that they had the right to withdraw their participation at any time without any consequences. Any identifier information from the participants was not used in the study. Permission from the primary school principal was also secured in terms of collecting data from students attending the school. Moreover, the local institutional ethics committee

TABLE 4 Taxonomy levels, learning outcomes and course activities.

Taxonomy level	Week	Learning outcomes (Students will be able to …)	Activity name	Task(s)
Understanding (Knowledge acquisition)	Week 1	• Realize that computers work with commands. • Explain that computers can be used for different purposes.	I'm getting to know the computer	• Defines and gives examples related to concepts such as computer, programming, software and algorithms.
	Week 2	• Design simple process flows for daily life situations. • Realize that process flows for daily life are like algorithms followed by computers.	Sequential operation steps	• Sorts steps into the correct order for some behaviours (e.g. sharpening a pencil).
Algorithmic design	Week 3	• Make logical queries for solutions of different problems.	What should I wear?[a]	• Chooses the right clothes from options for three different seasons.
			Shopping for stationery[a]	• Finds the cheapest stores to buy stationery items on a shopping list.
	Week 4	• Estimate the result of an algorithm.	Which picture?[a]	• Chooses pictures of creatures whose body parts were joined in accordance with given steps.
			Freight train[a]	• Considers the given algorithm, finds the right freight train whose wagons are sorted to be unloaded in the shortest time.
	Week 5	• Create an algorithm by sequencing given operational steps in a logical order.	I sorted step-by-step[a]	• Sorts steps correctly for actions such as brushing teeth, making paper airplanes.
	Week 6	• Find the incorrect operational step in a sequential list.	Let us find the mistakes[a]	• Finds errors by comparing given commands with the path marked on the grid map.
Abstraction	Week 7	• Create an algorithm for the solution of a simple, daily life problem.	I'm creating my own algorithm	• Creates an algorithm for activities such as table preparation and drinking a bottle of water that was in a school bag.
			Algorithm for foods	• Plays a game in which a character is selected, selecting food and drinks suitable for the character, and finding the shortest path to food and drinks on the grid map.
	Week 8	• Collect data for the solution of a problem related to daily life (and math, colour and art lessons).	Natural life park	• Finds data types (logical, numeric, or string) from answers given to questions regarding the given pictures.
		• Organize collected data according to their features. • Visualize the collected data.	I visualize the data	• Selects the correct animals according to information provided.
	Week 9	• Write pseudocode for a problem solution.	I am spreading jam on bread	• Writes steps for spreading jam on bread.
			I'm preparing a sandwich	• Finds commands for a suitable path to follow on the grid map to prepare a sandwich.
Evaluation	Week 10	• Test an algorithm.	I'm testing an algorithm	• Checks whether the path given on the grid map follows the given algorithm. • Finds an error in an algorithm.
		• Test an algorithm the students wrote themselves. • Debug an algorithm.	I tested, debugged[a]	• Creates algorithms in which a selected student in a class on the grid map performs various tasks (e.g. opens a window).

(Continues)

TABLE 4 (Continued)

Taxonomy level	Week	Learning outcomes (Students will be able to …)	Activity name	Task(s)
Decomposition	Week 11	• Identify a problem by dividing it into sub-problems.	Divide, make it easy	• Finds solutions and writing relevant operational steps to help a hungry and injured cat.
			The way to school	• Finds suitable vehicles and route to get to school based on available time and money.
Generalization	Week 12	• Discover that a problem may have different solutions.	Many ways, one solution	• Finds solutions to problems related to students' school life.
	Week 13 Week 14	• Design, implement and evaluate the solution to an original problem (using what was learned and group work).	Make & Tell your solution	• Designs a garden on the grid map and creates an algorithm in which a robot picks fruits and vegetables in the garden.

[a]Adopted from unplugged coding activities in the Information Technologies and Software (Level 1 & Level 2) coursebooks (Gülbahar, Kalelioglu, et al., 2019a, 2019b).

reviewed the procedures to be followed in the study and provided ethical approval for the study.

In the current study, data were collected at three time points (i.e. pretest, posttest and follow-up test). First, 1 week prior to the start of the unplugged coding course, the students' computational thinking skills were measured using the CTST in a paper-pencil format by the classroom teachers and the instructors who were informed about how to administer the test (i.e. pretest). The students were given 40 minutes to complete the test in their classrooms. Second, at the end of the course, the CTST was read ministered to the students by the researchers and instructors in order to measure their computational thinking skills after attending the unplugged coding course (i.e. posttest). In addition to the CTST, the students completed the 21st-Century Learning and Innovation Skills scale. Then, approximately ten weeks later following the end of the unplugged coding course, the CTST was administered again, but only to the third-grade students, by the instructors in order to measure the retention of their acquired computational thinking skills (i.e. follow-up test). Whereas all students took both the pretest and the posttest, only the third-grade students were available for the follow-up test as the fourth-grade students had graduated from the primary school by the time follow-up test was administered.

3.6 | Data analysis

A paired sample *t*-test was performed to test whether there was a statistically significant difference between the primary school students' pretest and posttest computational thinking skills' mean scores. Preliminary analyses showed no violation of the assumptions of normality. In addition, a repeated measure analysis of variance (ANOVA) was conducted in order to examine the difference among the third-grade primary school students' pretest, posttest and follow-up test computational thinking skills' mean scores. Prior to this, preliminary analyses were performed in order to ensure no violation of the assumptions of normality and sphericity (Field, 2009; Verma, 2016). As Mauchly's test of sphericity indicated violation of the assumption of sphericity ($\chi^2(2) = 25.30$, $p < 0.05$), Greenhouse–Geisser estimates of sphericity ($\varepsilon = 0.82$) was used to correct the degrees of freedom in the analysis. Moreover, a one-way repeated measure multivariate analysis of variance (MANOVA) was conducted in order to examine the difference between the students' pretest and posttest mean scores for each dimension of their computational thinking skills. The preliminary analyses indicated no violation of assumptions of multivariate normality, outliers, linearity, multicollinearity, or equality of covariance matrices, and the data of the study were considered to have passed the assumptions of normality, outliers, linearity, and multicollinearity (Verma, 2016).

Furthermore, a one-way between-groups analysis of covariance (ANCOVA) test was performed to analyse the differences of the students' posttest mean scores by gender, computer ownership, home Internet access and the duration of daily computer usage after controlling their pretest mean scores. The results of the preliminary analyses showed no violation of assumptions of normality, outliers, linearity, homogeneity of variance, multicollinearity, homogeneity of regression, or reliability of covariates (Field, 2009; Tabachnick & Fidell, 2007).

Lastly, the Pearson product–moment correlation coefficient was calculated to examine the relationship between the primary school students' posttest mean scores on computational thinking skills and their 21st-century learning and innovation skills. Prior to the analysis, the assumptions of outliers, linearity and normality were checked, and no violations were found (Field, 2009). In all analyses, the significance level was set as.05.

4 | FINDINGS

4.1 | Effect of unplugged coding course on the primary school students' CT skills

A paired sample *t*-test was conducted to examine any difference between the primary school students' pretest and posttest

TABLE 5 Means, standard deviations and MANOVA results for five dimensions of computational thinking skills.

	Pretest		Posttest			
Dimension	*M*	*SD*	*M*	*SD*	Univariate *F*(1, 211)	Multivariate *F*(5, 207)
Algorithmic Design	1.48	0.84	1.89	0.83	36.24*	65.54*
Abstraction	0.78	0.79	1.20	0.86	41.18*	
Evaluation	1.54	1.14	2.34	1.15	81.79*	
Decomposition	0.54	0.57	0.70	0.63	10.23*	
Generalization	0.44	0.56	0.62	0.60	10.18*	

Note: *F* ratios are Wilks's approximation of Fs.
*$p < 0.05$.

computational thinking skills' mean scores. There was a significant increase in the primary school students' scores in computational thinking skills from pretest ($M = 4.77$, $SD = 2.11$) to posttest ($M = 6.75$, $SD = 2.15$; $t\ (211) = -18.01$, $p < 0.05$, $\eta_p^2 = 0.61$). A one-way repeated measures MANOVA was conducted to investigate in which dimensions of computational thinking skills the primary school students' posttest mean scores were statistically significantly different from their pretest mean scores. The results of the analysis showed that there was a statistically significant difference between the students' pretest and posttest mean scores on overall computational thinking skills; $F\ (5, 207) = 65.54$, $p < 0.05$; *Wilks' Lambda* $= 0.39$; $\eta_p^2 = 0.61$. Regarding the five dimensions of computational thinking skills, there was a statistically significant increase seen in the students' test scores after attending the unplugged coding course in terms of their algorithmic design ($F\ (1, 211) = 36.24$, $p < 0.05$; $\eta_p^2 = 0.15$), abstraction ($F\ (1, 211) = 41.18$, $p < 0.05$; $\eta_p^2 = 0.16$), evaluation ($F\ (1, 211) = 81.79$, $p < 0.05$; $\eta_p^2 = 0.28$), decomposition ($F\ (1, 211) = 10.23$, $p < 0.05$; $\eta_p^2 = 0.05$) and generalization (F (1, 211 $= 10.18$, $p < 0.05$; $\eta_p^2 = 0.05$). The results of the analysis are summarized in Table 5.

A repeated measure ANOVA was conducted to investigate whether or not there was a statistically significant difference among the third-grade primary school students' pretest, posttest and follow-up test computational thinking skills' scores. The repeated measures ANOVA with a Greenhouse–Geisser correction showed that a statistically significant difference was found to exist among the students' mean computational thinking skills' test scores; $F\ (1.64, 168.90) = 42.21$, $p < 0.00$, $\eta_p^2 = 0.29$. Post hoc tests indicated that the students' posttest mean scores ($M = 6.03$ $SD = 1.75$) and follow-up test mean scores ($M = 5.70$, $SD = 2.32$) were statistically significantly higher than their pretest mean scores ($M = 4.08$, $SD = 1.71$, $p < 0.05$). However, there was no statistically significant difference between the students' posttest mean scores and their follow-up test mean scores ($p = 0.60$).

4.2 | Factors influencing the primary school students' CT skills

A one-way between-groups ANCOVA was performed in order to compare the students' posttest mean scores on computational thinking skills of the primary school students after controlling their pretest mean scores in terms of their gender. The covariate, pretest scores on computational thinking skills, were found to be statistically significantly related to the dependent variable $F\ (1, 209) = 220.05$, $p < 0.05$. After adjustment for their pretest scores on computational thinking, there was no statistically significant difference found in the posttest computational thinking skills' mean scores between the male ($M = 6.56$, $SD = 2.10$) and female ($M = 6.95$, $SD = 2.20$) primary school students, $F\ (1, 209) = 0.11$, $p = 0.74$.

A one-way between-groups ANCOVA was used to investigate differences between the posttest computational thinking skills' mean scores for those primary school students who owned a computer and those who did not after controlling for their pretest mean scores. Similar to the previous analysis, the covariate, the pretest scores on computational thinking skills, was found to have a statistically significant relationship with the posttest scores, $F\ (1, 209) = 222.67$, $p < 0.05$. After controlling their pretest scores on computational thinking, there was no statistically significant difference found between those students who owned a computer ($M = 6.56$, $SD = 1.97$) and those who did not ($M = 6.86$, $SD = 2.25$) with respect to their posttest mean scores, $F\ (1, 209) = 0.64$, $p = 0.42$.

Another one-way between-groups ANCOVA was performed with the posttest computational thinking skills' mean scores of those primary school students who had Internet access at home and those who did not, after controlling for their pretest mean scores. The covariate, the students' pretest scores, was found to be statistically significant, $F\ (1, 209) = 225.17$, $p < 0.05$. After adjustment for their pretest scores on computational thinking, there was no statistically significant difference found in the posttest mean scores on computational thinking skills between those students who had Internet access at home ($M = 6.77$, $SD = 2.14$) and those who did not ($M = 6.68$, $SD = 2.25$), $F\ (1, 209) = 0.93$, $p = 0.34$.

A one-way between-groups ANCOVA was conducted in order to investigate the influence of average daily computer usage of the primary school students on their posttest computational thinking skills' mean scores, after controlling for their pretest scores. The students were divided into four groups in terms of their average daily computer use (Group 1: none; Group 2: less than 1 h; Group 3: 1–2 h; and Group 4: more than 2 h). The covariate, the students' pretest mean scores, was found to be statistically significantly related to their posttest computational thinking skills' mean scores, $F\ (1, 207) = 220.21$, $p < 0.05$. After controlling for their pretest scores on computational

TABLE 6 Means, standard deviations and ANCOVA results for daily computer usage.

Average duration of daily computer use	Posttest Scores		ANCOVA
	M	*SD*	*F*(3, 207)
None	6.74	2.00	0.16
Less than 1 h	6.60	2.11	
1–2 h	6.94	2.17	
More than 2 h	6.71	2.37	

Note: * $p < 0.05$.

thinking, however, there was no statistically significant difference found in the students' posttest mean scores with respect to their average duration of daily computer use, F (3, 207) $= 0.16$, $p = 0.92$. The results of the ANCOVA are presented in Table 6.

4.3 | Relationship between CT skills and 21st-century learning and innovation skills

Lastly, Pearson product–moment correlation coefficients were used in order to investigate the relationship of the primary school students' posttest mean scores on computational thinking skills with their 21st-century learning and innovation skills (i.e. creativity and innovation skills, critical thinking and problem-solving skills, and collaboration and communication skills). The results showed that there were non-significant, small, and negative correlations found between the students' posttest scores and their creativity and innovation skills ($r = -0.04$, $p = 0.53$), and their critical thinking and problem-solving skills ($r = -0.05$, $p = 0.46$); whereas, there was a significant, small, and positive correlation found to exist between their posttest scores and their collaboration and communication skills ($r = 0.15$, $p < 0.05$).

5 | DISCUSSION

5.1 | Effect of unplugged coding course on the primary school students' CT skills

The findings of the current study revealed that the students' posttest mean scores and follow-up test mean scores were significantly higher than their pretest mean scores. In addition, there was no statistically significant difference found between the students' posttest mean scores and their follow-up test mean scores. This finding shows that the unplugged coding course, which is based on learning outcomes of a national ICT curriculum, had a significant effect on the primary school students' CT skills, and is also consistent with other studies that have shown that unplugged (or plugged) CT courses can positively contribute to the primary school students' CT skills (Park, 2019; Saxena et al., 2020; Song, 2019; Tran, 2019). Furthermore, it is pointed out that the use of unplugged activities is more beneficial for novice students' CT skills in terms of learning concepts (Gaio, 2017; Park, 2019; Saxena et al., 2020). Based on our study, it can be suggested that unplugged activities are beneficial for the conceptual development of CT skills' the primary school students (3rd and 4th grades) who had not taken any other course related to either information technology or coding prior to taking part in the current study.

Considering the CT taxonomy offered by Selby et al. (2014), the current study revealed that the students' CT skills in all categories of the taxonomy significantly improved following the unplugged course activities. The category in which the students' CT skills were most improved was 'evaluation', followed by 'algorithmic design' and 'abstraction'. The categories of 'generalization' and 'decomposition' were the skills areas that developed the least. The evaluation CT skill is considered easier for students to improve according to the hierarchy of perceived difficulty of the CT taxonomy (Selby, 2014, 2015b), while 'algorithmic design skill' is categorized as second in the hierarchy. Gaio (2017) implemented both types of plugged and unplugged activities with children and concluded that unplugged activities were shown to be beneficial for the meaningful development of algorithm concepts. For evaluation CT skill, it is likely to be more difficult for them to notice and correct their own mistakes if children study with plugged activities than creating their own algorithms. Saxena et al. (2020) conducted a study consisting of first unplugged and then plugged activities with children between the ages of 4 and 6 years old. They found that the use of algorithmic design skills first in the unplugged environment increased children's success in demonstrating algorithmic design skills in the plugged environment. Song (2019) reported that students who learned flowcharting through unplugged activities developed CT skills more and a greater interest in programming than students who use the Scratch tool after having learned about flowcharts. The findings obtained in the current study are partially consistent with the aforementioned other studies. During the 14-week course in the current study, the participant students took part in semi-structured and unstructured activities. The activities, which focused on dividing problems into meaningful smaller steps or bringing together meaningful steps in order to solve a problem, are likely to contribute to the development of students' algorithmic design skills. In addition, the activities that facilitate error analysis (e.g. detecting and correcting the process steps in solving a problem) could improve students' evaluation CT skills.

Abstraction skill is important in the computer science field as well as CT for all, it is recognized as being a difficult skill to develop (Angeli & Valanides, 2020; Gaio, 2017). This skill is addressed in terms of data and function levels (Selby, 2015b). In the current study, being able to distinguish types of data used for problem-solving is considered as data-level abstraction skill and being able to create different algorithms when problem-solving is considered as function-level abstraction. The activities in the current study facilitated abstraction skill at a basic level; therefore, in the CT skills test, the students' abstraction skills were evaluated accordingly. Angeli and Valanides (2020) studied how abstraction skills could be taught to young children, based on Piaget's perspective. They stated that children between the ages of seven and 11 years of age can solve problems applied to concrete objects but are unable to solve problems applied to abstract concepts. On the other hand, Waite et al. (2016) indicate that students use abstraction elements in their learning processes in general, and that abstraction can be taught to children from an early age with appropriate activities such as labelled diagrams, concept maps and storyboards in the context of CT. Similarly, Gibson (2012) advocate that young children are able to think abstractly when concrete reference systems are used as a means to focus their thoughts. Although the findings in our study revealed that primary school students improved their CT skills at the abstraction level, there is still a need for more research on what the scope of abstraction skill should be and on how best it can be taught to young children.

In the current study, primary school students developed solutions by breaking down problems using decomposition activities (e.g. the course activities of ‘Divide, make it easy’, and ‘The way to school’). Additionally, the children were able to apply the skills they had gained during previous activities through generalization activities based on their own solutions (e.g. the course activities of ‘Many ways, one solution’, and ‘Make & tell your solution’). However, there are no consistent findings on decomposition or generalization level CT skills in children. Angeli and Valanides (2020) stated that younger children are able to accomplish complex learning tasks more easily by dividing them into a series of subtasks. In the current study, it can be propounded that the course had less number of learning objectives related to decomposition and generalization skills so these skills were less developed as compared to other CT skills.

It is unlikely that unplugged course activities would result in equal development across all categories of the students' skills according to Selby's (2014) CT taxonomy. When the learning outcomes of the curriculum followed in the current research were grouped according to the taxonomy (see Figure 1); for the conceptual dimensions of CT, there were eight learning objectives for algorithmic design, five for abstraction, three for evaluation, one for decomposition, and two for generalization (see Table 3). Therefore, it can be stated that development of the students' CT skills was found to be in line with the course outcomes.

It is also considered important to investigate the persistence of CT skills acquisition in any CT-based research, and experimental studies involving some form of follow-up testing administered after some significant period will provide better evidence for CT development in the long-term view (Scherer et al., 2019). In the current study, the findings revealed no statistically significant difference between the students' posttest scores and their retention (follow-up) test scores. As a result, it may be concluded that the unplugged CT education contributed to the students' learning.

5.2 | Factors affecting the primary school students' CT skills

Another important finding of the current study was that no relationship was found to exist between the CT skills and gender of the participants. Similarly, some studies in the literature have reported that gender does not significantly influence CT skills (Kalelioğlu, 2015), or that gender has little influence on CT skills but a more significant influence on motivation (Del Olmo-Muñoz et al., 2020). On the other hand, some studies have revealed findings in favour of male students over female students in terms of success and interest in CT studies (Angeli & Valanides, 2020; Kong et al., 2018). There are also studies that have reported gender differences varying depending on the type of activity or problem that focuses on a certain CT skill (Román-González et al., 2017). Although there is no difference found between females and males in terms of coding competence, some studies reveal that females have a different approach to coding and different perspectives on coding during activities (Angeli & Valanides, 2020; Papavlasopoulou et al., 2020). Therefore, it can be concluded that there are different results regarding the role of gender in the CT skill development. Thus, it is suggested that there is a need for future studies examining the influence of gender on CT skills, which could enable to understand participation of both genders in learning on CT.

There are not enough studies showing the relationship between CT skills and demographic characteristics of children other than their gender in primary school-level. In addition, it is indicated that unplugged activities allow students to get a first grip on computational thinking processes by actively engaging them, and it is regarded for students who do not have access to the Internet, computers, and mobile devices (Zhan et al., 2022). In the current study, the relationship between CT skills and the demographic characteristics of computer ownership, daily use of computers, and having home Internet access was also investigated. The current study's findings showed that no significant relationship was found between the CT skills and students' demographics of computer ownership, daily computer use, and home Internet access. In today's world, CT is a necessary literacy to develop along with technology (Fletcher & Lu, 2009; Resnick et al., 2009). Children's CT skills are likely to be related to their perceptions and attitudes towards technology and also their sociodemographic characteristics associated with their relationship with technology. Although in the current study no relationship was found to exist between the participants' CT skills and their demographic characteristics, it may be suggested that there is a need for future studies to examine this relationship in different learning environments of CT skills (e.g. plugged, unplugged or tangible).

5.3 | Relation between CT skills and 21st-century learning and innovation skills

The findings of the current study revealed a small non-significant negative relationship between the students' posttest CT scores and their creativity and innovation skills, and with their critical thinking and problem-solving skills. On the other hand, a small positive correlation was found between the students' posttest CT scores and their collaboration and communication skills. Other studies on CT at the primary school-level, which were based on collaboration in CT education, have reported similar results (Angeli & Valanides, 2020; Kong et al., 2018; Kuo & Hsu, 2020). In the current study, collaborative learning approach was adopted in the learning activities, and therefore it may be stated that CT skills are positively related to the students' collaboration and communication skills.

A small number of studies have focused on the children's problem-solving skills at the primary school level. As an example, Asad et al. (2016) emphasize that students give different solutions to problems during activities, which helps to improve their problem-solving skills. On the other hand, in a study by Kalelioğlu (2015), it was stated that according to the reflective thinking skills test scores applied at the beginning and at the end of the teaching process, no differences in the students' reflective thinking skills for problem-solving were found. In the current study, students solved simple problems in semi-structured activities. Therefore, it may be stated that the activities in the current study did not lead students to a comprehensive critical thinking or problem-solving process, and that it may be expected that there is no relationship between the students' critical thinking and problem-solving skills and their CT skills.

There have not been many studies on the relationship between CT skills and creativity, however. Hershkovitz et al. (2019) examined the relationship between CT acquisition and creativity in a game-based learning environment with middle school students, and while their results revealed no relationship between computational creativity and CT acquisition, there was a positive relationship found between the students' individual analysis and their creativity skills. Kong et al. (2018) stated that students with better cooperative attitudes in CT education have more creative self-efficacy, but that collaboration attitudes in CT education are not related to programming competence. In the current study, the primary school students engaged in an unstructured activity which was aimed at improving their generalization skills (e.g. the course activity of 'Make your own solution'). Since the students completed a single activity where they could show their original analysis and solution, it may not be sufficient for them to develop new or creative solutions to a problem. Just as it is not possible to fully acquire all CT skills from a single course, it is probably not possible to fully develop 21st-century skills as an outcome from having attended a single unplugged course.

A positive relationship was found to exist between the students' collaboration and communication skills and their CT skills. It may be concluded that since the course design was based on collaborative learning activities, it was contributed to the development of primary students' CT skills, as well as to their collaboration and communication skills. Although a few studies investigated the relationship between CT skills and collaborative learning, studies show that collaborative learning supports the development of CT skills (Angeli & Valanides, 2020; Kong et al., 2018). Since the primary education course in the current study was not specifically designed to improve creativity and innovation, critical thinking and problem-solving skills, it may be expected that no significant relationship was revealed between these skills and CT.

6 | CONCLUSION

The current study has three main findings. First, the use of unplugged approach is beneficial to develop CT skills of the primary school students who encounter CT at first time. Second, although there is no relationship between the students' CT skills and their demographic characteristics (gender, computer ownership, daily computer use, home Internet access), it is recommended that future studies should examine in depth the relationship between the participants' CT skills and their demographic characteristics. Third, CT should be regarded as a literacy that has emerged within the framework of the requirements of the 21st-century information society. It may be suggested that course designs that take into account the skills necessary for the information society as well as the CT skills can be beneficial for students to support the development of both their CT and 21st-century skills.

Although the current research has provided evidence in terms of the effect of a course on CT skills and factors related to students' CT skills at the primary school level, the study group was limited to a single public primary school in Turkey, which thereby limits the generalizability of the study's results. Moreover, the current study employed a quasi-experimental design with one-group pretest-posttest with follow-up at 10 weeks following the intervention. In order to better control threats to internal validity, future studies could use the true experimental research design in order to examine the effect of unplugged coding courses on students' CT skills. In addition, due to summer vacation break, it was possible for the researchers to measure the students' CT skills for the follow-up 10 weeks after the post-test. Given this relatively long period of time for the study, it is possible the other factors a coding course which students might take during summer vocation, or the development of their CT skills due to other subjects (e.g. mathematics) could influence students' CT skills measured at the follow-up test. Furthermore, while the current research allowed certain conclusions to be drawn, discussed, and highlighted, the focus of the analysis was on students who were encountering CT for the first time within the framework of a set curriculum. CT skills seem an inevitable part of life in order to meet the today's social needs and also in the future and, therefore, they should rightfully be integrated into today's curricula to prepare for tomorrow's society.

FUNDING INFORMATION

The study was supported by a grant from the Scientific Research Project Unit of Kocaeli University under Grant number 2019/015.

CONFLICT OF INTEREST STATEMENT

We have approved the manuscript and agree with submission to *Journal of Computer Assisted Learning*. There are no conflicts of interest to declare.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/jcal.12850.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon request. Additionally, the CT Skill Test and the course activities which are developed within the research can be accessed at http://dusunencocuklar.com.

ETHICS STATEMENT

All procedures performed in this research were in the framework with the ethical standards and after the necessary permits obtained by the Kocaeli University Scientific Research and Publication Ethics Committee.

ORCID

Funda Dağ https://orcid.org/0000-0002-0868-6647
Evren Şumuer https://orcid.org/0000-0002-5345-8555
Levent Durdu https://orcid.org/0000-0003-3788-7226

REFERENCES

Alsancak Sarıkaya, D. (2020). Investigation of computational thinking in the context of ICT and mobile technologies. *International Journal of Computer Science Education in Schools*, *3*(4), 50–59. https://doi.org/10.21585/ijcses.v3i4.73

Alves, N. D. C., Von Wangenheim, C. G., & Hauck, J. C. (2019). Approaches to assess computational thinking competences based on code analysis in K-12 education: A systematic mapping study. *Informatics in Education*, *18*(1), 17–39. https://doi.org/10.15388/infedu.2019.02

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, *105*, 105954. https://doi.org/10.1016/j.chb.2019.03.018

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society*, *19*(3), 47–57. https://drive.google.com/file/d/1aQmsP0teOS76e_KHpQ6pYrBQh-MNMfae/view

Asad, K., Tibi, M., & Raiyn, J. (2016). Primary school pupils' attitudes toward learning programming through visual interactive environments. *World Journal of Education*, *6*(5), 20–26. https://doi.org/10.5430/wje.v6n5p20

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. https://doi.org/10.1145/1929887.1929905

Bebras. (n.d.). Bebras international challenge on informatics and computational thinking. https://www.bebras.org/

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20–29. https://researchportal.bath.ac.uk/en/publications/computer-science-unplugged-school-students-doing-real-computing-w

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing computational thinking in compulsory education: Implications for policy and practice. *European Commission*, EUR 28295 EN, 15-21. https://doi.org/10.2791/792158

Boyaci, S. D. B., & Atalay, N. (2016). A scale development for 21st Century skills of primary school students: A validity and reliability study. *International Journal of Instruction*, *9*(1), 133–148. https://doi.org/10.12973/iji.2016.9111a

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 annual meeting of the American Educational Research Association. https://dam-prod.media.mit.edu/x/files/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Caeli, E. N., & Bundsgaard, J. (2020). Computational thinking in compulsory education: A survey study on initiatives and conceptions. *Educational Technology Research and Development*, *68*(1), 551–573. https://doi.org/10.1007/s11423-019-09694-z

Caeli, E. N., & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. *TechTrends*, *64*, 29–36. https://doi.org/10.1007/s11528-019-00410-5

Ching, Y.-H., Hsu, Y.-C., & Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends*, *62*, 563–573. https://doi.org/10.1007/s11528-018-0292-7

Choi, H. (2013). Pre-service teachers' conceptions and reflections of computer programming using Scratch: Technological and pedagogical perspectives. *International Journal for Educational Media and Technology*, *7*(1), 15–25. https://jaems.jp/contents/icomej/vol7/IJEMT7.15-25.pdf

Cohen, L., Manion, L., & Morrison, K. (2018). *Research methods in education* (8th ed.). Routledge.

Computer Science Teachers Association. (2017). K-12 Computer science standards. https://www.csteachers.org/page/standards

CoolThink. (n.d.). A curriculum framework for Hong Kong students. https://www.coolthink.hk/en/ct/

Crocker, L., & Aigina, J. (2008). *Introduction to classical and modern test theory*. Cengage Learning.

CS Unplugged. (n.d.). Computer science without a computer. https://csunplugged.org/en/

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking - A guide for teachers* (p. 14). Project report. https://eprints.soton.ac.uk/424545/

Dagienė, V., & Sentance, S. (2016). It's computational thinking! Bebras tasks in the curriculum. In A. Brodnik & F. Tort (Eds.), *Informatics in schools: Improvement of informatics knowledge and perception* (pp. 28–39). Springer. https://doi.org/10.1007/978-3-319-46747-4_3

Dagienė, V., Sentance, S., & Stupurienė, G. (2017). Developing a two-dimensional categorization system for educational tasks in informatics. *Informatica*, *28*(1), 23–44. https://doi.org/10.15388/Informatica.2017.119

Del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of primary education. *Computers & Education*, *150*, 103832. https://doi.org/10.1016/j.compedu.2020.103832

Department for Education. (2013). National curriculum in England: Computing programmes of study. https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study

Dorling, M., & Walker, M. (2015). Computing progression pathways. https://community.computingatschool.org.uk/resources/1692/single

Faber, H. H., Wierdsma, M. D., Doornbos, R. P., van der Ven, J. S., & de Vette, K. (2017). Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network*, *12*, 13–24. https://etenjournal.com/2020/02/07/teaching-computational-thinking-to-primary-school-students-via-unplugged-programming-lessons/

Field, A. (2009). *Discovering statistics using SPSS: (and sex and drugs and rock 'n' roll)* (3rd ed.). SAGE.

Fletcher, G. H., & Lu, J. J. (2009). Education human computing skills: Rethinking the K-12 experience. *Communications of the ACM*, *52*(2), 23–25. https://doi.org/10.1145/1461928.1461938

Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2012). *How to design and evaluate research in education*. McGraw-Hill.

Gaio, A. (2017). Programming for 3rd graders, scratch-based or unplugged?. In T. Dooley & G. Gueudet (Eds.), *Proceedings of the 10th Congress of European Research in Mathematics Education* (pp. 3683–3690). DCU Institute of Education and ERME https://hal.archives-ouvertes.fr/hal-01950502/

Garvin, M., Killen, H., Plane, J., & Weintrop, D. (2019). Primary school teachers' conceptions of computational thinking. In *Proceedings of the*

50th ACM technical symposium on computer science education (SIGCSE'19) (pp. 899–905). Association for Computing Machinery.

Gibson, J. P. (2012). Teaching graph algorithms to children of all ages. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education* (pp. 34–39). ACM. https://doi.org/10.1145/2325296.2325308

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Gülbahar, Y., & Kalelioğlu, F. (2018). Bilişim teknolojileri ve bilgisayar bilimi: Öğretim programı güncelleme süreci (Information technologies and computer science: Curriculum update process). *Milli Eğitim Dergisi*, *47*(217), 5–23.

Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Self-efficacy perception scale for computational thinking skills: A validity and reliability study. *Turkish Journal of Computer and Mathematics Education*, *10*(1), 1–29. https://doi.org/10.16949/turkbilmat.385097

Gülbahar, Y. G., Kalelioglu, F., Kert, S. B., Karaosmanoğlu, G., Yoncacı, M., Kuleli, S., & Demircan, S. (2019a). *Bilişim Teknolojileri ve Yazılım Dersi: 1. seviye öğretmen rehberi [Information technologies and software course: Level 1 teacher guide]*. Milli Eğitim Bakanlığı.

Gülbahar, Y. G., Kalelioglu, F., Kert, S. B., Karaosmanoğlu, G., Yoncacı, M., Kuleli, S., & Demircan, S. (2019b). *Bilişim Teknolojileri ve Yazılım dersi: 2. seviye öğretmen rehberi [Information technologies and software course: Level 2 teacher guide]*. Milli Eğitim Bakanlığı.

Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM*, *51*(8), 25–27. https://doi.org/10.1145/1378704.1378713

Haladyna, T. M. (2016). Item analysis for selected-response test items. In S. Lane, M. R. Raymond, & T. M. Haladyna (Eds.), *Handbook of test development* (2nd ed.). Routledge.

Hermans, F., & Aivaloglou, E. (2017). To Scratch or not to Scratch?: A controlled experiment comparing plugged first and unplugged first programming lessons. In E. Barendsen & P. Hubwieser (Eds.), *Proceedings of the 12th Workshop on Primary and Secondary Computing Education* (pp. 49–56). ACM. https://doi.org/10.1145/3137065.3137072

Hershkovitz, A., Sitman, R., Israel-Fishelson, R., Eguíluz, A., Garaizar, P., & Guenaga, M. (2019). Creativity in the acquisition of computational thinking. *Interactive Learning Environments*, *27*(5–6), 628–644. https://doi.org/10.1080/10494820.2019.1610451

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (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

International Society for Technology in Education. (2011). Computational thinking in K-12 education leadership toolkit. https://cdn.iste.org/www-root/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4

K-12 Computer Science Framework. (2016). *ACM*. https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf

Kaila, E., Laakso, M. J., & Kurvinen, E. (2018). Teaching future teachers to code - Programming and computational thinking for teacher students. In K. Skala, M. Koricic, T. G. Grbac, M. Cicin-Sain, V. Sruk, S. Ribaric, S. Gros, B. Vrdoljak, M. Mauher, E. Tijan, P. Pale, & M. Janjic (Eds.), *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 677–682). IEEE. https://doi.org/10.23919/MIPRO.2018.8400127

Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., & Grise, K. (2018). Computational what? Relating computational thinking to teaching. *TechTrends*, *62*, 574–584. https://doi.org/10.1007/s11528-018-0290-9

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior*, *52*, 200–210. https://doi.org/10.1016/j.chb.2015.05.047

Ketenci, T., Calandra, B., Margulieux, L., & Cohen, J. (2019). The relationship between learner characteristics and student outcomes in a middle school computing course: An exploratory analysis using structural equation modeling. *Journal of Research on Technology in Education*, *51*(1), 63–76. https://doi.org/10.1080/15391523.2018.1553024

Kim, B., Kim, T., & Kim, J. (2013). Paper-And-Pencil Programming Strategy Toward Computational Thinking For Non-Majors: Design Your Solution. *Journal of Educational Computing Research*, *49*(4), 437–459. https://doi.org/10.2190/EC.49.4.b

Kim, C., Yuan, J., Vasconcelos, L., Shin, M., & Hill, R. B. (2018). Debugging during block-based programming. *Instructional Science*, *46*, 767–787. https://doi.org/10.1007/s11251-018-9453-5

Kong, S.-C. (2016). A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education*, *3*(4), 377–394. https://doi.org/10.1007/s40692-016-0076-z

Kong, S.-C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education*, *127*, 178–189. https://doi.org/10.1016/j.compedu.2018.08.026

Kong, S. C., & Wang, Y. Q. (2020). Formation of computational identity through computational thinking perspectives development in programming learning: A mediation analysis among primary school students. *Computers in Human Behavior*, *106*, 106–230. https://doi.org/10.1016/j.chb.2019.106230

Korkmaz, Ö., Çakır, R., & Özden, M. Y. (2015). Adaptation of computer thinking skill levels scale (CTSL) to secondary school level. *Gazi Journal of Educational Science*, *1*(2), 67–86.

Kuo, W.-C., & Hsu, T.-C. (2020). Learning computational thinking without a computer: How computational participation happens in a computational thinking board game. *The Asia-Pacific Education Researcher*, *29*(1), 67–83. https://doi.org/10.1007/s40299-019-00479-9

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. https://doi.org/10.1145/1929887.1929902

Lin, C.-H. (2012). An innovative change in technology integration: Training pre-service kindergarten teachers to be courseware designers. *Creative Education*, *3*(7), 1177–1183. https://doi.org/10.4236/ce.2012.37175

Lockwood, J., & Mooney, A. (2018). Computational thinking in secondary education: Where does it fit? A systematic literary review. *International Journal of Computer Science Education in Schools*, *2*(1), 41–60. https://doi.org/10.21585/ijcses.v2i1.26

Looi, C.-K., How, M.-L., Longkai, W., Seow, P., & Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. *Computer Science Education*, *28*(3), 255–279. https://doi.org/10.1080/08993408.2018.1533297

Millwood, R., Bresnihan, N., Walsh, D., & Hooper, J. (2018). *Review of literature on computational thinking. Paper funded by the National Council for Curriculum Assessment 2018. (pp. 10-11)* https://www.ncca.ie/media/3557/primary-coding_review-of-literature-on-computational-thinking.pdf

Muñoz-Repiso, A. G. V., & Caballero-González, Y. A. (2019). Robotics to develop computational thinking in early childhood education. *Comunicar*, *59*, 63–72. https://doi.org/10.3916/C59-2019-06

Oluk, A., & Korkmaz, Ö. (2016). Comparing students' scratch skills with their computational thinking skills in terms of different variables. *Online Submission*, *8*(11), 1–7.

Paf, M., & Dinçer, B. (2021). A study of the relationship between secondary school students' computational thinking skills and creative problem-solving skills. *The Turkish Online Journal of Educational Technology*, *20*(4), 1–15. https://eric.ed.gov/?id=EJ1313467

Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2020). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*, *105*, 105939. https://doi.org/10.1016/j.chb.2019.03.003

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Park, J. (2019). The development and application of computational fairy tales for elementary students. *International Journal of Higher Education*, *8*(3), 159–170. https://doi.org/10.5430/ijhe.v8n3p159

Perlis, A. J. (1964). Programming of digital computers. *Communications of the ACM*, *7*(4), 210–211.

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: Programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In S.-C. Kong & H. Abelson (Eds.), *Computational thinking education* (pp. 79–98). Springer.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, *72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Salvucci, S., Walter, E., Conley, V., Fink, S., & Saba, M. (1997). *Measurement error studies at the National Center for Education Statistics (NCES)*. U.S. Department of Education.

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education. *The Asia-Pacific Education Researcher*, *29*(1), 55–66. https://doi.org/10.1007/s40299-019-00478-w

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*, *111*(5), 764–792. https://doi.org/10.1037/edu0000314

Selby, C., Dorling, M., & Woollard, J. (2014). Evidence of assessing computational thinking. https://eprints.soton.ac.uk/372409/1/372409EvidAssessCT.pdf

Selby, C. C. (2014). *How can the teaching of programming be used to enhance computational thinking skills?* (Doctoral dissertation, University of Southampton). https://eprints.soton.ac.uk/366256/

Selby, C. C. (2015a). Computational thinking blooms. *ITTE Newsletter*, *74*, 6–9. http://itte.org.uk/wp/wp-content/uploads/2016/01/2015-74-1.pdf

Selby, C. C. (2015b). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. In *Proceedings of the workshop in primary and secondary computing education* (pp. 80–87). ACM. https://doi.org/10.1145/2818314.2818315

Selby, C. C., & Woollard, J. (2013). Computational thinking: The developing definition. https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf

Selby, C. C., & Woollard, J. (2014). Refining an understanding of computational thinking. https://eprints.soton.ac.uk/372410/1/372410UnderstdCT.pdf

Snow, E., Rutstein, D., Basu, S., Bienkowski, M., & Everson, H. T. (2019). Leveraging evidence-centered design to develop assessments of computational thinking practices. *International Journal of Testing*, *19*(2), 103–127. https://doi.org/10.1080/15305058.2018.1543311

So, H.-J., Jong, M. S.-Y., & Liu, C.-C. (2020). Computational thinking education in the Asian Pacific region. *The Asia-Pacific Education Researcher*, *29*, 1–8. https://doi.org/10.1007/s40299-019-00494-w

Song, J. B. (2019). The effectiveness of an unplugged coding education system that enables coding education without computers. *Universal Journal of Educational Research*, *10*(1), 129–137. https://doi.org/10.13189/ujer.2019.071514

Tabachnick, B. G., & Fidell, L. S. (2007). *Using multivariate statistics*. Pearson.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, *148*, 103798. https://doi.org/10.1016/j.compedu.2019.103798

Ternik, Ž., Todorovski, L., & Nančovska, Š. I. (2020). Assessing the agreement in the Bebras tasks categorisation. In K. Kori & M. Laanpere (Eds.), *Informatics in schools. Engaging learners in computational thinking (ISSEP 2020) Lecture notes in computer science* (Vol. 12518). Springer. https://doi.org/10.1007/978-3-030-63212-0_3

Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research*, *57*(1), 3–31. https://doi.org/10.1177/0735633117743918

Tsai, M.-J., Wang, C.-Y., & Hsu, P.-F. (2019). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research*, *56*(8), 1345–1360. https://doi.org/10.1177/0735633117746747

Turkish Ministry of National Education. (2018). *Bilişim Teknolojileri ve Yazilim Dersi Öğretim Programi* (*İlkokul 1, 2, 3 ve 4. Sınıflar*) [Information technologies and software course curriculum (1st-4th grade)]. http://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=407

Verma, J. P. (2016). *Repeated measures design for empirical researchers*. Wiley.

Waite, J., Curzon, P., Marsh, W., & Sentance, S. (2016). Abstraction and common classroom activities. In *Proceedings of the 11th workshop in primary and secondary computing education* (pp. 112–113). ACM. https://doi.org/10.1145/2978249.2978272

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies*, *22*(2), 445–468. https://doi.org/10.1007/s10639-016-9493-x

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

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A*, *366*(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Wing, J. M. (2011). Research notebook: Computational thinking-what and why? https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Wong, G. K.-W., & Cheung, H.-Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, *28*(4), 438–450. https://doi.org/10.1080/10494820.2018.1534245

Xing, D., & Lu, C. (2022). Predicting key factors affecting secondary school students' computational thinking skills under the smart classroom environment: Evidence from the science course. *Journal of Baltic Science Education*, *21*(1), 156–170. https://doi.org/10.33225/jbse/22.21.156

Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational thinking as an emerging competence domain. In M. Mulder (Ed.), *Competence-based vocational and professional education* (pp. 1051–1067). Springer. https://doi.org/10.1007/978-3-319-41713-4_49

Yildiz Durak, H., & Sarisepetci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, *116*, 191–202. https://doi.org/10.1016/j.compedu.2017.09.004

Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. In A. Cardoso, G. R. Alves, & M. T. Restivo (Eds.), *2020 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1905–1914). IEEE. https://doi.org/10.1109/EDUCON45650.2020.9125368

Zhan, Z., He, W., Yi, X., & Ma, S. (2022). Effect of unplugged programming teaching aids on children's computational thinking and classroom interaction: With respect to Piaget's four stages theory. *Journal of Educational Computing Research*, *60*(5), 1277–1300.

How to cite this article: 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*, 1–17. https://doi.org/10.1111/jcal.12850