Participatory Educational Research (PER)
Vol.12(4), pp. 268-285, July 2025
Available online at http://www.perjournal.com
ISSN: 2148-6123
http://dx.doi.org/10.17275/per.25.60.12.4

The Effect of Unplugged Coding Activities on 5th Grade Students' Problem-Solving Skills and Attitudes Towards Coding Learning

Elife Öksüz*
Republic of Turkey Ministry of National Education, Konya, Türkiye
ORCID: 0000-0002-1977-6576

Ertuğrul Usta
Department of Computer Education and Instructional Technology, Ahmet Kelesoglu Faculty of Education, Necmettin Erbakan University, Konya, Türkiye
ORCID : 0000-0001-6112-9965

Article history

Received:
19.02.2024

Received in revised form:
10.04.2025

Accepted:
15.05.2025

Key words:
coding; unplugged coding; problem-solving skill; attitude; middle school education

In recent years, research has increasingly focused on developing engaging, developmentally appropriate coding tools, including unplugged coding activities, which have been recommended as an effective starting point for teaching programming and helping children learn coding concepts through practical applications. This experimental study was conducted to reveal the effect of unplugged coding activities on the problem-solving skills and attitudes towards coding education of 5th grade middle school students. The study was designed using a one-group pretest-posttest weak experimental model within the scope of quantitative research methodology. Following the post-test, a structured questionnaire consisting of closed-ended items was administered to all participating students to collect supplementary data supporting the findings. The sample of this study consists of 5th-grade students, including 35 girls and 24 boys, who were enrolled in a public school in the Selçuklu district of Konya, Türkiye during the 2023-2024 academic year. According to the research findings, unplugged coding activities have been observed to have a significant and positive impact on the problem-solving skills and attitudes towards coding education of 5th-grade middle school students. Specifically, it was found that students gained a broader perspective when faced with problem situations and developed a more systematic and organized approach to solving problems. Additionally, students who initially exhibited hesitant and reluctant attitudes towards coding education showed a more positive attitude towards it after the implementation of the coding activities. The structured questionnaire, used as supplementary data, supported the findings from the pretest-posttest comparison, reinforcing positive changes in students' problem-solving skills and attitudes.

Introduction

Yesterday's routine tasks are being transformed by technology today and require essential 21st-century skills such as critical thinking, ethical reasoning, problem-solving, and innovation, which cannot be easily automated as adaptive problems and are expected to be

* Correspondency: elife_oksuz@hotmail.com

developed by our children to succeed in the future (Binkley, Erstad, Herman, Raizen, Ripley & Rumble, 2010; Koenig, 2011; Prensky, 2016; International Society for Technology in Education [ISTE], 2023). Problem-solving, a fundamental 21st-century skill, has become important due to rapid technological advancements and digital transformation (Shadiev & Wang, 2022; Saidov, 2023), and educational activities should focus on teaching problem-solving skills and their application to different problem situations rather than providing solutions to specific problems (Eppe, Gumbsch, Kerzel, Nguyen, Butz & Wermter, 2022). In this context, student-centered learning environments play a critical role in the development of students' problem-solving skills, and coding education stands out as an effective tool for developing these skills (Aytekin, Sönmez Çakır, Yücel & Kulaözü, 2018; Mills, Cope, Scholes & Rowe, 2024).

Coding is known for its complexity and difficulty, making it challenging for both teaching and learning (Hromkovič, 2006; Kaučič & Asič, 2011; Saeli, Perrenet, Jochems & Zwaneveld, 2011). Fessakis, Gouli and Mavroudi (2013) emphasize that learning activities involving programming for young children should be carefully designed to be meaningful, engaging, and achievable to avoid discouraging them. Learning environments with fun and engaging introductory activities for children new to programming can help overcome challenges and learning difficulties, while also motivating them for advanced programming in the future (Papert, 1993; Schwartz, Stagner & Morrison, 2006).

Unplugged coding activities, suitable for all age groups and conducted without computers, are considered a useful method to overcome the complexity of programming syntax (Gal-Ezer & Stephenson, 2009; Tonbuloğlu & Tonbuloğlu, 2019; Mills et al., 2024). Unplugged coding activities provide a hands-on approach for students to understand the fundamentals of computer science and develop problem-solving skills through narratives and scenarios (Bell, Witten & Fellows, 2015). This method dispels the common belief that learning programming is difficult and boring, and increases students' interest in computer science, leading to a more enjoyable and engaging learning experience (Bell, Alexander, Freeman & Grimley, 2009; Kotsopoulos, Floyd, Khan, Namukasa, Somanath, Weber & Yiu, 2017; Sun, Ouyang, Li & Zhu, 2021). It has been observed that individuals who perceive coding as easy to learn show a positive impact on their transition to more complex programming languages, and that coding activities support the development of problem-solving skills, one of the 21st-century competencies (Çetin, 2012; Mills et al., 2024).

Computer science and coding education

A common misconception about computer science is that it's mainly about basic computer skills like web browsing and running software. The real issue lies in viewing computer science solely through its applications, overlooking its critical aspects of problem-solving, creativity, and innovation, which are fundamental to the field (Szlávi & Zsakó, 2006; Hromkovič, 2006). According to this view, coding education is unnecessary since only a few students will become programmers. However, just like math education isn't about making everyone a mathematician, coding education aims to develop thinking and other essential skills. The view that limits information technology education to teaching programming is the exact opposite of the previous misconception, claiming everything it denies. This perspective overlooks the profound impact of information processing on daily life and fails to recognize the need for a new form of computer literacy. In reality, current complex and multifunctional technologies require systematic teaching of computer usage and logical thinking (Szlávi & Zsakó, 2006).

The integration of coding and programming into education has significantly increased over the past two decades, reflecting its importance in the development of critical 21st-century skills (Çakıcı, 2022; Chen, Yang, Metwally, Lavonen & Wang, 2023). This shift is reflected in curriculum revisions and the adoption of various teaching approaches to programming (Aydoğdu, 2019; Chen et al., 2023; Mills et al., 2024). However, coding education varies greatly between countries (Cortina, 2015; Webb, Davis, Bell, Katz, Reynolds, Chambers & Syslo, 2017; Gim, 2021; Ahn, Sung & Black, 2022). Some countries include coding in the primary curriculum, while others introduce it at secondary levels (Calder, 2010). On the other hand, countries like Australia, Estonia, the UK, and Greece have even started teaching programming from preschool (Popat & Starkey, 2019).

Despite the growing global demand for skilled computer experts driven by advancing technologies, statistical trends indicate a decline in the number of students choosing this field (Nascimento, Mendonça, Guerrero & Figueiredo, 2010). This decline can be attributed to the fact that students often perceive coding as a difficult and complex process, primarily due to the complexity of programming syntax (Rodriguez, Rader & Camp, 2016; Tonbuloğlu & Tonbuloğlu, 2019; Sun et al., 2021; Mills et al., 2024). Therefore, Szlávi and Zsakó (2006) emphasize that programming education should prioritize fundamental concepts rather than focusing on specific programming languages. Since algorithms are common in everyday life, understanding these concepts has become increasingly critical in our digitalized world. Furthermore, research by Geçitli and Bumen (2020) highlights that effective conceptualization and organization, along with consideration of students' prior knowledge and interests, are essential for successful programming education.

The implementation of introductory programming activities allows students to see the connections between coding concepts and their applications in various disciplines, which increases their engagement and interest not only in programming courses but also in other courses (Gim, 2021). In addition, selecting algorithmic problems from disciplines such as mathematics and physics during coding education fosters interdisciplinary learning and enhances problem-solving skills (Hromkovič, 2006; Calder, 2010; Saeli et al., 2011; Mills et al., 2024). In summary, Hromkovič (2006) argues that learning the different languages and thinking styles of various fields within a single discipline is one of the most valuable gifts for a computer science student. Moreover, algorithmic thinking, which is considered as the basic step of programming, provides individuals with the skills of planning tasks encountered in daily life, dividing these tasks into subtasks and problem solving (Fessakis et al., 2013; Ziatdinov & Musa, 2013). Learning programming involves mastering a language to effectively communicate with technical systems, which also encourages students to think critically about expressing ideas, thus enhancing their natural language skills (Hromkovič, 2006). Acquiring coding skills at an early age offers children holistic experiences that integrate communication, critical thinking, and problem-solving, which are crucial for success in the 21st-century digital world (McLennan, 2017; Lee & Junoh, 2019; Hollenstein, Thurnheer & Vogt, 2022).

The role of coding in the problem-solving process and the development of skills through unplugged coding activities

The programming process, which involves designing a problem-solving process and structuring it as a computer program, is a complex problem-solving experience that enables students to acquire strategic information to handle problem situations flexibly, while integrating various types of knowledge and technical skills. Thus, the aim of programming

education is not only to teach the use of programming tools, but also to provide real problem-solving experiences throughout the programming process (Nam, Kim & Lee, 2010). Computer programming skills are viewed as a vital and challenging educational field for students, deeply linked to the problem-solving process (Wang, Huang & Hwang, 2014). Research findings show that coding enhances problem-solving skills, creativity, logical reasoning, communication and academic success (Hamada, 1986; Şahin, Şahin & Heppner, 1993; Fessakis et al., 2013; Kalelioğlu & Gülbahar, 2014). In particular, it has been observed that unplugged coding activities, when combined with collaborative learning environments, have a positive impact on students' problem-solving skills (Chen et al., 2023).

There are many activities that encourage children to develop algorithms and support their problem-solving skills without the need for computer assistance (Cortina, 2015; Highfield, Paciga & Donohue, 2018). In particular, board games, card games and story-based activities offer students the opportunity to learn coding concepts in a fun and interactive way (Chen et al., 2023). Moreover, as a low-cost and effective method, these activities can be easily adapted to any society with a little creativity (Cortina, 2015). These unplugged coding activities, conducted without computers, encompass problem-solving processes and introduce key concepts in computer science (Bell et al., 2009). In these activities, a coding process resembling computer programming is used to plan problem-solving steps systematically and represent them through flowcharts. However, instead of using a machine-readable language, the process is carried out concretely with human-readable language (Lee & Junoh, 2019). Thus, children not only acquire coding knowledge, but also effectively develop 21st century skills such as systematic problem solving, discovery, strategic planning, organization, teamwork and active participation in discussions (Kalelioğlu, 2015).

The role of unplugged coding activities in developing attitudes towards coding

In recent years, there has been a growing interest in incorporating unplugged coding activities into computer science education to enhance students' attitudes towards coding (Tonbuloğlu & Tonbuloğlu, 2019; Sun et al., 2021; Chen et al., 2023). Unplugged activities are teaching methods to teach computational concepts without the use of computers, usually through games, puzzles and hands-on activities. These methods aim to present abstract concepts in a more understandable and accessible way, especially for younger students (Threekunprapa & Yasri, 2020).

Numerous studies have demonstrated that unplugged coding activities enhance students' engagement in coding and effectively foster positive attitudes towards it. For instance, in a study conducted by Tonbuloğlu and Tonbuloğlu (2019), it was revealed that middle school students who participated in unplugged coding activities demonstrated high levels of motivation and engagement. Students particularly enjoyed the activities because they were engaging and could be related to real-life situations. Similarly, Rodriguez et al. (2016) demonstrated that unplugged coding activities can enhance students' interest and motivation in computer science, highlighting the inherently engaging nature of these methods. These findings indicate that unplugged coding activities can play a pivotal role in fostering positive attitudes towards coding by offering an interactive and student-centered learning environment.

Furthermore, Sun et al. (2021) found that students in the student-centered unplugged programming group demonstrated higher levels of confidence, enjoyment, and future interest in programming compared to those in traditional lecture-based instruction. These students

exhibited more positive learning behaviors, such as engaging in peer discussions and asking questions. The findings suggest that unplugged coding activities can foster a positive attitude towards coding by providing an interactive and student-centered learning environment.

To summarize, incorporating unplugged coding activities into the curriculum is an effective way to support students' positive attitudes towards coding. These activities can increase students' interest and motivation towards coding by enabling them to experience coding in a more accessible and engaging way. Moreover, this approach may help students develop positive perceptions of computer science and strengthen their desire to pursue further studies in this field.

The research group of this study consists of students in 5th grade (aged 10-11). During the literature review, it was observed that there is a limited number and variety of studies on unplugged coding activities and there is a need for examples of the implementation of these activities in education (Sun et al., 2021). Furthermore, exploring unplugged coding activities that teach 21st-century skills without increasing screen time is believed to be beneficial for both educators and parents concerned about their children's screen usage. In order to fill a gap in the related literature, this study investigates the effects of unplugged coding activities on 5th grade middle school students' problem-solving skills and attitudes towards learning coding.

Method

A quantitative method approach was adopted in the study. Quantitative research is the process of collecting numerical data to be analyzed and testing a specific hypothesis through these data (Creswell, 2003). Quantitative research is based on statistical analysis of numerical data collected using structured data collection tools such as scales, questionnaires and tests, and objective interpretation of the results. In this way, the relationships between the variables of the research are determined, hypotheses are tested and the findings obtained can be generalized (Büyüköztürk, Kılıç Çakmak, Akgün, Karadeniz & Demirel, 2012). In order to obtain the research data, one group pretest-posttest model, which is one of the experimental designs, was adopted as the research model. The study was conducted using the one-group pretest-posttest model due to the insufficient number of students. In the one-group pretest-posttest experimental design, the same measurement tools are applied twice to a single group, before and after the intervention, to assess the effect of the application on the dependent variable (Karasar, 2011; Büyüköztürk et al., 2012). The dependent variables of this research are problem-solving skills and attitudes towards learning coding, while the independent variable is unplugged coding activities.

Three scales were used to collect the data from the study. Before the experimental process, the participants were administered the “Problem Solving Scale” and the “Attitude Scale for Coding Education” as pretests to assess their problem-solving skills and attitudes towards coding education. After these pre-tests, the students participated in a 10-week training program based on unplugged coding activities. After the completion of the training program, the same participants were again administered the “Problem Solving Scale” and the “Attitude Scale for Coding Education” as post-tests. In addition, the “Student Opinion Form on the Learning Process” was administered to the students along with the post-test and data on the learning process were collected.

Participants

The sample group of the study consists of 35 female (59.3%) and 24 male (40.7%) students, a total of 59 students, who are attending a public middle school and are in the 5th grade. The students participating in the study form a heterogeneous group with different levels of learning and achievement. In selecting the study group for this research, a combination of purposive sampling and convenient sampling methods was used. Through purposive sampling, facts, events and situations that are thought to provide a large number of in-depth data can be examined and understood in detail (Yıldırım & Şimşek, 2013). The reason for using the purposive sampling method in this study is the requirement that participants have not received prior training in areas such as algorithms and coding, as well as the fact that coding courses are being taught for the first time as a mandatory subject in the 5th grade curriculum. Moreover, the convenient sampling method was applied as the researcher (1st author) conducted the study with 5th grade students who volunteered from the school where they are employed. Therefore, a more effective sample was planned by targeting individuals or groups that are both easily accessible and compatible with the research purpose.

Data acquisition tools

Problem solving scale

The original scale developed by Heppner & Peterson (1982) for adults was translated into Turkish by Şahin et al. (1993). The scale used in the study is the re-adapted version for 5th grade students, based on the Turkish adaptation by Kardaş, Anagün & Yalçınoğlu (2014), after conducting validity and reliability studies. The validity and reliability of the developed scale were conducted with 285 students attending primary schools in Turkey. Confirmatory Factor Analysis (CFA) was conducted to determine the factor structures of the scale, and it was found to be statistically appropriate. When the subdimensions of the scale were examined, three subfactors were identified: confidence in problem-solving ability, approach-avoidance and personal control. Confidence in problem-solving ability reflects the student's trust in their problem-solving skills; approach-avoidance reflects their approach to overcoming challenging problems; and personal control reflects the student's sense of personal control over the situation. The scale is a four-point Likert-type rating scale and consists of 20 items. The score range of the problem-solving inventory is between 20 and 80. Items 1, 2, 7, 14, 15 and 18 in the scale are scored in reverse order, with 1=4, 2=3, 3=2, and 4=1 (Kardaş, 2013). The Cronbach's Alpha reliability of the Problem-Solving Scale was calculated, and it was indicated that the reliability coefficient was found to be 0.74. According to the research findings, it has been reported that the Problem-Solving Scale is a valid and reliable scale with linguistic equivalence.

After the implementation of unplugged coding activities, the internal reliability coefficient was calculated using the data collected from the research group regarding the Problem-Solving Scale. In the conducted reliability test, the Cronbach's Alpha reliability coefficient for the 20-item Problem Solving Scale was found to be .617. This value is considered acceptable for educational research but is regarded as relatively low reliability (Karasar, 2011).

Attitude scale for coding education (ASCE)

The Attitude Scale for Coding Education (ASCE), developed by Karaman and Büyükalan Filiz (2019), was used to measure students' attitudes towards coding. The scale is a

five-point Likert-type instrument consisting of 41 items, designed to assess students' attitudes towards coding education. The validity and reliability of the Attitude Scale for Coding Education were established through a rigorous statistical analysis process conducted with a sample of 503 middle school students (247 boys and 256 girls) from five different schools during the 2018–2019 academic year. Exploratory Factor Analysis was applied to the pre-test data, and based on the statistical significance of the data, Confirmatory Factor Analysis was conducted. It has been indicated that the Kaiser-Meyer-Olkin (KMO) value is 0.96, which meets an acceptable level of adequacy for the data. As a result of the analyses, the variance value for the two-factor structure of the 41 items was determined to be 47.89%. It has been indicated that the ratio of the chi-square value obtained from the scale to the degrees of freedom is below 2.5, indicating an excellent level and this finding supports the factor structure of the data set. The reliability of the Attitude Scale for Coding Education was assessed using Cronbach's Alpha, which yielded a coefficient of .960, indicating a high level of internal consistency. These results demonstrate that the scale is a valid and reliable instrument for measuring middle school students' attitudes towards coding education.

After the implementation of unplugged coding activities, the internal reliability coefficient was calculated using the data collected from the research group regarding the Attitude Scale for Coding Education. The reliability analysis revealed a Cronbach's Alpha coefficient of .973 for the 41-item scale, indicating a high level of reliability.

Student opinion form on the learning process

Quantitative data were collected from the study group to examine the impact of unplugged coding activities on students' problem-solving skills and attitudes towards coding education. These data were intended to support and complement the findings obtained from the Problem-Solving Scale and the Attitude Scale for Coding Education. The data were collected through a structured questionnaire titled "Student Opinion Form on the Learning Process," developed by the researcher (1st author). The form was developed based on expert opinions to ensure content and face validity. Initially, a draft version of the form was prepared and sent to four different experts. The form was revised in line with the opinions of the experts, and the form was finalized with a total of 12 closed-ended items (9 positive and 3 negative).

Procedure for implementation

A 10-week application process was followed in the research to determine the effects of unplugged coding activities on students' problem-solving skills and their attitudes towards coding. Before the application, since the students participating in the research had no prior knowledge of coding and would be taking coding lessons for the first time, the researcher (1st author) provided the students with information about the meaning of coding and coding activities. After a brief introduction, the “Problem Solving Scale” and the “Attitude Scale for Coding Education” were administered to the students as a pre-test, and the transition to unplugged coding activities was made from the 2nd week.

The teaching process was structured based on the activities in the 5th grade Information Technologies and Software Course Teacher's Guide developed in cooperation with the Ministry of National Education and Google (Kesf@ Projesi, 2018). The activities were implemented in accordance with the annual and weekly lesson plans, with two class hours per week (each lasting 40 minutes). Throughout the process, a constructivist learning approach was adopted, and student-centered teaching methods such as collaborative learning, group

discussions, problem solving, peer learning, peer assessment, game-based learning, classroom guidance based on reflective thinking after activities were used. Each weekly lesson began with a meaningful problem situation that students could relate to in their daily lives, continued with hands-on tasks, and concluded with student presentations. In the second week, within the framework of this approach, strategies such as trial-and-error, strategic thinking, logical inference, and applying solutions to simpler similar problems were used prominently during problem-based activities. For example, in the "Towers of Hanoi Puzzle" activity, students attempted to solve the three-disc version by building upon the solution strategies they had previously used with the two-disc version. In the third week, the "What Should I Do Now?" game and the fishbone diagram activity were utilized to develop brainstorming skills, cause-effect reasoning, peer discussions, and problem-solving strategies through the use of visual organizers. During the fourth week, students followed written and verbal instructions through activities such as paper folding and giving directions, enhancing their skills in understanding, generating, and sequencing instructions, as well as error analysis. These activities required students to identify the necessary steps to complete a task, plan the process, and create instructions in a logical order. The fifth week focused on real-life context–based data collection and interpretation activities. Students engaged in tasks such as baking a cake and game analysis to practice classifying and associating data. In the sixth week, activities aimed at improving pattern recognition, estimation, and logical reasoning were implemented to enhance students' thinking skills. Between the seventh and ninth weeks, core coding skills such as algorithmic thinking, sequencing, direction setting, and creating flowcharts were reinforced through game-based learning. During these weeks, concept reinforcement, teamwork, peer learning, and task distribution within groups were emphasized. Activities such as "Robot's Route" and "Tospaa Coding Game" served to consolidate the skills students had acquired in the previous weeks.

After the unplugged coding activities, the "Problem Solving Scale", the "Attitude Scale for Coding Education" and the "Student Opinion Form on the Learning Process" were applied as post-tests in the 10th week.

The implementation process of the research and the unplugged coding activities are summarized in Table 1.

Table 1 The Implementation Process of the Research and the Unplugged Coding Activities

Week	Subject/Process	Implementation activities
Week 1	Pre-test Application with Data Collection Tools	Problem Solving Scale, Attitude Scale for Coding Education
Week 2	Puzzle Challenge	The Wolf, The Lamb, and The Grass Problem The Towers of Hanoi Puzzle
Week 3	I Found the Best Solution	What Should I Do Now Game Fishbone Worksheet
Week 4	Follow the Instructions	Paper Folding Activity Giving Directions Activity
Week 5	Give Me the Data	Cake Making Activity Game Analysis Study
Week 6	I Think Logically	Number Guessing Activity Colors of Nature Worksheet
Week 7	A Tale of Algorithms	Mix-up Game Activity

Week 8	I Change the Flow	Robot's Route Activity Oh No! The Flowcharts are Mixed Up!
Week 9	I Think Algorithmically	Tospaa, Coding Game without Computer
Week 10	Post-test Application with Data Collection Tools	Problem Solving Scale, Attitude Scale for Coding Education, Student Opinion Form on the Learning Process

Data analysis

In this study, the data analysis process was carried out using the Statistical Package for Social Sciences (SPSS) 22 statistical program. First of all, the information on the demographic characteristics of the students who participated in the study and their responses to the measurement tools were transferred to a digital environment after verifying their validity. The demographic data of the participants were analyzed using descriptive statistics, including arithmetic mean, standard deviation, percentage, and frequency values. In addition, a paired-samples t-test was used to determine whether there was a significant difference between the pre-test and post-test scores of the study group on the Attitude Scale for Coding Education and the Problem-Solving Scale. According to Köklü, Büyüköztürk, and Çokluk Bökeoğlu (2007, pp. 167–168), the paired-samples t-test is used to assess whether there is a statistically significant difference between the means of two related groups. In this context, related groups mean that the same participants are present in both groups, and each participant is measured on the same dependent variable under two different conditions. The reason for applying the paired-samples t-test in the study is that it is a parametric statistical technique, which requires that the distributions show normality in both measurements, the pre-test and post-test scores of a single group are compared, and the variances are homogeneous. The two most commonly used tests for testing normality are the Kolmogorov-Smirnov test and the Shapiro-Wilk test. Since our sample size is greater than 30 ($n > 30$), the normality of the distribution of the quantitative variable was tested using the Kolmogorov-Smirnov test. The result was $p > 0.05$, indicating that the variable conforms to a normal distribution. Accordingly, inferential statistical analyses were conducted using parametric tests.

Results

This study was conducted to examine the effect of unplugged coding activities on 5th grade students' problem-solving skills and attitudes towards coding. In this direction, “Problem Solving Scale” and “Attitude Scale Towards Coding Education” scales were applied to the students as pre-test and post-test.

Table 2 presents the descriptive statistics of the pre-test and post-test results obtained from the Problem-Solving Scale, along with the results of the dependent samples t-test used to assess whether the difference between the scores is statistically significant.

Table 2 Descriptive Statistics and Dependent Samples t-Test Results for Pre-test and Post-test Scores on the Problem-Solving Scale

Test Type	N	Min.	Max.	$\bar{X}$	SD (Standard deviation)	df (Degrees of freedom)	t	p
Pre-test Findings	59	38.00	75.00	54.27	6.71	58	-3.79	.000*
Post-test Findings	59	36.00	75.00	58.13	8.52			

*p<0.05

As shown in Table 2, the pre-test results of the Problem-Solving Scale indicated that students had an arithmetic mean of $\bar{X}$ = 54.27 and a standard deviation of SD = 6.71 before the unplugged coding activities. In the post-test, the mean score increased to $\bar{X}$ = 58.13, with a standard deviation of SD = 8.52. According to the dependent samples t-test, this increase was statistically significant [t(58) = -3.79, p <.05], favoring the post-test results. These findings suggest that the unplugged coding activities implemented in the study had a positive effect on students' problem-solving skills.
Table 3 presents the descriptive statistics of the pre-test and post-test results obtained from the Attitude Scale Towards Coding Education, along with the results of the dependent samples t-test used to assess whether the difference between the scores is statistically significant.

Table 3 Descriptive Statistics and Dependent Samples t-Test Results for Pre-test and Post-test Scores on the Attitude Scale Towards Coding Education

Test Type	N	Min.	Max.	$\bar{X}$	SD (Standard deviation)	df (Degrees of freedom)	t	p
Pre-test Findings	59	52.00	167.00	117.30	36.62	58	-7.17	.000*
Post-test Findings	59	64.00	200.00	154.83	26.40			

*p<0.05

As shown in Table 3, the pre-test results of the Attitude Scale Towards Coding Education indicated that students had an arithmetic mean of $\bar{X}$ = 117.30 and a standard deviation of SD = 36.62 before the unplugged coding activities. In the post-test, the arithmetic mean increased to $\bar{X}$ = 154.83, with a standard deviation of SD = 26.40. According to the results of the dependent samples t-test, this increase was statistically significant [t(58) = -7.17, p <.05], favoring the post-test scores. These findings suggest that while students initially had an ambivalent attitude towards coding education, their attitudes became significantly more positive following the implementation of the unplugged coding activities. In summary, the findings indicate that the unplugged coding activities implemented in the study were effective in enhancing students' attitudes towards coding education.

To support and complement the findings obtained from the Problem Solving Scale and the Attitude Scale for Coding Education, a structured questionnaire was administered to all students following the post-tests. This questionnaire provided quantitative data on students' perceptions regarding the effectiveness of unplugged coding activities in the learning process. The percentage and frequency values of the students' responses are presented in Table 4.

Table 4 Student Opinions on the Learning Process

"Student Opinion Form on the Learning Process" Data	Yes/No	f	%
1. I feel happy with the unplugged coding activities.	Yes	55	93,2
	No	4	6,8
2. Unplugged coding activities contribute to the development of my problem-solving skills.	Yes	56	94,9
	No	3	5,1
3. Unplugged coding activities spark my curiosity and capture my interest.	Yes	51	86,4
	No	8	13,6
4. Unplugged coding activities make the lessons enjoyable.	Yes	55	93,2
	No	4	6,8
5. Unplugged coding activities contribute to my academic success.	Yes	51	86,4
	No	8	13,6
6. There are times when I get confused during the unplugged coding activities.	Yes	21	35,6
	No	38	64,4
7. I would like to do more unplugged coding activities to learn coding.	Yes	46	78,0
	No	13	22,0
8. I would like to improve what I have learned in this course and design my own games and animations in the future.	Yes	48	81,4
	No	11	18,6
9. It is difficult for me to solve problems through scenarios in this course.	Yes	12	20,3
	No	47	79,7
10. There are times when I feel inadequate while doing unplugged coding activities.	Yes	24	40,7
	No	35	59,3
11. I would like to take this course again.	Yes	53	89,8
	No	6	10,2
12. Unplugged coding activities increase my interest in coding.	Yes	55	93,2
	No	4	6,8

As shown in Table 4, students responded with a very high level of 'Yes' to 9 out of 12 items measuring positive opinions, while they answered 'No' at a level above the moderate to the 3 items measuring negative opinions. It was observed that 94.9% of the students answered "Yes" to the statement "Unplugged coding activities contribute to the development of my problem-solving skills."; 79.7% of the students answered "No" to the statement "It is difficult for me to solve problems through scenarios in this course."; 93.2% of the students answered "Yes" to the statements "I feel happy with the unplugged coding activities.", "Unplugged coding activities make the lessons enjoyable." and "Unplugged coding activities increase my interest in coding."

The findings show that unplugged coding activities have the potential to have a positive impact on students' problem-solving skills and attitudes towards coding education.

Discussion

The main aim of this study is to investigate the effect of unplugged coding activities on the problem-solving skills and attitudes towards coding education of 5th grade middle school students. The results of the study indicate that unplugged coding activities may have a positive effect on the participants' problem-solving skills and attitudes towards coding education. However, since there was no control group in this study, the possibility that the observed changes may be affected by other uncontrolled variables should be taken into account.

A statistically significant difference in favor of the post-test was found between the pre-test and post-test scores of the "Problem-Solving Scale" for the participating children. Additionally, the quantitative results obtained from the "Student Opinion Form on the Learning Process" were found to support these findings. Based on these results, it can be stated that unplugged coding activities have the potential to enhance children's problem-solving skills.

A review of the relevant literature reveals that similar findings have been reported in other studies. Tağci (2019) the effects of coding education on primary school students were examined, incorporating both unplugged coding activities and computer-based coding activities. The research demonstrated that these activities were effective in developing students' problem-solving, critical thinking, collaborative learning, and logical reasoning skills. Kim, Kim and Kim (2014) found in their study that students who participated in unplugged coding activities exhibited higher logical thinking skills compared to students who received programming training with Logo. Additionally, it has been stated that these activities increased students' willingness to learn computer science and helped them acquire skills such as mapping, diagramming, and problem-solving. Aydoğdu (2019), in his study examining 6th grade students' algorithmic thinking skills during unplugged coding activities, explained that students became more detailed and organized in the strategies they developed for problem situations and showed progress in strategy development. Küçükkara and Aksüt (2021), in their study with preschool students, reported that all students successfully completed the problem situations presented using algorithmic skills and did not struggle while doing so.

On the other hand, Çakıcı and Özdemir (2022), in their study examining the effects of unplugged coding activities on primary school students, stated that the activities improved students' problem-solving skills, although some students initially faced difficulties, and overcame these challenges in subsequent activities to become successful. The study highlighted specific areas of difficulty, such as identifying directions, problem-solving, and applying the steps. This situation may be due to students not having had enough practice, not having received prior coding education, or the fact that the students participating in the study were 3rd-grade primary school students. Rodriguez et al. (2016) stated that students need to practice individually in order to fully grasp a concept. In coding education, considering the complex nature of programming, it is important to select activities that are appropriate for students' developmental characteristics (Kert & Uğraş, 2009). Additionally, special attention should be given to introductory coding activities for students who are starting to code in elementary schools (Papert, 1993; Schwartz et al., 2006; Kaučič & Asič, 2011; Saeli et al., 2011).

A statistically significant difference was observed between the pre-test and post-test scores of the students who participated in the study on the “Attitude scale for coding education” with the post-test scores being higher. Additionally, the quantitative results obtained from the

"Student Opinion Form on the Learning Process" were found to support these findings. Based on these results, it can be stated that unplugged coding activities have the potential to improve children's attitudes towards coding education.

Çelik Kırçalı (2019) stated that unplugged coding activities, with their engaging scenarios, draw students in, allow them to be active in the learning process through the activities, and provide the excitement of encountering new activities in each lesson, making them as interesting as computer-based tools. Similarly, Rodriguez et al. (2016) highlighted that B3 (Bilgisayarsız Bilgisayar Bilimi - Computer Science Unplugged) activities, characterized as engaging and hands-on learning experiences without digital devices, significantly increased students' interest and motivation in computer science.

Küçükkara and Aksüt (2021) stated in their research with preschool students on unplugged coding activities that the students enjoyed the activities and expressed a desire to engage in more coding activities. Secer (2020) stated that students initially exhibited reluctant attitudes towards unplugged coding lessons due to their past experiences, but paper-and-pencil coding activities brought about a positive change in their attitudes towards the lesson and made them more willing to develop a new game using coding. Çakıcı (2022) found in his study with elementary school students that they enjoyed participating in unplugged coding activities at a high level, believed that offering coding activities as a subject in elementary school would be beneficial, and stated that they would continue to engage in unplugged coding activities by supporting the knowledge they had learned through different channels in the future. Other studies have also reported that students found unplugged coding activities enjoyable and beneficial, and that these activities had a positive impact on their attitudes towards the lesson (Olmo-Muñoz, Cózar-Gutiérrez & González-Calero, 2020; Çimşir, 2019; Aydoğdu, 2019; Sun et al., 2021; Chen et al., 2023).

In conclusion, unplugged coding activities have the potential to enhance students' problem-solving skills and their attitudes towards coding education. Moreover, using unplugged coding activities as a part of basic programming education or integrating them as a complementary approach can potentially enrich students' learning experiences and outcomes. However, since this study did not include a control group, the possibility that the observed changes may be influenced by other uncontrolled variables should be considered. Particularly in situations where the lack of technological resources in schools makes computer-based coding tools inaccessible, unplugged coding activities may serve as an effective alternative and could be beneficial for achieving certain programming learning outcomes.

Finally, this study investigated the impact of unplugged coding activities on problem-solving skills. The effect of unplugged coding activities on various skills, such as collaborative work, mathematical thinking, creative thinking, and debugging, can also be examined. This study was conducted using a one-group pre-test-post-test design due to the insufficient number of students. It may be suggested that future experimental studies include a control group.

Declarations

Acknowledgments: We would like to thank the students who participated in this study and everyone who made our research possible.

This article is derived from the master's thesis of the first author.

Funding: The authors declare no financial support was received for this research.

Ethics Statements: This study was approved by the Ethics Committee of the Scientific Research Ethics Board of the Faculty of Social and Human Sciences at Necmettin Erbakan University, and all participants provided informed consent prior to their participation. The privacy and confidentiality of participants were maintained throughout the research process.

Conflict of Interest: They have no conflict of interest.

Informed Consent: This study was approved by the Ethics Committee of the Scientific Research Ethics Board of the Faculty of Social and Human Sciences at Necmettin Erbakan University. All participants were informed about the purpose, procedures, and potential risks of the study before participating. They voluntarily provided their written consent to take part in the research, and their privacy and confidentiality were maintained throughout the study.

Data availability: The data supporting the findings of this study are openly available and can be accessed freely through the journal's open-access platform.

References

Ahn, J., Sung, W., & Black, J. B. (2022). Unplugged debugging activities for developing young learners' debugging skills. *Journal of Research in Childhood Education, 36*(3), 421-437. https://doi.org/10.1080/02568543.2021.1981503

Aydoğdu, E. (2019). Investigation of students' algorithmic thinking skills in the process of unplugged programming activities (Master's thesis). Trabzon University, Trabzon.

Aytekin, A., Sönmez Çakır, F., Yücel, Y. B., & Kulaözü, İ. (2018). Coding science directed to future and some methods to be available and coding learned. *Eurasian Journal of Researches in Social and Economics (EJRSE), 5*(5), 24–41.

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.

Bell, T., Witten, I. H., & Fellows, M. (2015). *Computer Science Unplugged: An enrichment and extension programme for primary-aged students* (S. Jarman, Ed.; R. Adams & J. McKenzie, Adapt.). Online: self-published.

Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., & Rumble, M. (2010). *Defining 21st century skills* (Draft White Paper 1). Assessment and Teaching of 21st Century Skills (ATC21S), University of Melbourne.

Büyüköztürk, Ş., Kılıç Çakmak, E., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2012). *Bilimsel araştırma yöntemleri* [Scientific research methods]. Ankara: Pegem Academy.

Calder, N. (2010). Using Scratch: an integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, *15*(4), 9- 14.

Chen, P., Yang, D., Metwally, A. H. S., Lavonen, J., & Wang, X. (2023). Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. *International Journal of STEM Education, 10*(1), 47. https://doi.org/10.1186/s40594-023-00434-7

Cortina, T. J. (2015). Reaching a broader population of students through unplugged activities. *Communications of the ACM, 58*(3), 25-27.

Creswell, J. W. (2003). *Research design: Qualitative, quantitative, and mixed methods approaches* (2nd ed.). Thousand Oaks, CA: Sage

Çakıcı, Y. (2022). *The effect of unplugged coding education on primary school students' gather attention, problem solving and algorithmic thinking skills* (Unpublished master's thesis). Mersin University, Mersin.

Çakıcı, Y., & Özdemir, S. M. (2022). The effect of unplugged coding education on primary school students' gather attention, problem solving and algorithmic thinking skills. *International Journal of Science and Education, 5*(3), 235-254.

Çelik Kırçalı, A. (2019). *The evaluation of the plugged and unplugged tools used in algorithm teaching at K12 level in terms of various variables* (Unpublished master's thesis). Marmara University, İstanbul.

Çetin, E. (2012). *The effect of computer programming education on children's problem-solving skills* (Unpublished master's thesis). Gazi University, Ankara.

Çimşir, E. (2019). *Effect of computer science unplugged activities on students' academic achievements in programming teaching* (Unpublished master's thesis). Ondokuz Mayıs University, Samsun.

Eppe, M., Gumbsch, C., Kerzel, M., Nguyen, P. D., Butz, M. V., & Wermter, S. (2022). Intelligent problem-solving as integrated hierarchical reinforcement learning. *Nature Machine Intelligence, 4*(1), 11-20.

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63*, 87–97. https://doi.org/10.1016/j.compedu.2012.11.016

Gal-Ezer, J., & Stephenson, C. (2009). The current state of computer science in US high schools: A report from two national surveys. *Journal for Computing Teachers, 1*(13), 1-5.

Geçitli, E., & Bumen, N. (2020). Text-based programming instruction through tiered activity: a collaborative action research. *Asian Journal of Instruction*, *8*(2), 64-85. https://doi.org/10.47215/aji.791967

Gim, N. G. (2021). Development of life skills program for primary school students: Focus on entry programming. *Computers, 10*(5), 56. https://doi.org/10.3390/computers10050056

Hamada, R. M. (1986). *The Relationship between Learning Logo and Proficiency in Mathematics* [Unpublished doctoral dissertation], Colombia: Columbia University.

Heppner, P. P., & Peterson, C. H. (1982). The development and implications of a personal problem-solving inventory. *Journal of Counseling Psychology, 29* (1), 66–75. https://doi.org/10.1037/0022-0167.29.1.66

Highfield, K., Paciga, K. A., & Donohue, C. (2018). Supporting whole child development in the digital age. In S. J. Danby, M. Fleer, C. Davidson, M. Hatzigianni (Eds.), *Digital childhoods: International Perspectives on Early Childhood Education and Development, 22* (pp. 165-182). Singapore: Springer https://doi.org/10.1007/978-981-10-6484-5_11

Hollenstein, L., Thurnheer, S., & Vogt, F. (2022). Problem solving and digital transformation: Acquiring skills through pretend play in kindergarten. *Education Sciences, 12*(2), 92. https://doi.org/10.3390/educsci12020092

Hromkovič, J. (2006). Contributing to general education by teaching informatics. In R. T. Mittermeir (Ed.), *Informatics education - the bridge between using and understanding computers* (pp. 25-37). Berlin: Springer.

International Society for Technology in Education [ISTE] (2023). *ISTE standards for students.* Retrieved from https://www.iste.org/standards/iste-standards-for-students

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior, 52*(1), 200-210. https://doi.org/10.1016/j.chb.2015.05.047

Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education, 13*(1), 33-50. https://doi.org/10.15388/infedu.2014.03

Karaman, U., & Büyükalan Filiz, S. (2019). Development of the attitude scale for coding education (ASCE). *Future Visions Journal, 3*(2), 36-47. doi: 10.29345/futvis.80

Karasar, N. (2011). *Bilimsel araştırma yöntemi* [Scientific research method]. Ankara: Nobel Publishing.

Kardaş, N., Anagün, Ş. S., & Yalçınoğlu, P. (2014). Adaptation of problem solving inventory for the elementary school students: Confirmatory factor analysis. *Electronic Journal of Social Sciences, 13*(51), 182-194.

Kardaş, N. (2013). *The impact of argumentation focused teaching in science education on the decision making and problem solving skills of students* (Unpublished master's thesis). Eskişehir Osman Gazi University, Eskişehir.

Kaučič, B., & Asič, T. (2011, May). Improving introductory programming with Scratch?. In *MIPRO, 2011 Proceedings of the 34th International Convention*, 1095-1100.

Kert, S. B., & Uğraş, T. (2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneği [Simplicity and fun in programming education: Scratch example]. In *The First International Congress of Educational Research*. Çanakkale, Türkiye.

Kesf@ Projesi. (2018). *5. sınıf bilişim teknolojileri ve yazılım dersi öğretmen rehberi* [Grade 5 information technologies and software course teacher's guide]. Retrieved from http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-ve-yazilim-dersi-ogretmen-rehberi.pdf

Kim, B., Kim, T., & Kim, J. (2014). 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

Koenig, J. A. (2011). Assessing 21st century skills: Summary of a workshop. Washington, DC: The National Academies Press. https://doi.org/10.17226/13215

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education, 3*(2), 154-171. https://doi. org/10.1007/s40751-017-0031-2

Köklü, N., Büyüköztürk, Ş., & Çokluk Bökeoğlu, Ö. (2007). *Sosyal bilimler için istatistik* [Statistics for social sciences] (2nd ed.). Ankara: Pegem Academy.

Küçükkara, M. F., & Aksüt, P. (2021). Okul öncesi dönemde bilgisayarsız kodlama eğitimine bir örnek: Problem çözme becerileri için etkinlik temelli algoritma [An example of unplugged coding activities in preschool period: Activity-based algorithm for problem solving skills]. *Journal of Inquiry Based Activities, 11*(2), 81–91.

Lee, J., & Junoh, J. (2019). Implementing unplugged coding activities in early childhood classrooms. *Early Childhood Education Journal, 47*(6), 709-716.

Mclennan, D. P. (2017). Creating coding stories and games. *Teaching Young Children, 10* (3), 18-21.

Mills, K. A., Cope, J., Scholes, L., & Rowe, L. (2024). Coding and computational thinking across the curriculum: A review of educational outcomes. *Review of Educational Research.* https://doi.org/10.3102/00346543241241327

Nam, D., Kim, Y., & Lee, T. (2010). The effects of scaffolding-based courseware for the Scratch programming learning on student problem solving skills. Paper presented at the *18th International Conference on Computers in Education*, Putrajaya, Malaysia.

Nascimento, M. R., Mendonça, A. P., Guerrero, D. D. S., & Figueiredo, J. C. A. (2010, October). Teaching programming for high school students: A distance education experience. Paper presented at the *40th ASEE/IEEE Frontiers in Education Conference*, Washington, DC.

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

Papert, S. (1993). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, *128*, 365-376. https://doi.org/10.1016/j.compedu.2018.10.005

Prensky, M. (2016). *Education to better their world: Unleashing the power of 21st-century kids.* New York: Teachers College Press.

Rodriguez, B., Rader, C., & Camp, T. (2016). Using student performance to assess CS unplugged activities in a classroom environment. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 95-100. https://doi.org/10.1145/2899415.2899465

Saeli, M., Perrenet, J., Jochems, W. M. G., & Zwaneveld, B. (2011). Teaching programming in secondary school: A pedagogical content knowledge perspective. *Informatics in Education, 10*(1), 73–88. https://doi.org/10.15388/infedu.2011.06

Saidov, A. (2023, April 26). *Skills-first approach: How cutting-edge skills data can help build an adaptable, resilient workforce?* World Economic Forum. https://www.weforum.org/agenda/2023/04/skills-workforce-data/

Schwartz, J., Stagner, J., & Morrison, W. (2006). Kid's Programming Language (KPL). *The 33rd International Conference and Exhibition on Computer Graphics and Interactive Techniques.* Boston, Massachusetts.

Secer, M. (2020). *The effects of arduino based coding and paper-pencil based coding activities on students' computational thinking skills, problem solving skills and STEM attitudes in information technologies and software lesson* (Unpublished Doctoral Thesis). Mersin University, Mersin.

Shadiev, R., & Wang, X. (2022). A review of research on technology-supported language learning and 21st century skills. *Frontiers in Psychology, 13*, 897689. https://doi.org/10.3389/fpsyg.2022.897689

Sun, D., Ouyang, F., Li, Y., & Zhu, C. (2021). Comparing learners' knowledge, behaviors, and attitudes between two instructional modes of computer programming in secondary education. *International Journal of STEM Education, 8*(54). https://doi.org/10.1186/s40594-021-00311-1

Szlávi, P., & Zsakó, L. (2006, November). Programming versus application. In R. T. Mittermeir (Ed.), *Informatics education – the bridge between using and understanding computers: 2nd International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (ISSEP 2006), Proceedings* (Lecture Notes in Computer Science, Vol. 4226, pp. 48–58). Berlin, Heidelberg: Springer.

Şahin, N., Şahin, N. H., & Heppner, P. P. (1993). Psychometric properties of the problem solving inventory in a group of Turkish university students. *Cognitive Therapy and Research, 17*(4), 379-396. doi: 10.1007/BF01177661

Tağci, Ç. (2019). *Research of effects of coding education on primary school students* (Unpublished doctoral dissertation). Afyon Kocatepe University.

Threekunprapa, A., & Yasri, P. (2020). Unplugged coding using flowblocks for promoting computational thinking and programming among secondary school students. *International Journal of Instruction, 13*(3), 207–222. https://doi.org/10.29333/iji.2020.13314a

Tonbuloğlu, B., & Tonbuloğlu, İ. (2019). The effect of unplugged coding activities on computational thinking skills of middle school students. *Informatics in Education, 18*(2), 403–426. https://doi.org/10.15388/infedu.2019.19

Wang, H. Y., Huang, I., & Hwang, G. J. (2014). Effects of an integrated Scratch and project-based learning approach on the learning achievements of gifted students in computer courses. In *2014 IIAI 3rd International Conference on Advanced Applied Informatics* (pp. 382–387). IEEE. https://doi.org/10.1109/IIAI-AAI.2014.85

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

Yıldırım, A., & Şimşek, H. (2013). *Sosyal bilimlerde nitel araştırma yöntemleri* [Qualitative research methods in social sciences]. Ankara: Seçkin Publishing.

Ziatdinov, R., & Musa, S. (2013). Rapid mental computation system as a tool for algorithmic thinking of elementary school students development. *European Research, 25*(7), 1105-1110.