CJLT RCAT
Canadian Journal of Learning and Technology
La Revue canadienne de l'apprentissage et de la technologie

Volume 50 (2) Summer / été 2024

Effects of Robotic Coding on Computational Thinking Skills of Secondary School Students

Effets du codage robotique sur les compétences de pensée computationnelle des élèves du secondaire

***Serhat Altıok**, Kırıkkale University, Turkey*

***Memet Üçgül**, Kırıkkale University, Turkey*

Abstract

In recent years, computational thinking has garnered increased attention as an essential problem-solving skill. One of the methods to develop students' computational thinking skills is robotic coding activities. This study sought to investigate the impact of robotic coding activities on the self-efficacy perceptions of secondary school students' computational thinking skills. A one-group pretest-posttest quasi-experimental design was employed, involving 32 secondary school students. These students, organized in groups of four, engaged in hands-on robotic coding activities using Lego Mindstorms EV3 Education robots over a total of 20 hours. Data were collected before and after the robotic coding activities using the Self-Efficacy Perception Scale for Computational Thinking Skills (SEPSCTS) instrument, comprising 36 items categorized into five factors. The data were analyzed using paired samples *t*-tests and analysis of covariance (ANCOVA). The results demonstrated a significant increase in students' self-efficacy perceptions of computational thinking skills following the activities, with this increase observed consistently across genders. Finally, the challenges encountered during research and practice were reported, along with the study's limitations, to inform future research endeavours.

Keywords: computational thinking, perception, robotic coding, self-efficacy

Résumé

Ces dernières années, la pensée computationnelle a fait l'objet d'une attention accrue en tant que compétence essentielle pour la résolution de problèmes. L'une des méthodes pour développer les compétences des élèves en matière de pensée computationnelle est le codage robotique. Cette étude visait à examiner l'impact des activités de codage robotique sur les perceptions d'auto-efficacité des compétences de pensée computationnelle des élèves du secondaire. Un modèle quasi expérimental

prétest-post-test à groupe unique a été utilisé, impliquant 32 élèves du secondaire. Ces élèves, organisés en groupes de quatre, ont participé à des activités pratiques de codage robotique en utilisant des robots *Lego Mindstorms EV3 Education* pendant un total de 20 heures. Les données ont été collectées avant et après les activités de codage robotique à l'aide de l'instrument *Self-Efficacy Perception Scale for Computational Thinking Skills* (SEPSCTS pour ses sigles en anglais), qui comprend 36 éléments répartis en cinq facteurs. Les données ont été analysées à l'aide de tests t pour échantillons appariés et d'analyses de covariance (ANCOVA). Les résultats ont démontré une augmentation significative des perceptions d'auto-efficacité des élèves en matière de compétences de pensée computationnelle à la suite des activités, cette augmentation étant observée de manière cohérente entre les genres. Finalement, les défis rencontrés au cours de la recherche et de la pratique ont été rapportés, ainsi que les limites de l'étude, afin d'informer les recherches futures.

Mots-clés: pensée computationnelle, perception, codage robotique, auto-efficacité

Introduction

Computational Thinking

Computational thinking was pioneered by Papert (1980) who suggested that individuals must be taught essential thinking skills related to programming, and by Perlis (1962) who highlighted the importance of computer technology in solving real-life problems with its structure and operational logic. Computational thinking is defined as "problem-solving, system designing, and understanding human behaviour using the concepts of computer science" (Wing, 2006, p. 33). However, Sysło and Kwiatkowska (2013) argued that the focus should be on thinking skills based on the principles of programming and the systematic solution of problems rather than just programming skills for solutions, designs, and gains as emphasized in the general definition offered by Wing (2006). Similarly, the International Society for Technology in Education (ISTE) suggested that instead of teaching specific skills in the field of informatics, individuals should be encouraged to adopt the habit of applying this skill, which reflects algorithmic, creative, critical thinking, and problem-solving skills, in every area of life. In this regard, a joint study conducted by ISTE in collaboration with the Computer Science Teachers Association (CSTA) in 2011 defined the wordnet related to the concept of computational thinking to include the following components: data collection, analysis, representation, decomposition, abstraction, algorithms and procedures, automation, simulation, and parallelization.

When examining frequently cited publications in the literature, it becomes evident that the fundamental concepts of the computational thinking process are categorized in various ways. Computational thinking encompasses a set of key concepts and steps that are essential for problem-solving and system design in the field of computer science. These concepts have been classified by various researchers in the literature. According to Wing (2006; 2008; 2011), the key concepts include problem decomposition, abstraction, algorithms, automation, and generalization. Barr and Stephenson (2011) categorized them as abstraction, algorithms and procedures, automation, problem decomposition, parallelization (parallel processing), and simulation. Lee et al. (2011) emphasized abstraction,

automation, and analysis. Selby and Woollard (2013) identified abstraction, algorithms or algorithmic thinking, problem decomposition, evaluation, and generalization as key components. Lastly, Angeli et al. (2016) outlined abstraction, algorithms or algorithmic thinking, problem decomposition, debugging, and generalization as fundamental elements of computational thinking.

It is not surprising that the basic concepts are perceived and classified differently depending on the experiences of practice-based researchers. However, most studies tend to refer to Wing's (2008) classification, which was then adapted to practice and transformed into a framework that includes the following basic steps:

1. **Problem decomposition**: the process of breaking down a problem into smaller, more manageable parts to facilitate a more efficient solution.
2. **Abstraction**: the prioritization and classification of solution methods and necessary information, based on their significance (inclusion and exclusion), and the conceptualization of solution approaches.
3. **Algorithm (algorithmic thinking)**: the design and planning of the necessary steps to effectively and efficiently resolve a problem.
4. **Automation (automatization)**: the technological structuring of algorithms, enabling them to be applied autonomously and limitlessly within their respective contexts.
5. **Generalization (generalizing)**: the adaptation of the technological solutions generated to make them applicable or feasible for addressing other similar or different problems.

   (pp. 3717–3719)

When examined comprehensively in terms of both concept and process, computational thinking distinguishes itself from problem-solving skills, to which it closely relates, in several significant ways (ISTE & CSTA, 2011). First, it involves the structuring of solutions to problems from a technology-oriented perspective. It emphasizes not only the hierarchical nature of problems and sub-problems but also the analysis of patterns and the recognition of similarities among them. Additionally, it assesses the suitability and, ideally, the efficiency of the methods and sources of solutions. Furthermore, computational thinking encompasses the capacity to automate technological solutions for similar problems and to transfer and generalize these solutions to address diverse problem domains. In this regard, improving computational thinking cannot be achieved solely through efforts to enhance problem-solving skills; such efforts would yield limited results. Therefore, as described in the literature, there are specialized activities and practices designed to enhance computational thinking skills, along with numerous strategies and methods that can be employed in these learning environments.

In the literature review conducted by Berikan (2018), strategies and methods currently employed or recommended for improving computational thinking skills include project-based learning (Bower et al., 2017), product-based learning (Brennan & Resnick, 2012), questioning and discussion strategies (ISTE & CSTA, 2011), and problem-based learning (Mingo, 2013). Similarly, Gülbahar et al. (2019) reported the use of computer games (Apostolellis et al., 2014), simulations (Basawapatna et al., 2013),

unplugged activities (Curzon et al., 2014), and traditional programming activities for enhancing computational thinking skills. Additionally, the literature encompasses the use of mobile programming activities (Morelli et al., 2011), course processes (Israel et al., 2015; Rubinstein & Chor, 2014), and curriculum designs (Bers et al., 2014) to foster computational thinking skills.

In addition to the aforementioned strategies, methods, and activities designed to enhance computational thinking skills, it is possible that alternative approaches have not yet been explored. Ackermann (2001) points out that all strategies, methods, or activities related to the development of computational thinking skills, whether studied or not, are based on Piaget's constructivism and Papert's constructionist theory. In their literature review study, Lye and Koh (2014) examined 27 studies focused on the development of computational thinking skills at the K–12 level and found that these studies were grounded in these theories.

Built upon the principles of learning-by-doing and problem-solving in real-life contexts, these theories serve as the foundation for the assertion made by Hsu et al. (2018) that many educators consider programming to be the most direct and effective method for teaching computational thinking skills, as it encourages independent problem-solving. Consequently, traditional programming, unplugged programming activities, and contemporary approaches like robotics and physical programming are steadily gaining popularity in the quest to develop computational thinking skills.

Robotic Coding

The use of robotics in education can be categorized into three primary tendencies: robotics as learning objectives, robotics as learning aids, and robotics as learning tools (Eguchi, 2012). In contrast, Sullivan and Heffernan (2016) classified the use of robotics in education into “first-order uses” and “second-order uses” (p. 3). They proposed that robots could serve as a means for direct instruction in robotics (first-order uses) or as analogical tools for teaching in other academic domains (second-order uses). Educational robots, defined as robots used as learning tools within classroom settings, fall into this category (Eguchi, 2017).

The history of educational robots traces back to the pioneering work of Seymour Papert, the creator of the Logo programming language and the proponent of constructionist theory (Papert, 1993). Constructionism is founded on the principle that learning takes place when individuals actively construct a meaningful product (Harel & Papert, 1991). In fact, Harel and Papert provided the simplest definition for constructionism as “learning by making” (para. 1). In this context, constructionism shares similarities with the constructivist approach, viewing knowledge as a personal experience to be constructed rather than a commodity to be transmitted, encoded, stored, and reused (Ackermann, 2001).

Robotic activities assist students in constructing new knowledge if the pedagogical process is also engaging active learners to collaborate with their peers and adopt the role of researchers (Atmatzidou & Demetriadis, 2016). The hands-on nature of educational robots, which allows students to learn through practical experience, fosters an engaging and stimulating educational environment by seamlessly integrating technology into the subject matter (Eguchi, 2017). Furthermore, educational robotics offers students an opportunity to work collaboratively (Alimisis, 2013; Bers, 2008), providing a

tangible means of comprehending abstract concepts and ideas (Bers, 2008). For example, an educational robot operating under the same program will exhibit different movements on smooth surfaces like parquet floors compared to uneven surfaces like carpets, due to variations in frictional forces generated by the surfaces. This immediate feedback from the educational robot renders abstract concepts visible and facilitates students' grasp of these concepts (Eguchi, 2016).

From a programming perspective, the physical nature of educational robots offers students immediate and tangible feedback on the efficiency of the algorithms they have written (Eguchi, 2012; Sullivan & Heffernan, 2016). Educational robotics kits enable even beginners to program complex robotic behaviours using the graphical programming environments. As a result, educational robotics kits can be effectively used with young learners. Moreover, the literature features several studies conducted with kindergarteners on robotic coding (Bers et al., 2014; Chalmers, 2018; Kazakoff et al., 2013).

In summary, educational robotics kits are frequently used to achieve various objectives, including the development of programming skills (Atmatzidou et al., 2008; Baek et al., 2019), enhancement of problem-solving abilities (Nugent et al., 2016; Witherspoon et al., 2017), and the teaching of cooperative learning skills (Eguchi, 2014a; Mitnik et al., 2008). Furthermore, there is a body of research demonstrating that educational robotics can also contribute to the development of computational thinking skills (Atmatzidou & Demetriadis, 2016; Bers et al., 2014; Constantinou & Ioannou, 2018; Leonard et al., 2016; Noh & Lee, 2020).

Robotic Coding and Computational Thinking

Educational robotics is recognized as an effective tool for enhancing students' computational thinking skills (Baek et al., 2019; Chalmers, 2018; Eguchi, 2014b, 2016) because students must consider the robot design and behaviour programming that will best interact with the physical world within the context of a given situation or problem. During this process of generating solutions, participants first thoroughly analyze the problem and the constraints of the real world. Then, based on their findings and the educational robot's capabilities, including its perception of the real world and the responses to stimuli encoded in the program, they assess the program's numeric and decision-making values. If the educational robot exhibits unexpected behaviour, it may indicate flaws in the implementation of ideas or conditions encountered that were either overlooked or misjudged during the abstraction phase (Lee et al., 2011). Once the coded program operates successfully on the robot, automation is achieved.

It is acknowledged that the number of experimental studies investigating the impact of robotic coding on students' computational thinking skills is limited (Berland & Wilensky, 2015; Ioannou & Makridou, 2018). Notably, the literature highlights a scarcity of studies conducted at the K–12 level, indicating insufficiency in this area (Constantinou & Ioannou, 2018; Witherspoon et al., 2017). Recognizing this gap, the present study aims to explore the effect of robotic coding activities on secondary school students' self-efficacy perceptions of computational thinking skills and whether any significant gender-based differences exist. The following research questions guided this study:

1) How do robotic coding activities influence the self-efficacy perceptions of secondary school students regarding computational thinking skills?

2) Are there any gender-based differences in the impact of robotic coding activities on the self-efficacy perceptions of secondary school students regarding computational thinking skills?

Method

Research Design

In this study, which investigated secondary school students' self-efficacy perceptions of computational thinking skills, a one-group pretest-posttest quasi-experimental design was employed. This design, aligned with the quantitative research approach, facilitates the repeated analysis of numerical data using statistical methods and is frequently used to examine causality through comparisons or deductions in educational settings where it is not possible to exert full control over variables (Büyüköztürk et al., 2017; Cohen et al., 2017). A one-group pretest-posttest quasi-experimental design involves obtaining measurements or observations related to a single group by administering the same measurement instrument at different points in time (before and after implementation; Fraenkel et al., 2011; McMillan & Schumacher, 2013).

Participants

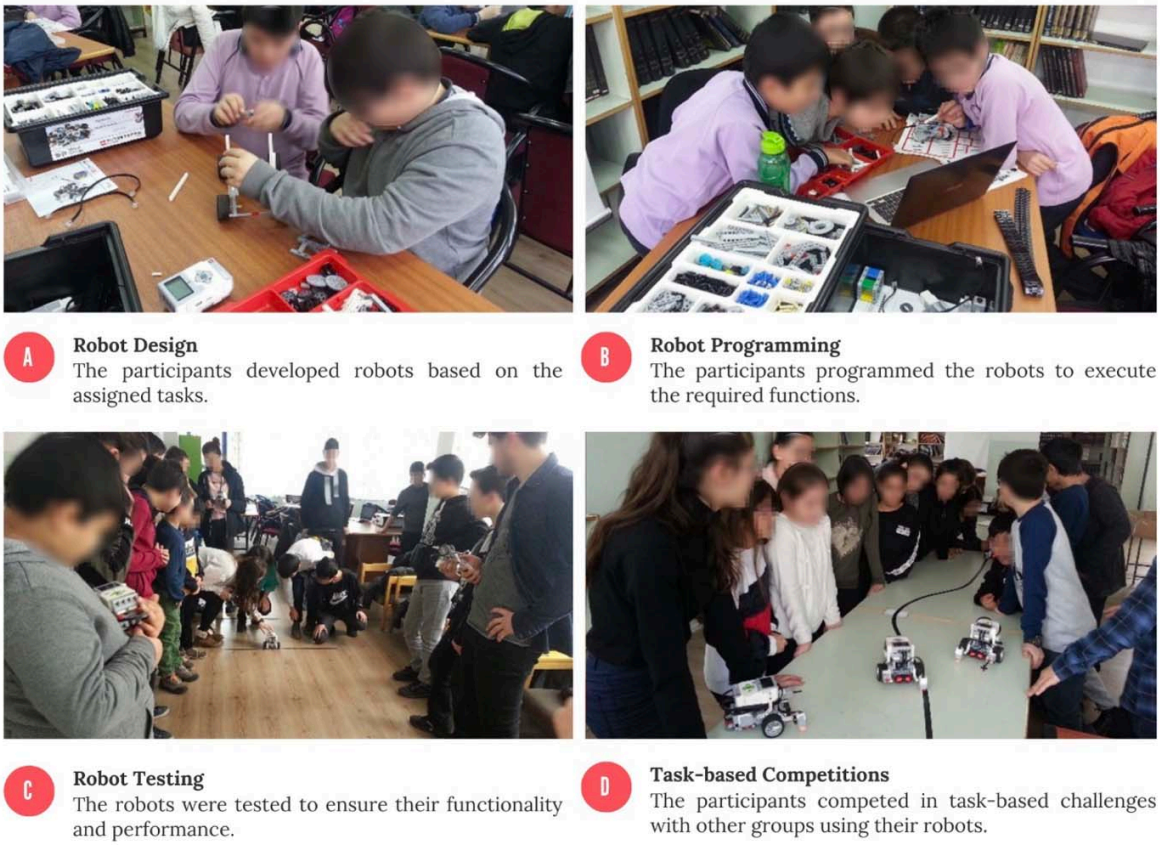
The study comprised a cohort of 32 students (17 males and 15 females) enrolled in a public secondary school in Turkey. The secondary school level consists of 5^{th}, 6^{th}, 7^{th}, and 8^{th} grade levels. Participant selection for the robotic coding activities, conducted outside of regular school hours, was facilitated by the school administration, drawing from a pool of volunteers. Initially, 40 students enrolled in these activities, however, 8 students were subsequently excluded from the study for various reasons, including discrepancies in the use of aliases between the pretest and posttest, as well as non-participation in the posttest assessment. Consequently, data from 32 students (20 in the sixth grade and 12 in the eighth grade) were considered for analysis. Participants were further categorized based on their grade levels, resulting in 9 males and 11 females in the sixth grade, and 8 males and 4 females in the eighth grade.

Robotic Coding Activities

Group activities involving Lego Mindstorms EV3 Education robots and expansion kits were conducted in the school library, which was physically designed to facilitate these activities. Due to space constraints and a limited number of robotic kits available (n=5), separate sessions were organized for sixth and eighth grades. The participants engaged in a robotics curriculum delivered over a five-week period, comprising a total of 20 instructional hours. This curriculum was implemented through weekly sessions, each lasting four hours. Figure 1 shows a selection of photographs captured during the robotic coding activities.

Figure 1

Images of Robotic Coding Activities



Note. Images illustrate the stages of the robotic coding activities. Panel A: Participants constructed robots based on the tasks. Panel B: Participants programmed the robots to execute the specified functions. Panel C: The robots were tested to ensure that the components designed for the task worked. Panel D: Competitions utilizing their robots to address specific task-oriented challenges.

Students were organized into groups of four, with each group assigned an experienced senior university student as a mentor. Mentors received training on Lego Mindstorms EV3 robotic kits and had worked on various projects. Mentors provided guidance to their respective student groups to assist them in completing the assigned tasks. It's important to note that the mentors did not provide direct solutions to the problems; rather, they offered the students support when needed.

Lego Mindstorms, one of the most widely used kits in the field of robotics (Benitti & Spolaôr, 2017), was employed for these activities. Specifically, the Lego Mindstorms EV3 Education robotic kit, designed specifically for educational purposes (Afari & Khine, 2017), was used. The sets include a programmable smart brick that serves as a compact computer, controlling the motors and collecting data from sensors. It also consists of three servo motors, seven data cables, one USB cable, a rechargeable

battery, as well as gears, treads, wheels, axles, and other technical components[1]. The EV3 programming software features a user-friendly drag-and-drop graphics interface, enabling students to concentrate on computational concepts rather than the syntax of programming languages (Ching et al., 2018). Additionally, the software offers the capability to visualize graphs of data collected from sensors through its experiment interface.

Students received hands-on instruction on robot programming and operation using the Lego Mindstorms kit, including guidance on configuring robot movements, using screen and sound functions, understanding and using sensors such as ultrasonic, color, touch, and gyroscope sensors, performing arithmetic and logical operations, logging data, and creating graphic representations. To engage student motivation, several games and competitions were incorporated into the activities. The final day of the program was dedicated to robotics projects, during which each group independently designed and programmed their robots to address real-life problems, such as creating an electronic cane. Sample two-hour activity plans are presented in the Appendix. The aims of the activities were as follows:

- to trigger students' creativity by enabling them to develop their own robots,
- to develop critical thinking and problem-solving skills by presenting unstructured problems,
- to develop students' social skills, communication and co-operation skills, and leadership and responsibility skills by directing them to group work,
- to develop critical thinking and problem-solving skills by programming robots for specific purposes,
- to create a student-centred learning environment where real-life problems are presented during the activities and to direct students to multidimensional and interdisciplinary thinking, and
- to trigger students' curiosity, research, and learning desires with a learning environment where robots can be developed in an unlimited way.

Instruments

The study used the Self-Efficacy Perception Scale for Computational Thinking Skills (SEPSCTS), which was developed by Gülbahar and colleagues in 2019. This scale consists of 36 items categorized into five factors. The reliability coefficients associated with the various factors of the scale fall within the range of .762 to .930, with an impressive overall reliability coefficient of .943 (Table 1). The researchers affirm that this scale serves as a suitable and dependable instrument for assessing the perceptions of self-efficacy in computational thinking skills among students aged 10 to 14 years.

[1] http://education.lego.com

Table 1

Items and Reliability Coefficients of SEPSCTS Factors

Factor	Items *n*	Cronbach's alpha (α) coefficient
Algorithm design competency	9	.930
Problem-solving competency	10	.880
Data-processing competency	7	.856
Basic programming competency	5	.838
Self-confidence competency	5	.762
Overall	36	.943

Data Collection and Analysis

The SEPSCTS was applied to all participants both before and after the robotic coding activities. An equal number of pretest and posttest questionnaires were analyzed to determine the appropriate statistical technique for the study. Data collected were processed using IBM SPSS Statistics (Version 23), and skewness and kurtosis values were examined to assess normality violations. According to George and Mallery (2019), skewness and kurtosis values within the range of ±1 are considered ideal, while values within the range of ±2 are considered normal for various psychometric purposes. In this study, the skewness and kurtosis values were found to be within the range of ±2 for only two of the sub-scales, and within the range of ±1 overall. Consequently, parametric tests were chosen for the quantitative data analysis. To compare students' self-efficacy perceptions for computational thinking skills before and after the activities, a paired samples *t*-test was employed. Additionally, to investigate potential differences in students' self-efficacy perceptions based on gender, one-way covariance analysis (ANCOVA) was used (Pedhazur & Schmelkin, 2013).

Results

Self-Efficacy Perceptions of Computational Thinking Skills

The significance levels of the *t*-test, as well as the means and standard deviations of the pretest and posttest data collected through the scale to assess the impact of robotic coding activities on the self-efficacy perceptions of secondary students regarding computational thinking skills, are detailed in Table 2.

Prior to the activities, students exhibited the lowest mean scores on the basic programming competency subscale and the highest mean scores on the problem-solving competency subscale. After the activities, the students continued to have the lowest mean scores on the basic programming

competency subscale, but they now had the highest mean scores on the algorithm design competency subscale, surpassing their previous high in problem-solving competency. According to the paired-samples *t*-test results, the students' self-efficacy perceptions significantly increased in all sub-dimensions and the total self-efficacy scores of the scale ($p < .05$). Cohen's *d* effect sizes were calculated for all sub-dimensions and total SEPSCTS scores to assess the magnitude of the changes between the pretest and posttest. The results indicate a medium-effect size for problem-solving and self-confidence competencies and a large-effect size for the total SEPSCTS, algorithm design, data processing, and basic programming competencies. Based on these findings, it was concluded that the robotic coding activities conducted with secondary school students had a positive effect on their self-efficacy perception of computational thinking skills.

Table 2

Results of the t-*Test for Self-Efficacy Perceptions in Computational Thinking Skills*

Factor	Pretest		Posttest		*t*(31)	*p*	Cohen's *d*
	M	*SD*	*M*	*SD*			
Algorithm design competency	2.01	0.44	2.70	0.30	7.79	.000*	1.38
Problem-solving competency	2.45	0.27	2.60	0.30	2.45	.020*	0.43
Data-processing competency	2.26	0.41	2.63	0.34	5.14	.000*	0.91
Basic programming competency	1.83	0.57	2.39	0.45	5.07	.000*	0.90
Self-confidence competency	2.38	0.43	2.61	0.30	2.38	.024*	0.42
Total self-efficacy perception	2.20	0.30	2.61	0.26	6.50	.000*	1.15

Note. N=32. *p < .05.

Gender Differences

Assumptions were assessed before the ANCOVA, and it was observed that the assumptions were satisfied concerning the normal distribution of the dependent variable, homogeneity of variance (Levene's test), and equivalence of regression line slopes. A summary of posttest scores on SEPSCTS when adjusted for pretest scores of SEPSCTS (the covariate) for both male and female students, reveal no statistically significant differences in total scores or across all factors (Table 3). In other words, the positive impact of robotic coding activities on the self-efficacy perceptions of secondary school students regarding computational thinking skills was consistent regardless of gender.

Table 3

ANCOVA Results for SEPSCTS by Gender

Factor	Gender	*n*	Posttest		$F(1, 29)$	*p*	partial η^2
			M	*SD*			
Algorithm design competency	Male	17	2.73	0.24	0.37	.547	0.47
	Female	15	2.67	0.36			
Problem-solving competency	Male	17	2.57	0.34	0.50	.487	0.21
	Female	15	2.63	3.36			
Data-processing competency	Male	17	2.67	0.28	1.80	.190	0.21
	Female	15	2.60	0.41			
Basic programming competency	Male	17	2.51	0.42	2.33	.138	0.57
	Female	15	2.27	0.46			
Self-confidence competency	Male	17	2.60	0.31	0.09	.766	0.14
	Female	15	2.62	0.29			
Total self-efficacy perception	Male	17	2.62	0.22	0.33	.572	0.15
	Female	15	2.58	0.31			

Discussion

Although computational thinking skills may initially seem to be primarily associated with the fields of engineering and science, Furber (2012) argued that these skills are essential for anyone aspiring to lead a high-quality life. However, computational thinking sets itself apart from other competencies that are expected from individuals, such as problem-solving, product development, project work, and finding solutions, due to its distinct technological efficiency perspective. From this viewpoint, it is suggested that the use of computational tools can lead to more effective and efficient problem-solving outcomes (Cuny et al., 2010, as cited in Wing, 2010). Therefore, it is believed that analyzing problems within this computational context, classifying them, and synthesizing the findings can yield generalizable solutions (ISTE & CSTA, 2011).

Wing (2008) asserted that the development of computational thinking skills can commence as early as preschool, while Barr and Stephenson (2011) argued it is more suitable to introduce these

activities in secondary school, aligning with Piaget's (1936) formal operational stage (typically beginning at age 11). Given that study participants were students in the sixth and eighth grades, the choice aligns with the varying expert opinions found in the literature. It is noteworthy that the methods and approaches selected for educational activities aimed at enhancing computational thinking skills are just as crucial as determining the appropriate starting level for these activities.

The SEPSCTS instrument, used in this research, has five sub-dimensions: algorithm design, problem-solving, data processing, basic programming, and self-confidence competencies. The findings revealed that students demonstrated significant improvements across these key competencies, indicating that students' self-efficacy perceptions in computational thinking were enhanced as a result of participating in the robotic coding activities.

Algorithm Design

Algorithmic thinking is a key component of computational thinking, and engaging in robotics coding helps students grasp and apply algorithmic thinking by creating step-by-step instructions for their robots (Grover & Pea, 2018). This study demonstrated significant improvement in students' algorithm design skills after 20 hours of engagement with educational robotics activities. The related literature showed that the development of algorithmic thinking can begin at a young age with suitable, simpler tasks (Ching & Hsu, 2024). Robotics activities provide young learners with a hands-on approach to developing algorithmic thinking through tangible robots, interactive exercises, and real-time feedback, supporting earlier research on fostering these skills in young children (Futschek & Moschitz, 2011). This study's demonstration of enhanced algorithmic thinking skills mirrors the conclusions drawn in similar studies across the literature, such as Yilmaz Ince and Koc's (2021) study on robotics programming education. However, Atmatzidou and Demetriadis (2014) reported that most students had difficulty describing the algorithm in a clear and accurate manner in their study with educational robotic activities used for secondary school students.

Problem-Solving

Computational thinking involves problem-solving (Wing, 2006), and is identified as one of its core concepts (Yadav et al., 2016). The findings of the study revealed that the students' problem-solving competencies, as measured by the SEPSCTS, had significantly improved by the end of the intervention. Many studies report that robotic activities had positive effects on problem-solving skills (Atmatzidou & Demetriadis, 2014; Blanchard et al., 2010; Constantinou & Ioannou, 2018; Eguchi, 2014b; Karp & Maloney, 2013; Petre & Price, 2004). Blanchard et al. (2010) emphasized that robotic coding activities require the use of problem-solving skills and creative thinking. Robotic-based tasks demand that students coordinate the physical movements of the robot with the virtual commands of their programming system, which is recognized as a detailed and intricate endeavour.

Programming

When reviewing the literature, it becomes evident that educational activities designed to enhance computational thinking skills often revolve around STEM (Science, Technology, Engineering, and

Mathematics) and programming activities. Since this study specifically investigates the impact of robotic coding activities on computational thinking skills, it focuses on programming-related activities while excluding STEM-based approaches. Programming has been recognized as particularly effective in developing various cognitive skills, including problem-solving, metacognitive thinking, creativity, analytical thinking, critical thinking, and algorithmic thinking (Akpınar & Altun, 2014; Atmatzidou & Demetriadis, 2016; Fesakis & Serafeim, 2009; Psycharis & Kallia, 2017). Therefore, it is reasonable to assume that programming activities can significantly contribute to the enhancement of computational thinking skills. The students' self-efficacy in programming skills has shown a significant increase, which aligns with the anticipated results of this study. This is because children must program the robots using a computer in order to interact with educational robots. This result is consistent with the existing body of research on the topic. Integrating robots into programming lessons helps make abstract concepts more tangible, both visually and physically, which facilitates students' understanding of computer science ideas (Noh & Lee, 2020). Programming is crucial for developing computational thinking (Lye & Koh, 2014) and stands out as one of the most effective methods for nurturing this skill (Pala & Mıhçı Türker, 2021). Existing research has demonstrated that programming education is directly linked to the development of computational thinking skills (Avello et al., 2020). For instance, Sun et al. (2022) found that students' attitudes towards programming were a significant predictor of their computational thinking abilities.

Computational Thinking

This study aimed to investigate the effects of robotic coding activities on students' self-efficacy perceptions of computational thinking. Expectations were aligned with prior research in the realm of computational thinking, which also demonstrated positive outcomes. The analysis revealed that the robotic coding activities had a statistically significant positive impact on students' self-efficacy perceptions of computational thinking. This finding aligns with similar studies in the literature, particularly those related to programming education. For instance, Karaahmetoglu and Korkmaz (2019) found that Arduino-based robotics had a significantly more positive effect on students' computational thinking skills compared to block-based programming. Similarly, Yilmaz Ince and Koc (2021) found significant improvements in algorithmic and critical thinking skills related to computational thinking, but no significant effect on creativity, cooperation, and problem-solving skills. In another study, Kert et al. (2020) revealed that robotic coding activities more effectively enhanced computational thinking skills compared to the group focused solely on programming. Also, Constantinou and Ioannou (2018) conducted two quasi-experimental studies—in their first study, they observed statistically significant improvements in computational thinking skills based on students' scores, while in their second study, significant improvements in computational thinking skills were noted among students in the experimental group, while the control group did not exhibit significant progress. Similarly, Noh and Lee (2020) employed a one-group pretest-posttest research design, akin to our present study, and the results demonstrated a statistically significant improvement in students' computational thinking skills due to robotic programming instruction.

Despite the similarity of common findings across studies employing diverse approaches and research designs, there are nuanced differences to consider. In a comprehensive literature review, Ioannou and Makridou (2018) examined research related to the influence of educational robots on the computational thinking skills of K–12 students. They stated that while the studies generally indicated the development of 21st-century skills, including computational thinking components such as decomposition, abstraction, algorithmic thinking, and debugging, it remains unclear to what extent these specific facets of computational thinking skills were enhanced in the context of robotic coding and computational thinking studies.

Gender

Witherspoon et al. (2017) identified a significant difference between pretest and posttest scores on computational thinking, however, the study found no notable gender-based differences. Remarkably, our study aligns closely with these findings, showing a similar significant improvement in computational thinking skills and a lack of gender-based differences. Likewise, Atmatzidou and Demetriadis (2016) reported no gender-based differences at the end of their implementation. However, they noted that females required more instructional time to achieve the same level of computational skills as males. Noh and Lee's (2020) study, like ours, demonstrated a statistically significant improvement in students' computational thinking skills as a result of robotic programming instruction. Importantly, the researchers observed no significant gender-based variations in the enhancement of computational thinking skills.

Upon comprehensive review of our study outcomes in conjunction with findings from related literature, it is clear that robotic coding activities have a positive impact on secondary school students' computational thinking skills. Consequently, these activities bear substantial significance for researchers looking to enhance this skill. The finding that gender does not significantly affect the outcomes is another noteworthy result of the study. Given these findings, the following recommendations are proposed, recognizing the limitations of the present study.

Conclusions and Recommendations

This study investigated the effects of robotic coding instruction, using the Lego Mindstorms Education EV3 set, on students' self-efficacy in computational thinking skills. Findings revealed a statistically significant improvement in students' perceptions of their abilities, with this positive change being consistent across genders.

The results of this study are limited by several factors, which future research could address by considering the following aspects. First, the study involved 32 students from the sixth and eighth grades. Future researchers in a similar context can enhance the diversity of their studies by including participants from various K–12 levels; focusing on participants from the same grade level may contribute to data reliability. Second, considering that this study employed a one-group pretest-posttest quasi-experimental design within a quantitative approach, researchers in similar contexts are encouraged to incorporate a control group when following a quantitative approach alone and to explore qualitative methods for deeper insights. Third, this study relied solely on the SEPSCTS. Diversifying data

collection methods is another valuable recommendation. Future research can enrich data by incorporating various methods such as project or product evaluations, design assessments, portfolio evaluations, and algorithm assessments, as suggested by the literature for evaluating computational thinking skills (Brennan & Resnick, 2012; Chen et al., 2017; Koh et al., 2010). Fourth, the study was conducted with a limited number (5) of robotics kits. Conducting studies with individual students or pairs can eliminate some constraints and offer valuable comparability with activities that do not involve robotic programming. Fifth, extending instruction duration beyond the 20 hours provided in this study for more practice and project development is recommended. In longer-duration studies, participants could gain experience in instructor-guided project development for each task before independently developing authentic projects, recognizing the challenges in project development even after successfully completing individual tasks. Lastly, while ready-to-go robot kits are valuable, they are better suited for lower-grade levels in K–12 education. To nurture creativity and authenticity among higher-grade-level students, researchers might explore approaches involving physical programming with Arduino and similar platforms.

References

Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference? *Massachusetts Institute of Technology: Future of Learning Group publication*, *5*(3), 438–448. https://learning.media.mit.edu/content/publications/EA.Piaget%20_%20Papert.pdf

Afari, E., & Khine, M. S. (2017). Robotics as an educational tool: Impact of Lego Mindstorms. *International Journal of Information and Education Technology, 7*(6), 437–442. https://doi.org/10.18178/ijiet.2017.7.6.908

Akpınar, Y., & Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi [The need for programming education in schools of the information society]. *Elementary Education Online*, *13*(1), 1–4. https://ilkogretim-online.org/index.php/pub/article/view/6168

Alimisis, D. (2013). Educational robotics: Open questions and new challenges. *Themes in Science and Technology Education*, *6*(1), 63–71. https://files.eric.ed.gov/fulltext/EJ1130924.pdf

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. *Journal of Educational Technology & Society*, *19*(3), 47–57. http://www.jstor.org/stable/jeductechsoci.19.3.47

Apostolellis, P., Stewart, M., Frisina, C., & Kafura, D. (2014). RaBit EscAPE: A board game for computational thinking. In O. S. Iversen, P. Markopoulos, C. Dindler, F. Garzotto, C. Frauenberger, & A. Zeising (Eds.), *Proceedings of the 2014 Conference on Interaction Design and Children* (pp. 349–352). ACM. https://doi.org/10.1145/2593968.2610489

Atmatzidou, S., & Demetriadis, S. (2014). How to support students' computational thinking skills in educational robotics activities. In D. Alimisis, G. Granosik, & M. Moro (Eds.), *Proceedings of 4th International Workshop Teaching Robotics, Teaching With Robotics & 5th International Conference Robotics in Education* (pp. 43–50). University of Padova. https://www.terecop.eu/TRTWR-RIE2014/files/00_WFr1/00_WFr1_06.pdf

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75*, 661–670. https://doi.org/10.1016/j.robot.2015.10.008

Atmatzidou, S., Markelis, I., & Demetriadis, S. (2008). The use of LEGO Mindstorms in elementary and secondary education: Game as a way of triggering learning. In S. Carpin, I. Noda, E. Pagello, M. Reggiani, & O. Stryk (Eds.), *International Conference of Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)* (pp. 22–30). Springer. http://www.dei.unipd.it/~emg/downloads/SIMPAR08-WorkshopProceedings/TeachingWithRobotics/atmatzidou_et_al.pdf

Avello, R., Lavonen, J., & Zapata-Ros, M. (2020). Codificación y robótica educativa y su relación con el pensamientocomputacional y creativo. Una revisión compresiva [Coding and educational robotics and their relationship with computational and creative thinking. A comprehensive review]. *Revista de Educación a Distancia (RED)*, *20*(63). https://doi.org/10.6018/red.413021

Baek, Y., Yang, D., & Fan, Y. (2019). Understanding second grader's computational thinking skills in robotics through their individual traits. *Information Discovery and Delivery, 47*(4), 218–228. https://doi.org/10.1108/IDD-09-2019-0065

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

Basawapatna, A. R., Repenning, A., & Lewis, C. H. (2013). The simulation creation toolkit: An initial exploration into making programming accessible while preserving computational thinking. In T. Camp & P. Tymann (Chairs), *Proceedings of the 44th ACM Technical Symposium on Computer Science Education* (pp. 501–506). ACM. https://doi.org/10.1145/2445196.2445346

Benitti, F. B. V., & Spolaôr, N. (2017). How have robots supported STEM teaching?. In M. Khine (Ed.), *Robotics in STEM education* (pp. 103–129). Springer. https://doi.org/10.1007/978-3-319-57786-9_5

Berikan, B. (2018). *Formative evaluation of "problem solving data sets" learning experience designed to improve computational thinking skills* [Unpublished doctoral dissertation]. Gazi University.

Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology*, *24*(5), 628–647. https://doi.org/10.1007/s10956-015-9552-x

Bers, M. U. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. Teachers College Press.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Blanchard, S., Freiman, V., & Lirrete-Pitre, N. (2010). Strategies used by elementary schoolchildren solving robotics-based complex tasks: Innovative potential of technology. *Procedia-Social and Behavioral Sciences*, *2*(2), 2851–2857. https://doi.org/10.1016/j.sbspro.2010.03.427

Bower, M., Wood, L., Lai, J., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education*, *42*(3), 53–72. http://dx.doi.org/10.14221/ajte.2017v42n3.4

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, Canada,* 1–25. https://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2017). *Bilimsel araştırma yöntemleri [Scientific research methods*]. Pegem Atıf İndeksi.

Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, *17*, 93–100. https://doi.org/10.1016/j.ijcci.2018.06.005

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, *109*, 162–175. https://doi.org/10.1016/j.compedu.2017.03.001

Ching, Y.-H., & Hsu, Y.-C. (2024). Educational robotics for developing computational thinking in young learners: A systematic review. *TechTrends*, *68*(3), 423–434. https://doi.org/10.1007/s11528-023-00841-1

Ching, Y.-H., Hsu, Y.-C., & Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends*, *62*(6), 563–573. https://doi.org/10.1007/s11528-018-0292-7

Cohen, L., Manion, L., & Morrison, K. (2017). *Research methods in education* (8th Edition). Routledge. https://doi.org/10.4324/9781315456539

Constantinou, V., & Ioannou, A. (2018). Development of computational thinking skills through educational robotics. In V. Dimitrova, S. Praharaj, M. Fominykh, & H. Drachsler (Eds.), *EC-TEL Practitioner Proceedings 2018: 13th European Conference on Technology Enhanced Learning* (pp. 1–11). CEUR-WS. http://ceur-ws.org/Vol-2193/paper9.pdf

Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014, November). Introducing teachers to computational thinking using unplugged storytelling. In C. Schulte, M. E. Caspersen, & J. Gal-Ezer (Chairs), *Proceedings of the 9th Workshop in Primary and Secondary Computing Education* (pp. 89–92). ACM. https://doi.org/10.1145/2670757.2670767

Eguchi, A. (2012). Educational robotics theories and practice: Tips for how to do it right. In B. Barker, G. Nugent, N. Grandgenett, & V. Adamchuk (Eds.), *Robots in K-12 education: A new technology for learning* (pp. 1–30). IGI Global Scientific Publishing. https://doi.org/10.4018/978-1-4666-0182-6.ch001

Eguchi, A. (2014a). Educational robotics for promoting 21st century skills. *Journal of Automation, Mobile Robotics and Intelligent Systems, 8*(1)*,* 5–11.

Eguchi, A. (2014b). Learning experience through RoboCupJunior: Promoting engineering and computational thinking skills through robotics competition. In *2014 ASEE Annual Conference & Exposition* (pp. 24.852.1–24.852.18). https://www.doi.org/10.18260/1-2--20743

Eguchi, A. (2016, March). Computational thinking with educational robotics. In G. Chamblee & L. Langub (Eds.), *Society for Information Technology & Teacher Education 27th International Conference* (pp. 79–84). Association for the Advancement of Computing in Education. https://www.learntechlib.org/p/172306

Eguchi, A. (2017). Bringing robotics in classrooms. In M. Khine (Ed.), *Robotics in STEM education* (pp. 3–31). Springer. https://doi.org/10.1007/978-3-319-57786-9_1

Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with "scratch" on future teachers' opinions and attitudes about programming and ICT in education. *ACM SIGCSE Bulletin*, *41*(3), 258–262. https://doi.org/10.1145/1595496.1562957

Fraenkel, J., Wallen, N., & Hyun, H. (2011). *How to design and evaluate research in education* (8th ed.). McGraw-Hill Education.

Furber, S. (2012). *Shut down or restart? The way forward for computing in UK schools*. Royal Society. https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf

Futschek, G., & Moschitz, J. (2011). Learning Algorithmic Thinking with Tangible Objects Eases Transition to Computer Programming. In *Lecture notes in computer science* (pp. 155–164). https://doi.org/10.1007/978-3-642-24722-4_14

George, D., & Mallery, P. (2019). *IBM SPSS statistics 26 step by step: A simple guide and reference* (16th ed.). Routledge. https://doi.org/10.4324/9780429056765

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer science education: Perspectives on teaching and learning in school* (pp. 19–38). Bloomsbury Academic. https://www.doi.org/10.5040/9781350057142.ch-003

Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik özYeterlik algısı ölçeği: Geçerlik ve güvenirlik çalışması [The self-efficacy perception scale for computational thinking skill: Validity and reliability study]. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *10*(1), 1–29. https://turcomat.org/index.php/turkbilmat/article/view/194

Harel, I. E., & Papert, S. E. (1991). *Constructionism*. Ablex Publishing.

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. http://doi.org/10.1016/j.compedu.2018.07.004

Ioannou, A., & Makridou, E. (2018). Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *Education and Information Technologies*, *23*(6), 2531–2544. https://doi.org/10.1007/s10639-018-9729-z

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, *82*, 263–279. https://doi.org/10.1016/j.compedu.2014.11.022

International Society for Technology in Education [ISTE] & Computer Science Teachers Association [CSTA]. (2011). *Computational Thinking in K–12 Education: Leadership Toolkit*. https://cdn.iste.org/www-root/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4

Karaahmetoglu, K., & Korkmaz, Ö. (2019). The effect of project-based Arduino educational robot applications on students' computational thinking skills and their perception of basic STEM skill levels. *Participatory Educational Research*, *6*(2), 1–14. https://doi.org/10.17275/per.19.8.6.2

Karp, T., & Maloney, P. (2013). Exciting young students in grades K–8 about STEM through an afterschool robotics challenge. *American Journal of Engineering Education*, *4*(1), 39–54. https://files.eric.ed.gov/fulltext/EJ1057112.pdf

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, *41*(4), 245–255. https://doi.org/10.1007/s10643-012-0554-5

Kert, S. B., Erkoç, M. F., & Yeni, S. (2020). The effect of robotics on six graders' academic achievement, computational thinking skills and conceptual knowledge levels. *Thinking Skills and Creativity*, *38*, Article 100714. https://doi.org/10.1016/j.tsc.2020.100714

Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the automatic recognition of computational thinking for adaptive visual language learning. In C. Hundhausen, E. Pietriga, P. Díaz, & M. B. Rosson (Eds.), *Proceedings: 2010 IEEE symposium on visual languages and human-centric computing* (pp. 59–66). IEEE. https://doi.org/10.1109/VLHCC.2010.17

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://users.soe.ucsc.edu/~linda/pubs/ACMInroads.pdf

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology*, *25*(6), 860–876. https://doi.org/10.1007/s10956-016-9628-2

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K–12?. *Computers in Human Behavior*, *41*, 51–61. https://psycnet.apa.org/doi/10.1016/j.chb.2014.09.012

McMillan, J. H., & Schumacher, S. (2013). *Research in education: Evidence-based inquiry* (7th ed.). Pearson.

Mingo, W. D. (2013). *The effects of applying authentic learning strategies to develop computational thinking skills in computer literacy students* [Doctoral dissertation, Wayne State University]. Digital Commons @ Wayne State. https://digitalcommons.wayne.edu/cgi/viewcontent.cgi?article=1673&context=oa_dissertations

Mitnik, R., Nussbaum, M., & Soto, A. (2008). An autonomous educational mobile robot mediator. *Autonomous Robots*, *25*(4), 367–382. https://doi.org/10.1007/s10514-008-9101-z

Morelli, R., De Lanerolle, T., Lake, P., Limardo, N., Tamotsu, E., & Uche, C. (2011, March). Can android app inventor bring computational thinking to K-12. In T. J. Cortina, E. L. Walker, L. Smith King, D. R. Musicant, & L. I. McCann (Eds.), *Proceedings of the 42nd ACM technical symposium on computer science education (SIGCSE'11)* (pp. 1–6). ACM. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=dcc775e72f78e102b611bc3f5561933711bd1fad

Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, *68*(1), 463–484. https://doi.org/10.1007/s11423-019-09708-w

Nugent, G., Barker, B., Grandgenett, N., & Welch, G. (2016). Robotics camps, clubs, and competitions: Results from a US robotics project. *Robotics and Autonomous Systems*, *75*, 686–691. https://doi.org/10.1016/j.robot.2015.07.011

Pala, F. K., & Mıhçı Türker, P. (2021). The effects of different programming trainings on the computational thinking skills. *Interactive Learning Environments*, *29*(7), 1090–1100. https://doi.org/10.1080/10494820.2019.1635495

Papert, S. A. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books. https://dl.acm.org/doi/pdf/10.5555/1095592

Pedhazur, E. J., & Schmelkin, L. P. (2013). *Measurement, design, and analysis: An integrated approach*. Psychology Press. https://doi.org/10.4324/9780203726389

Perlis, A. (1962). The computer in the university. In M. Greenberger (Ed.), *Computers and the world of the future* (pp. 180–219). MIT Press.

Petre, M., & Price, B. (2004). Using robotics to motivate "back door" learning. *Education and Information Technologies*, *9*(2), 147–158. https://doi.org/10.1023/B:EAIT.0000027927.78380.60

Piaget, J. (1936). *Origins of intelligence in the child*. London: Routledge & Kegan Paul.

Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science, 45*(5), 583–602. https://doi.org/10.1007/s11251-017-9421-5

Rubinstein, A., & Chor, B. (2014). Computational thinking in life science education. *PLoS computational biology*, *10*(11), Article e1003897. https://doi.org/10.1371/journal.pcbi.1003897

Selby, C., & Woollard, J. (2013). *Computational thinking: The developing definition* (356481). University of Southampton Institutional Repository. https://eprints.soton.ac.uk/356481/

Sullivan, F. R., & Heffernan, J. (2016). Robotic construction kits as computational manipulatives for learning in the STEM disciplines. *Journal of Research on Technology in Education*, *48*(2), 105–128. https://doi.org/10.1080/15391523.2016.1146563

Sun, L., Hu, L., & Zhou, D. (2022). Programming attitudes predict computational thinking: Analysis of differences in gender and programming experience. *Computers & Education*, *181*, Article 104457. https://doi.org/10.1016/j.compedu.2022.104457

Sysło, M. M., & Kwiatkowska, A. B. (2013, February). Informatics for all high school students. In I. Diethelm & R. T. Mittermeir (Eds.), *Informatics in schools. Sustainable informatics education for pupils of all ages: 6th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 43–56). Springer. https://doi.org/10.1007/978-3-642-36617-8_4

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: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Wing, J. M. (2011). Research notebook: Computational thinking—What and why. *The Link Magazine, 6*, 20–23. https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Witherspoon, E. B., Higashi, R. M., Schunn, C. D., Baehr, E. C., & Shoop, R. (2017). Developing computational thinking through a virtual robotics programming curriculum. *ACM Transactions on Computing Education (TOCE)*, *18*(1), Article 4, 1–20. https://doi.org/10.1145/3104982

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational Thinking for All: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yilmaz Ince, E., & Koc, M. (2021). The consequences of robotics programming education on computational thinking skills: An intervention of the Young Engineer's Workshop (YEW). *Computer Applications in Engineering Education*, *29*(1), 191–208. https://doi.org/10.1002/cae.22321

Appendix

Sample Robotic Activities

Activity Name: 3m Competition

Purpose of the Activity: To ensure that students who examined the relationship between diameter and circumference in the previous activity work in a more motivated way, and to teach or reinforce the relationship between the road taken and the circumference of the wheel by doing and experiencing.

Duration of the Activity: 45 minutes

Materials to be Used: 5 Lego Mindstorms Education Set, 5 expansion sets, 5 computers, black electrical tape

Number of Participants: 20 (5 groups)

How the activity was conducted:

Competition: A competition is organized to make the participants work in a more fun way. Before the competition, the purpose of the competition is clearly explained to the students. It is emphasized that the competition is just for fun, and the result should not be evaluated in any other way.

A start line is set and a finish line is marked 3m away from the start line. Participants are asked to program their robots to stop at the closest distance to this line. It is stated that they must find the number of laps required for the program by calculating, and that they will not be allowed to make any attempts.

This competition is repeated for the other two wheels.

Activity Name: Ultrasonic Sensor

Purpose of the Activity: The aim of the activity is to introduce the concept of sensors, to enable students to establish a relationship between our sensory organs and sensors, and to program a robot that avoids obstacles using an ultrasonic sensor.

Duration of the Activity: 45 minutes

Materials to be Used: 5 Lego Mindstorms Education Set, 5 expansion sets, 5 computers

Number of Participants: 20 (5 groups)

How the activity was conducted:

Ultrasonic sensor: Students are introduced to the ultrasonic sensor and how it works. During the lecture, they are asked how bats detect objects. It is explained that the ultrasonic sensor of the robot works in a similar way.

How to use the ultrasonic sensor while programming is explained in the EV3 software interface. A simple to complex structure is followed. First, it is explained how to use the ultrasonic sensor with the wait for block. Then examples of transferring the data from the sensor to another block (wire) are given.

In addition, the loop block that students will use during the activity and how to use it will be explained.

Applications related to the ultrasonic sensor are made, such as a robot that turns back when it sees an obstacle or a robot that prints the distance of an object on the screen when approaching an object.

Authors

Serhat Altıok, is a research assistant at Kırıkkale University in Turkey. Dr. Altıok's expertise and research interests include coding instruction, robotics, 3D modeling and printing, artificial intelligence, technology integration in teacher education, instructional design, and material development. His focus is to rove teaching and learning processes through technical and pedagogical approach integration.
Email: serhataltiok@hotmail.com *ORCID*: https://orcid.org/0000-0001-6656-8692

Memet Üçgül, is an associate professor at Kırıkkale University in Turkey. Dr. Üçgül's fields of expertise and research interests include robotics, 3D modeling, and printing with a focus on developing innovative solutions and applications to enhance both technical skills and practical learning outcomes.
Email: memet3gul@gmail.com *ORCID*: https://orcid.org/0000-0001-5462-0449