Journal of Cases on Information Technology
Volume 27 • Issue 1 • January-December 2025

Effects of Robotics Programming on Enhancing Computational Thinking Among Middle School Students in Saudi Arabia

Wafa Ahmad Alalawi
https://orcid.org/0000-0001-6421-775X
Princess Nourah Bint Abdulrahman University, Saudi Arabia

Mohd Nihra Haruzuan Bin Mohamad Said
https://orcid.org/0000-0001-9103-9530
Universiti Teknologi Malaysia, Malaysia

Mohd Fadzil Bin Abdul Hanid
https://orcid.org/0000-0003-2884-6393
Universiti Teknologi Malaysia, Malaysia

ABSTRACT

The study aimed to investigate the computational thinking traits of third-grade middle school students while using RoboMind for programming within a project-based learning setup. It also aimed to determine if incorporating RoboMind into project-based instruction improved students' computational thinking skills, attitudes toward programming, and programming performance based on project outcomes. Implemented through a quasi-experimental design with 138 students from two Saudi Arabian schools, the results indicated a significant improvement in computational thinking skills and positive changes in attitudes toward programming post-RoboMind implementation. The study identified four primary dimensions and seven sub-dimensions of computational thinking, emphasizing its diverse aspects. Furthermore, notable correlations were found between RoboMind project scores and post-RoboMind computational thinking scores in the experimental group. These findings advocate for the integration of RoboMind into programming education, especially within Saudi Arabian educational institutions.

KEYWORDS

RoboMind, Computational Thinking, Project Based Learning, Robotic Programming

INTRODUCTION

The application of educational robotics refers to the use of computers as tools to help students acquire, analyze, model, and control different physical processes, enabling children to gain skills, such as critical thinking, teamwork, scientific observation, and planning. Adopting educational robotics in the learning environment presents learners with knowledge that encourages their creativity and allows them to realistically play with new ideas and technologies, allowing students to understand advanced concepts, such as in the artificial intelligence field, cognition, and simulation. The learning approach of educational robotics also sets learners on the path to acquiring problem-solving skills and raises their level of social interaction and creativity. Additionally, the interaction with robotic productions offers students a chance to be realistic in problem-solving since these procedures are

DOI: 10.4018/JCIT.371403

followed in solving real-life problems (Karatrantou & Panagiotakopoulos, 2008). Furthermore, Bilgic & Dogusoy (2023) discovered that incorporating block-based programming activities in lessons can enhance the learning experience for students by allowing them to actively participate and apply their knowledge in real-world scenarios. Additionally, the use of gamified activities can serve as a source of motivation for students toward both the programming activities and the course. Therefore, the adoption of educational robotics is excellent in helping students to discover and use diverse notions associated with scientific concepts, programming languages, and technology, while using an interdisciplinary approach in education (Karatrantou & Panagiotakopoulos, 2012).

The primary channel used in teaching and knowing the core programming structures include the user interface, programming language, and environment (Mason & Cooper, 2014). Regardless of the programming language taught or aspects included in learning programming languages, it is essential for students to learn the construction of algorithms using the core structures, such as selection, and iteration, as well as the mechanics of compiling (Dijkstra, 1972). This results in the teaching programming of educational robotics being complex and challenging to achieve success (Denning & McGettrick, 2005; McCracken et al., 2001). For instance, the challenges are attributed to the simultaneous interaction of different concepts. Students must understand the problem statement, navigate syntax rules, construct algorithms semantics, and navigate the programming interface to compile and execute a program (Jenkins, 2002).

Some argue that students in middle schools lack programming skills and lack the willingness to be immersed in coding compared to undergraduates. However, the generation's views have changed, calling for reforms to the existing education system. Several studies have proposed the incorporation of computational thinking (CT). For example, Voogt et al. (2014) proposed the inclusion of CT in the curriculum, considering the importance of the subject in developing analytical and thinking skills. The widely proposed topic to be added to CT is computer programming. However, the issue of using the educational robot (Shim et al., 2017) is actually simplified by programming complexity, particularly for students.

RoboMind in Enhancing CT Among School Students

The RoboMind application, often utilized in educational contexts, employs its simple scripting language to facilitate beginners' understanding of computer science (CS) through robot simulation programming(Nofitasari et al., 2017). Robotics programming is gaining the attention of educators and researchers because it allows the building and manipulation of tangible real-world creations through programming, which can foster motivation, self-ownership over learning, and heightened interest in science, technology, engineering, and mathematics (STEM)-related subjects (Bers et al., 2014; Ucgul & Cagiltay, 2014). The introduction of computer programming into schools is also driven by the potential of CT, which is a transdisciplinary concept with broad implications for problem-solving in any context (Kafai et al., 2014; Lye & Koh, 2014). CT comprises three dimensions: concepts, practices, and perspectives, but most studies in schools focus only on the concept dimension and overlook the practices and perspectives dimensions (Brennan & Resnick, 2012; Lye & Koh, 2014). This narrow focus is a significant gap in the current literature.

Robotics programming offers opportunities to highlight the practices and perspectives dimensions of CT as a powerful mindset for problem-solving across various subjects through task-based and project-based learning (PjBL) activities (Bers et al., 2014). Research conducted by Yang et al. (2023) has shown that using robot programming can positively affect the development of other cognitive abilities, including executive functions that assist students in formulating, analyzing, and solving problems. RoboMind software is a programming assistance tool that helps in learning and advancing programming concepts, problem-solving, and critical thinking (Yuana & Maryono, 2016). Recently, there has been a renewed interest in integrating computer programming and robotics programs, like RoboMind, in educational environments (Kafai et al., 2014). This interest is not new and has resurfaced in cycles over the last few decades.

CT Characteristics in Saudi Arabia Schools

Recently, there has been a growing emphasis on the relationship between CT and programming, with numerous studies highlighting its importance (Alyahya & Alotaibi, 2019; Ferreira et al., 2017). CT, as defined by Wing (2016), is a problem-solving approach that involves breaking down complex problems into smaller, more manageable components to develop a solution using a logical and algorithmic approach. While CT skills have been extensively studied in many developed countries, research in Saudi Arabia has been limited.

The Ministry of Education in Saudi Arabia recognizes the importance of CT in education, particularly in computer programming, and has identified the enhancement of CT skills as a key strategic objective aligned with Saudi Vision 2030 goals (Government of Saudi Arabia, 2016). The Ministry of Education has launched the courses schooling system, a new schooling scheme with a national curriculum across all subjects and grade levels, including CS curriculums. The CS curricula are based on the U.S. Computer Science Teachers Association K-12 CS standards (Al Salman et al., 2013) that include five essential and complementary standards, with CT being considered a fundamental skill. The Ministry of Education has partnered with Tatweer Co. for Educational Services to create vocational education workshops for CS school teachers to familiarize them with the new CS national curriculum. While the workshops are mandatory for all CS school teachers, many still lack adequate CT knowledge. This underscores the urgent need for professional development programs that can support teachers in implementing CT effectively.

PjBL as an Instructional Design of Robotics

Over the years, PjBL has been a popular approach to teach robotics, and numerous studies have investigated its effectiveness. One of the pioneers in developing robotics kits for children was Papert, who emphasized the significance of active and personalized learning, exploration, and creation in his constructionism theory (Kafai et al., 2014). The student-centred approach, thematic and topical focus, and research-based methodology of PjBL make it an appropriate instructional intervention in robotics studies, given the nature of robotics instruction. When robotics is taught in a PjBL environment, it allows for the exploration of how educational technology, such as robotics, can improve learning experiences by promoting collaboration, cooperation, and problem-solving within real-life contexts (Ching et al., 2018; Nugent et al., 2010). In more formal educational settings, project-based robotics instruction offers several advantages over task-based instruction, including the possibility for students to take charge of their learning with the teacher serving as a facilitator. Project-based robotics courses can motivate and engage students in STEM subjects by providing authentic learning scenarios that allow students to take charge of their learning process (Bers et al., 2014). This engagement can lead to novel opportunities and pathways for learning that students may not have discovered without exposure to robotics. Moreover, project-based robotics instruction has attracted students who may not have demonstrated an interest in robotics alone because project-based robotics is connected to other academic disciplines, especially STEM (Alimisis, 2013).

PROBLEM STATEMENT

Recently, many research studies have emphasized the significance of CT (CT) in programming, made clear by the growing number of studies that concentrate on CT and its relation to programming (Alyahya & Alotaibi, 2019). CT skills have been extensively studied in many developed countries. In Saudi Arabia, however, research of such nature have been limited and there is also a certain knowledge gap in truly understanding the essence of computing as a field of science (AlHumoud et al., 2014). For instance, in Saudi Arabia, improving CT skills is one of the strategic objectives of the Ministry of Education (Government of Saudi Arabia, 2016).

The ability of programming can lead students to solve problems, to predict solutions and tests them, and then to manipulate those solutions to achieve program goals, which lead to CT. According to Brandell (2010), problem-solving is characterized as a cognitive–affective–behavioral procedure that involves individuals (or groups) striving to recognize, uncover, or create effective strategies for managing challenges encountered in daily life. Failure to provide opportunities that satisfy students' curiosity or challenge their cognitive abilities may result in negative consequences, including the development of dependent people. Thus, it is essential to adopt a technology system that fosters cognitive stimulation by allowing students to independently find information, solve problems, and exercise autonomy in decision-making.

It is also important to create bright classrooms that allow for cognition stimulation. This is because creating an appropriate learning environment helps students develop their skills, including their ability to use CT to solve problems. As of today, the significance of learning computer skills has been widely acknowledged, and educators are dealing with digital natives. Hence, it is imperative to consider our education system and surpass the mere instruction of basic computer skills, such as accessing software or educational games that are already well-known by children, more than their parents or even their teachers (Alyahya & Alotaibi, 2019; Romero et al., 2017).

Many studies advise the development of rich programming environments for use with younger students to improve and motivate their learning in programming, especially when integrating programming with other skills, such as analytic thinking and creativity (Liu et al., 2013). To date, there are many programming environments, such as the RoboMind programming language (Yuana & Maryono, 2016), Scratch (Resnick et al., 2009), and Alice (Dann et al., 2006). These environments can help students learn programming successfully (Liu et al., 2013). Recently, RoboMind education has designed cross-curriculum tools that can be integrated into the classroom. RoboMind computing scheme of work helps teachers deliver coding concepts using a real-world context and through robotics programming. Several studies have shown that using robotics "can provide a perceptual space to ground abstract concepts and provide experience with errors caused by physical artifact" (Liu et al., 2013, page 433). When students use robotics in programming, they can immediately see the result of their commands and change them according to those results. This allows fast feedback and boosts their confidence in robotics programming. A relationship is then established between the students and robotics, which influences their students' attitude toward learning programming.

RESEARCH QUESTIONS

The current study aimed to explore the following research questions:

1. What are the characteristics of CT exhibited by third-grade middle school students when learning programming?
2. What are the effects of using RoboMind with PjBL to learn programming on third-grade middle school students' programming proficiency, in terms of improving students' attitudes toward learning programming?
3. To what extent does using RoboMind with PjBL effective in enhancing third-grade middle school students' programming skills?

RESEARCH HYPOTHESIS

The present study addressed the research questions by investigating the following research hypotheses:

H1. There is no statistically significant difference at the level of 0.05 between the middle-school students' CT mean scores of pre/post-RoboMind use.
H2. There is no statistically significant difference at the level of 0.05 between the scores of the student's attitude toward programming mean scores of pre/post-RoboMind in the experimental group.
H3. There is no statistically significant relationship at the level of 0.05 between the scores of the RoboMind final project and CT scores of the post-RoboMind in the experimental group.

CONCEPTUAL FRAMEWORK

The conceptual framework of The current study organizes theories and the study's variables to accomplish the study objectives. The current study used a PjBL approach with robotics to introduce and teach programming to students to help them develop their CT skills and improve their attitudes toward learning programming. Within the 21st century, CT is a fundamental skill required for every student (Wing, 2006). The current study believed that students' performance can be improved when they develop their CT skills and adopt a positive attitude toward learning programming. According to Munson et al. (2011), attitudes toward information technology exhibit many characteristics, including (1) confidence, i.e., feeling comfortable in learning programming concepts and having self-confidence; (2) interest, i.e., the ability to innovate when creating new programs; (3) stereotypes, i.e., the idea that programming is more related to men than women; and (4) usefulness, i.e., the beliefs on the value of programming education.

According to Grover and Pea (2013), abstraction, systematic thinking, symbolic representation, algorithmic thinking, modularization, efficiency, and debugging are the most broadly known characteristics of CT (Chen et al., 2017; Wing, 2006; Wing, 2008). The International Society for Technology in Education, in 2015, advocated for CT to have four components: decomposition, pattern recognition, abstraction, and algorithm design. The problems are divided into smaller manageable parts, logically organizing and analyzing data, then representing this data through abstraction, such as models and simulation (Korkmaz et al., 2017). The CT test developed by Román-González (2015) was utilized in the current study to measure the development of CT skills for middle school students. Moreover, Moreno-León et al. (2015) has demonstrated that his CT test was built to assess the same elements or features provided by Grover and Pea (2013). The current study proposed that problem-solving facilitates the development of CT skills. Therefore, the current study employed the RoboMind to facilitate a flexible learning process that supports CT skills, while simultaneously changing students' attitudes toward learning programming.

PHASES OF THE STUDY

This part discusses the phases used in the development of the RoboMind project. The main objective was to determine the fourth research objective: to develop guidelines for implementing RoboMind in learning programming for third-grade students in middle school. Thus, the RoboMind project based on a PjBL model and the constructivism theory based on psychology was developed. The theory pertains to the acquisition of knowledge and learning, stating that experiencing things and finding ways to reflect on personal experiences develop students' understanding and harness their knowledge.

The Context of the Treatment

In the current study, the RoboMind final project involved experimental groups. Students in the experimental groups were in their third grade at a school in Saudi Arabia. They enrolled in the CS course, which was a compulsory course. The RoboMind project was introduced to them during their second semester.

Table 1. Brief description of RoboMind tasks (weekly basis)

<table>
<tr><th>Week</th><th>Task topic</th><th colspan="3">Description</th></tr>
<tr><td>1</td><td>Robots in our life</td><td colspan="3">Give examples of using robots in our daily life</td></tr>
<tr><td>2</td><td>RoboMind basic instructions</td><td colspan="3">2.1 Write the appropriate commands to move the robot in a square shape.
2.2 If you are running from your computer, download the RoboMind program and do the following challenge:
But if you are running from your smart device, a phone, or a palm: From RoboMind Academy, do the first simulation challenge.</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Loops commands</td><td colspan="3">3.1 Write the appropriate commands to continuously move the robot in a square shape without stopping.
3.2 Use loops to shortcut the following codes</td></tr>
<tr><td>forward (3)
right
forward (3)
left
forward (3)
right</td><td>forward (3)
right
forward (3)
right</td><td>forward (3)
left
forward (3)
right
forward (3)</td></tr>
<tr><td>4</td><td>Paint commands</td><td colspan="3">4.1 Write the code for the robot so that the letter of your first name in English is written in black and implement it on the program, photo the implementation, and attach it.</td></tr>
<tr><td>5</td><td>Grab commands</td><td colspan="3">5.1 Does the robot do according to the commands in the picture?
True or False
5.2 The commands in the picture enable the robot to reach the white point.</td></tr>
<tr><td>6</td><td>See commands and logical expressions</td><td colspan="3">Activity</td></tr>
<tr><td>7</td><td>Final project</td><td colspan="3">Make the robot walk on the white line, and at the end of the line, make the robot stop moving and end the program. Execute the program on the RoboMind, save the file in your name and send it.</td></tr>
</table>

The current study aimed to improve students' attitudes toward learning programming and help them develop their CT skills by implementing a PjBL approach with robotics when introducing and teaching programming to students (see Figure 1 to Figure 3). The development of CT skills among students fosters positive attitudes toward learning programming, thereby enhancing their achievement.

Figure 1. RoboMind interface

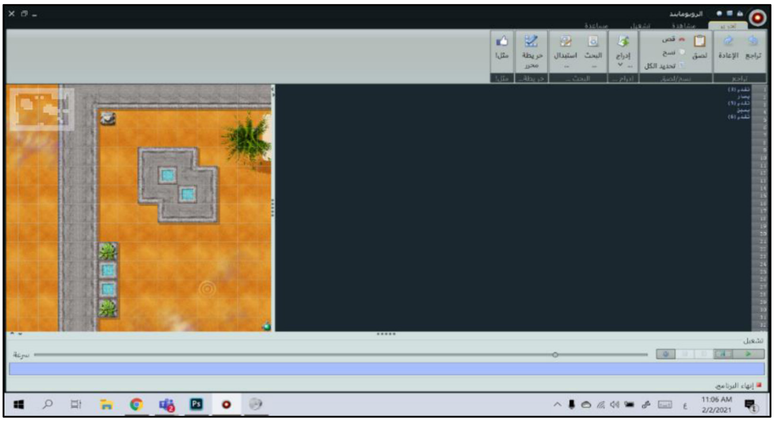


Figure 2. RoboMind movement, final position

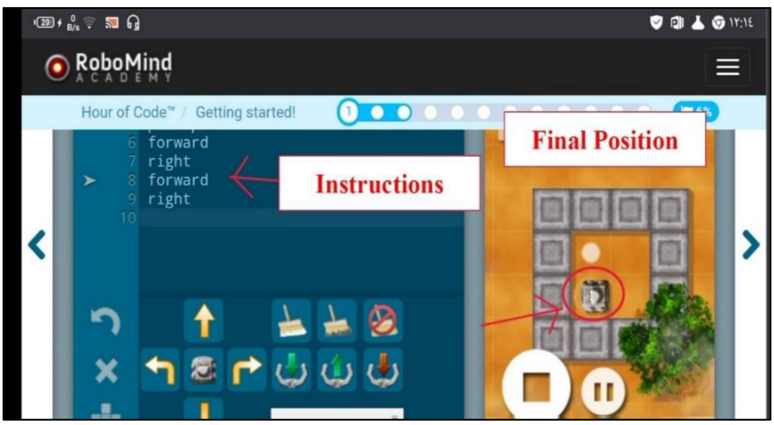


Figure 3. Final RoboMind project with the instructions written by students

METHODOLOGY

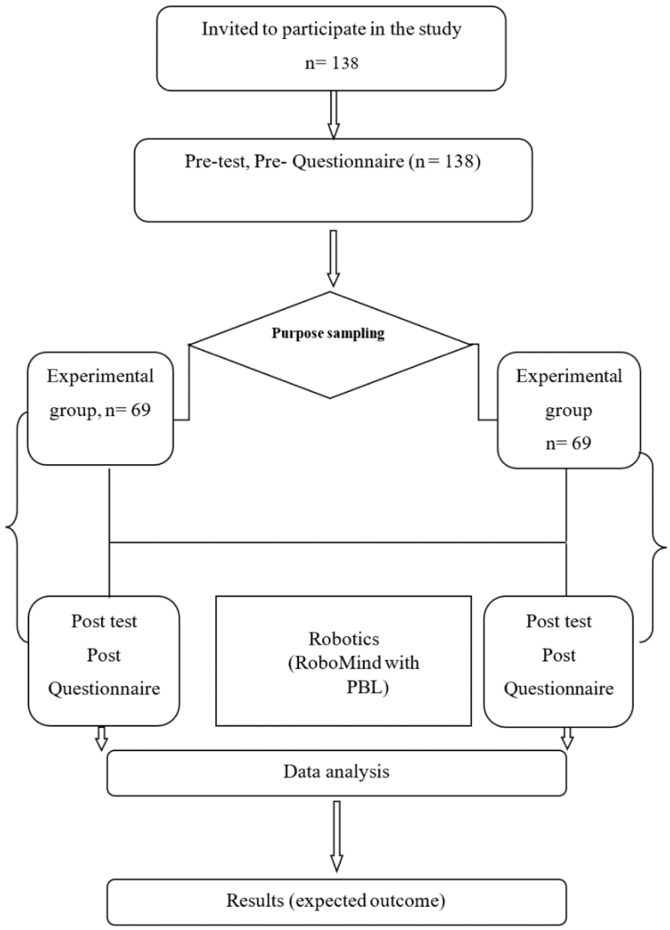
Research Design

Based on the research objectives and the inability to manipulate the conditions and experiences of the study participants, a quasi-experimental design was appropriate (Creswell & Zhang, 2009). Quasi-experimental methods are typically used to evaluate the effectiveness of a program with non-randomized participants (Creswell & Zhang, 2009). The study focused on educational programs and the selection of schools was based on accessibility. A pre/post-experiment design was employed to study two experimental groups from different schools, as the inclusion of a control group was impractical and unethical (Shadish et al., 2002). Researchers are faced with limited resource situations, particularly during the COVID-19 pandemic, that require the recruitment and management of multiple groups, and researchers also have limited time, funding, and access to participants, thus researchers choosing to focus on treatment groups and evaluate the effects of interventions within that group. Due to the pandemic, the study was conducted online.

Participants (Sample)

The experimental group experienced unit 4, smart devices and robots in RoboMind. This unit was chosen for its programming concepts and the utilization of RoboMind. The sample of the current study consisted of 138 students from two schools in Saudi Arabia. The students selected in this study were enrolled in a compulsory course, which is the CS course, in the 3rd grade in Saudi Arabia school. There were two separate sections in each school, each with 69 students. A total of 138 students were included based on the number of students in each section in Saudi schools. Figure 4 illustrates the flowchart of the study.

Figure 4. Flowchart of the intervention of the study



Note. PBL = project based learning.

Instruments

In order to accomplish the research objectives, the following tools were administered to the experimental group prior to and upon completion of the course: (1) a CT assessment (Román-González et al., 2017) to gauge students' proficiency in CT skills and a mini project utilizing RoboMind to evaluate students' fundamental programming knowledge, (2) a survey to assess students' attitudes toward utilizing educational robotics (Munson et al., 2011), and (3) the CT evaluation to prepare for and perform t-test analysis (conducted twice for both the experimental and control groups). The use of RoboMind is free under the student license, which requires online registration, and students must adhere to all intellectual property rights. The service and desktop client are owned by the company, and the user has no rights in or to the service, desktop client, nor content other than the right to use them in accordance with the terms of this license.

Validity

Validity pertains to the degree to which an instrument accurately measures the intended construct. In the context of an agent, validity is the extent to which a tool aligns with the abstract concept being investigated. A high level of validity implies the absence of consistent errors in the measuring instrument. When an agent is valid, it precisely captures the concept it is designed to measure.

Therefore, careful attention is required when selecting a sample and designing research in order to achieve high validity. In the current study, experts and supervisors in the bidding and tendering environment reviewed the modified questionnaire, as well as the study methodology.

Questionnaire Statistical Validity

Two statistical tests were employed to validate the study questionnaire. The first test was to confirm criterion-related validity. The test utilized the Spearman test to determine the correlation coefficient between each item in the field and the total. The second test was to confirm structure validity. The Spearman test was also employed to evaluate the questionnaire's overall validity and the validity of each field. This test measured the correlation coefficient between one field and all the other areas of the questionnaire that shared the same level on a comparable scale.

Internal Consistency

To measure the internal consistency of the questionnaire, a pilot sample of 50 questionnaires was used to calculate the correlation coefficients between each paragraph in a given field and the entire field. The results revealed the p-values and correlation coefficients for each field item. If the p-values were less than 0.01, the correlation coefficients for the field would be considered significant at $\alpha = 0.01$. Based on these results, it was inferred that the paragraphs within this field were valid and consistent in measuring the intended construct.

Instrument Reliability

The reliability of an instrument refers to the consistency of the instrument in measuring the attribute it is intended to measure. To assess the reliability of the test, instrument reliability was administered to the same sample of individuals on two occasions. The scores were then analyzed by calculating the reliability coefficient. In general, a reliability coefficient of 0.7 or higher is considered acceptable. To account for the challenging circumstances faced by the participants, a two-week interval within the same month was set between the two tests.

Data Collection and Analysis

The study used online- based sets of questionnaires. An online survey data was collected via the Google forum online survey. The overall duration of survey data collection took six months; then, the pre-experiment data collection took another six months. The research was moved from face-to-face to online-based classes based on the COVID-19 pandemic. This process went through a few steps. Firstly, Saudi educational institutions moved their courses into online courses as a trial period after all schools were shut down. All schools were officially informed to continue their academic studies online. The data analysis methods are illustrated in Table 2.

Table 2. Data analysis methods

Research Question	Instrument	Data Analysis
(1) What are the characteristics of CT exhibited by third-grade middle school students when learning programming?	**CT-TEST**	**- Descriptive statistics. - Normality tests. - Paired samples t-test. - Eta Squared.**
(2) What are the effects of using RoboMind with project-based learning to learn programming on third-grade middle school students' programming proficiency, in terms of: (a) Improving students' attitudes toward learning programming?	**Attitude questionnaire CT-TEST**	**- Descriptive statistics. - Normality tests. - Paired samples t-test. - Eta Squared.**
(3) To what extent does using RoboMind with project-based learning effective in enhancing third-grade middle school students' programming skills?	**Attitude questionnaire CT-TEST**	**-Pearson correlation coefficients.**

Note. CT = computational thinking.

FINDINGS

What Are the Characteristics of CT Exhibited by Third-Grade Middle School Students When Learning Programming?

To answer the first research question, the data was presented as in Table 3.

Table 3 illustrates the descriptive statistics of CT. The results showed that the standard deviation of the total CT pre-test was 2.95, while the standard deviation for the entire CT in the post-test was 6.80. Thus, according to the pre/post-CT tests, CT skills increased over the course of this experiment.

Table 3. Descriptive statistics of computational thinking (CT) (n = 138)

Test	Variable	Minimum	Maximum	Mean	Std. Deviation
Pre-test	Basic directions and sequences	0	4	1.73	1.01
	Loops	0	6	2.83	1.48
	Conditionals	0	7	2.91	1.63
	Functions	0	4	1.08	1.04
	Total CT	1	13	8.55	2.95
Post-test	Basic directions and sequences	0	4	2.66	1.04
	Loops	1	8	4.59	2.11
	Conditionals	1	12	5.80	3.48
	Functions	0	4	2.12	1.23
	Total CT	6	28	15.17	6.80

Note. CT = computational thinking.

It is evident from Table 3 that the multivariate data of the current study conformed to a normality distribution, indicated by the univariate normal test for the continuous variables. In particular, the test results indicated the standard univariate distribution of the scores of the five variables of CT in the pre/post-tests due to the non-statistical significance of the *Z*-scores for skewness and kurtosis coefficients in the case of all variables, except for functions dimension. Agostino and Pearson (1973) explain that the degree of the variable is normally distributed when *Z*-score is not significant (see Table 3).

Table 4. Normality tests of computational thinking (CT) (n = 138)

Test	Variable	Skewness				Kurtosis			
		Statistic	Std. Error	Z-Score	P-Value	Statistic	Std. Error	Z-Score	P-Value
Pre-test	Basic directions and sequences	0.05	0.21	0.27	0.79	-0.33	0.41	0.81	0.42
	Loops	0.02	0.21	0.10	0.93	-0.27	0.41	-0.621	0.54
	Conditionals	0.07	0.21	0.37	0.72	-0.28	0.41	-0.631	0.59
	Functions	0.41	0.21	1.96	0.05	-0.64	0.41	-2.15	0.03
	Total CT	-0.05	0.21	-0.24	0.81	-0.21	0.41	0.41	0.68
Post-test	Basic directions and sequences	-0.23	0.21	-1.14	0.26	-0.57	0.41	-1.80	0.078
	Loops	-0.08	0.21	-0.42	0.68	-0.54	0.41	-1.63	0.108
	Conditionals	-0.04	0.21	-0.18	0.85	-0.56	0.41	-1.76	0.08
	Functions	-0.04	0.21	-0.18	0.86	-0.66	0.41	-2.26	0.028
	Total CT	-0.08	0.21	-0.40	0.69	-0.43	0.41	-1.19	0.24

Note. CT = computational thinking.

Table 5 illustrates the effect size that corresponded to the eta squared values. Cohen (1988) presented the rules for effect size corresponding to the ETA squared values, as in the following table.

Table 5. Effect size corresponding to the eta squared values

Eta Squared Values	Effect Size
Eta squared < 0.059	Weak
0.059 ≤ Eta squared < 0.138	Medium
0.138 ≤ Eta squared < 0.232	Large
0.232 ≤ Eta squared	Very large

According to Table 6, null hypothesis, H1 is rejected ($p < 0.05$). The results of the t-test are significant, indicating that there is a difference between the average scores achieved by students in the pre/post-tests across all dimensions of CT, including basic directions and sequences, loops, conditionals, and functions. Furthermore, the overall scores of CT indicated that the mean of post-test scores was superior in all instances.

Table 6. The results of the paired samples t-test results between the pre/post-tests in computational thinking (CT) among middle school students in the experimental group (n = 138)

	Variable	Test	Mean	Std. Deviation	T-Test	df	Sig. (2-tailed)	Eta Squared η²	Effect Size
1	Basic directions and sequences	Pre-test	1.73	1.01	8.93	137	0.01	0.368	Very large
		Post-test	2.66	1.04					
2	Loops	Pre-test	2.83	1.48	9.87	137	0.01	0.416	Very large
		Post-test	4.59	2.11					
3	Conditionals	Pre-test	2.91	1.63	9.06	137	0.01	0.374	Very large
		Post-test	5.80	3.48					
4	Functions	Pre-test	1.08	1.04	8.23	137	0.01	0.331	Very large
		Post-test	2.12	1.23					
	Total CT	Pre-test	8.55	2.95	12.15	137	0.01	0.519	Very large
		Post-test	15.17	6.80					

Note. CT = computational thinking.

The results indicate that the mean scores for all dimensions of CT, including basic directions and sequences, loops, conditionals, and functions, as well as the overall CT score, were significantly higher in the post-test compared to the pre-test. The study found that the experimental group of middle school students exhibited a substantial effect size in enhancing their CT skills as a result of using RoboMind, as evidenced by the range of eta squared (η2) values observed, which fell between 0.331 and 0.519. According to Cohen (1988), an eta squared value that exceeds 0.232 is deemed to be a significant effect size. The analysis of eta squared values indicates that RoboMind had a substantial impact on the variance in both the dimension scores and the overall CT scores, accounting for a range of 33.1 percent to 51.9 percent. This suggests that RoboMind is a significant contributor to the explained variance.

What are the Effects of Using RoboMind With PjBL on the Learning of Programming of Third-Grade Middle School Students?

Table 7 illustrates the descriptive statistics pertaining to the students' attitudes toward programming. The results showed that the standard deviation of the pre-test for attitude toward programming was 8.65, while the the post-test was 8.71.

Table 7. Descriptive statistics of attitude toward programming (n = 138)

Test	Variable	Minimum	Maximum	Mean	Std. Deviation
Pre-test	Confidence	4	16	11.04	2.37
	Interest	4	16	11.46	2.92
	Stereotypes	4	12	9.30	1.81
	Usefulness	4	16	11.04	2.71
	Total attitude toward programming	16	56	42.84	8.65

continued on following page

Table 7. Continued

Test	Variable	Minimum	Maximum	Mean	Std. Deviation
Post-test	Confidence	5	16	11.94	2.63
	Interest	4	16	12.67	2.61
	Stereotypes	8	16	11.96	2.33
	Usefulness	5	16	12.36	2.50
	Total attitude toward programming	26	64	48.93	8.71

According to Table 8, multivariate data achieved the assumption of a normality distribution. The results of the univariate normal test for the continuous variables indicated that the univariate distribution of the scores of the five variables of attitude toward programming was standard in the pre/post-tests due to the non-statistical significance of the Z-scores for skewness and kurtosis in the case of all variables.

Table 8. Normality tests of attitude toward programming (n = 138)

Test	Variable	Skewness				Kurtosis			
		Statistic	Std. Error	Z-Score	P-Value	Statistic	Std. Error	Z-Score	P-Value
Pre-test	Confidence	.006	.206	0.03	0.98	-.079	.410	-0.04	0.97
	Interest	-.006	.206	-0.02	0.98	-.281	.410	-0.65	0.52
	Stereotypes	-.041	.206	-0.20	0.84	-.259	.410	-0.57	0.57
	Usefulness	.002	.206	0.01	0.99	-.213	.410	-0.43	0.67
	Total attitude toward programming	.027	.206	0.13	0.90	-.166	.410	-0.29	0.77
Post-test	Confidence	-.175	.206	-0.87	0.39	-.430	.410	-1.19	0.24
	Interest	-.238	.206	-1.17	0.24	-.537	.410	-1.63	0.10
	Stereotypes	-.107	.206	-0.53	0.60	-.594	.410	-1.90	0.06
	Usefulness	-.160	.206	-0.79	0.43	-.523	.410	-1.57	0.12
	Total attitude toward programming	-.179	.206	-0.88	0.38	-.453	.410	-1.28	0.20

According to the findings presented in Table 9, null hypothesis, H^2 is rejected ($p< 0.05$). The results of the t-test are significant, indicating that there is a difference between the mean scores in the pre/post-tests in all dimensions of attitude toward programming, namely confidence, interest, stereotypes, and usefulness. Furthermore, the overall scores of attitudes toward programming indicated a preference for the mean of post-test scores in all cases. The mean scores for all dimensions of attitude toward programming and the overall degree of attitude toward programming in the post-test were statistically higher than their counterparts in the pre-test.

Table 9. Paired samples t-test results between the pre/post-tests in attitude toward programming among middle school students in the experimental group (n = 138)

	Variable	Test	Mean	Std. Deviation	T-Test	df	Sig. (2-tailed)	Eta Squared η^2	Effect Size
1	Confidence	Pre-test	11.04	2.37	4.66	137	0.01	0.137	Medium
		Post-test	11.94	2.63					
2	Interest	Pre-test	11.46	2.92	7.13	137	0.01	0.271	Very large
		Post-test	12.67	2.61					
3	Stereotypes	Pre-test	9.30	1.81	10.7	137	0.01	0.455	Very large
		Post-test	11.96	2.33					
4	Usefulness	Pre-test	11.04	2.71	6.97	137	0.01	0.262	Very large
		Post-test	12.36	2.50					
5	Total attitude toward programming	Pre-test	42.84	8.65	14.22	137	0.01	0.596	Very large
		Post-test	48.93	8.71					

The obtained eta squared η2 value equal to 0.137 indicated that RoboMind had a medium effect size in the confidence dimension of the middle school students in the experimental group. The eta squared value also indicated that RoboMind explained 13.7 percent of the variance in the scores of confidence dimension of the middle school students in the experimental group during the post-test, relative to their pre-test scores. This indicated a moderate variance explained by RoboMind. According to Cohen (1988), the eta squared value lower than 0.138 indicates a moderate effect size. The eta squared η2 values that ranged from 0.262 to 0.596 indicated that RoboMind had a considerable effect size in three dimensions of attitude toward programming (interest, stereotypes, and usefulness) and the overall degree of attitude toward programming among the middle school students in the experimental group. The eta squared values also indicated that RoboMind explained the ratios that ranged from 26.2 percent to 59.6 percent of the variance in the scores of interest, stereotypes, and usefulness dimensions. In addition, there was a significant increase in the overall scores of programming attitude exhibited by the middle school students in the experimental group during the post-test phase as compared to their pre-test scores. This increase was largely attributed to the influence of RoboMind. According to Cohen (1988), a value of eta squared greater than 0.232 signifies a substantial effect size.

To What Extent Does Using RoboMind With PjBL Effective in Enhancing Third-Grade Middle School Students' Programming Skills?

The results presented in Table 10 indicated a significant correlation between the final project scores in RoboMind and all dimensions encompassing basic directions and sequences, loops, conditionals, and functions, as well as the overall score of the CT test in the post-test administered to middle school students. The results indicated a positive correlation between the scores of the dimensions (basic directions and sequences, loops, conditionals, and functions) and the overall score for the CT test, and the scores of the RoboMind final project in the post-test for middle school students. In essence, higher scores in the former are associated with higher scores in the latter. The findings indicated that the third null hypothesis failed to be accepted.

Table 10. The results of pearson correlation coefficients between RoboMind final project scores and computational thinking (CT) scores of post-RoboMind among middle school students in the experimental group (n = 138)

Test	Variables	RoboMind Final Project	
		Pearson Correlation	Sig. (2-tailed)
Post-test	Basic directions and sequences	0.336	0.01
	Loops	0.480	0.01
	Conditionals	0.582	0.01
	Functions	0.420	0.01
	Total CT	0.715	0.01

Note. CT = computational thinking.

DISCUSSION

The findings of the current study indicate that the RoboMind intervention is successful in improving CT skills. The standard deviation of total CT scores increased from 2.95 in the pre-test to 6.80 in the post-test, indicating an improvement in CT skills. According to previous studies, CT skills enable effective problem-solving, system design, and understanding of human behaviors (Basu et al., 2016). Nevertheless, the lack of computational concepts in primary and middle schools has been acknowledged by many researchers (Barr & Stephenson, 2011; Basogain et al., 2018; Basu et al., 2016). The results of the current study present a statistically significant difference in mean scores between pre/post-tests in all dimensions of CT and the overall scores of CT. These findings are consistent with previous studies that have found RoboMind to be an effective and accessible tool for learning programming concepts (García-Peñalvo & Mendes, 2018). For instance, a study exploring the use of robots in a learning environment found that students had a positive attitude toward using robots for learning (Chen et al., 2017).

Furthermore, the mean scores of the middle school students in the experimental group exhibited a statistical increase in the post-test from the pre-test in all dimensions of CT, including basic directions and sequences, loops, conditionals, and functions, and the overall degree of CT. This is evidenced by the eta squared values, which ranged from 0.331 to 0.519, indicating that RoboMind has a significant effect size in all dimensions of CT among the experimental group (Cohen, 1988). Previous literature also supports the importance of finding appropriate abstractions, creating simulations, and defining real-world problems to improve CT skills (Basawapatna et al., 2013; Wing, 2006; Wing, 2008; Yadav et al., 2014).

Robotics can enhance students' problem-solving skills, motivation, and interest in programming. The goal of learning CT is to enhance students' problem-solving skills to be applied in diverse contexts, including programming. CT is beneficial for school students as they are exposed to an overwhelming amount of information, and it helps them understand complex systems and develop their thinking abilities. The difference between computer programming, CT, and algorithmic thinking is critical to understand the necessity of learning CT. Previous studies have shown that PjBL with robotics helps students develop their CT skills and improve their attitudes toward programming (Grover & Pea, 2013; Liu et al., 2013; Munson et al., 2011; J. Wing, 2006; Wing, 2008). The results of the univariate regular test show no significant difference in the attitudes of middle school students toward programming before and after the intervention.

Although robotics has been lauded by initial adopters as a successful addition to classroom instruction; for instance, given that educational robots can enhance classroom learning by promoting student engagement with the subject matter, there exists a dearth of quantitative research on the potential of robotics to enhance programming education. A review of previous studies by Kubilinskiene et al.

(2017) found that most research involving robots in education has focused on physics and mathematics subjects. However, the current study deviates from that trend. Also, most studies on robotics in classrooms have focused on secondary and postsecondary students and have been conducted outside of school settings, in contrast to the current study. In terms of teaching methods to use with robots, Altin and Pedaste (2013) have identified several strategies, including problem-based, constructivist, competition-based, discovery-based, communication-based, and PjBL. Although all approaches can be effective, the current study employed PjBL because it allows students to apply their knowledge and create their digital experiences instead of just consuming information in addition to motivating and promoting student engagement in learning activities (Spolaôr & Benitti, 2017).

In all four aspects of attitude toward programming (confidence, interest, stereotypes, and usefulness), pre-test and post-test scores show a significant difference, with post-test scores being higher in every instance. Previous research has suggested that attitudes positively influence learning and cognition (Korkmaz et al., 2017). As a result, negative attitudes, lack of motivation, and poor perceptions of the program can significantly affect academic performance (Korkmaz et al., 2017). Also, negative attitudes toward learning programming can cause many adolescents to ignore programming as a career (Carter, 2006), especially if they are unaware of the job descriptions of a computer scientist (Grover & Pea, 2013). In short, recognizing students' attitudes toward computer programming is essential in CS education because attitudes can affect students' academic performance and their willingness to embrace programming (Baser, 2013).

Scholars have argued that negative attitudes, motivation, and perceptions are critical factors, which have a profound effect on academic performance, in comparison to other factors (Anastasiadou & Karakos, 2011; Erdogan et al., 2008; Korkmaz et al., 2017). Studies on attitudes toward learning programming have been done at various levels, but most of these studies have been done at the undergraduate level. For example, in the first grade of undergraduate engineering (Korkmaz et al., 2017), this study relied on robotics, discussing the activities that identify learners' attitudes in mastering computer programming, academic achievement, and self-efficacy. Another study was conducted on middle schools; the sample included 320 middle school students who joined in voluntary after school (38%) or elective game programming courses in 2009–2011 (Denner et al., 2014).

According to this study findings, they are in accordance with a study done by Witherspoon and Schunn (2019), who documented that the use of robotics encourages significant involvement of students, as noted in the increased interest of female students learning, increasing their appeal to pursue additional opportunities in mastering programming. Therefore, it is proposed that there is a need to modify the teaching methods and students' attitudes. It is also vital to bring changes to the traditional teaching methods (Gomes et al., 2012). In this sense, we need to change the strategies in teaching programming and change students' attitudes toward programming by using robotics, as this study plans.

Previous research has shown that robotics programming can help students develop problem-solving skills (Çalişkan, 2020; Çetin & Demircan, 2020), promote interest in learning robotic programming (Noh & Lee, 2020), significantly improve CT and creativity (Noh & Lee, 2020), and develop imagination and decision-making skills (Holowka, 2020). Students have also demonstrated positive attitudes toward learning and high motivation to learn about robotics (Kaloti-Hallak et al., 2015). Nevertheless, a study by Álvarez-Herrero (2020) found no consensus on whether using other types of practices beyond floor robots is beneficial for developing CT in early childhood education. Although there were significant differences in engineering design ability and cognitive load, a different study by Zhong et al. (2020) found no significant differences in students' learning attitudes, programming skills, or engagement between virtual and physical robots.

The current study identifies a statistically significant relationship between RoboMind final project scores and all dimensions (basic directions and sequences, loops, conditionals, and functions) and the overall score of the CT test in the post-test. This suggests that the higher the scores in these dimensions and the overall score for the CT test, the higher the scores in the RoboMind final project scores for middle school students in the post-test. These findings are consistent with previous studies (McCracken et al., 2001) that found that robotic programming camps for students are an appropriate pedagogical technique, instrument, and evaluation which can increase students' motivation and lead to significant learning by shaping students' ideas on CS and engaging them in programming interests (Nusayr & Da Silva, 2019). Witherspoon and Schunn (2019) also identified that students in classrooms with teachers who rated CT as a more important instructional goal demonstrate better learning gains on the CT assessment, and these effects are stronger for female students.

CONCLUSION

This research project introduced RoboMind as a learning tool for programming in Saudi Arabian schools. It was the first experiment of its kind conducted in the country to be carried out on students at an early age level. The findings of this study indicate the significance of educators understanding the process of incorporating the tool with the syllabus and proficiently delivering the content. RoboMind imparts knowledge of CT, a crucial skill for the 21st century, teaching students about logic, automation, and technology through the programming of a virtual robot, fostering the development of logical reasoning skills essential for addressing real-world problems. Teachers should also consider the accuracy of the stories they share with students to provide ethical knowledge in return for a higher level of engagement. The current study also emphasizes the importance of teachers sharing information, identifying resources, and acquiring new knowledge. Moreover, the study assessed the students' learning outcomes in relation to the teachers' performance after using the tool. Pre/post-involvement surveys were employed to assess the students' perceptions. Overall, the study contributes to the literature on how third-grade students can develop CT skills through PjBL with RoboMind.

COMPETING INTERESTS

The authors of this publication declare there are no competing interests.

FUNDING STATEMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. Funding for this research was covered by the authors of the article.

PROCESS DATES

February 28, 2025
Received: March 20, 2024, Revision: August 27, 2024, Accepted: September 3, 2024

CORRESPONDING AUTHOR

Correspondence should be addressed to MOHD NIHRA HARUZUAN MOHAMAD SAID (Malaysia, nihra@utm.my]

ACKNOWLEDGEMENTS

The work was supported/funded by the Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS) [grant number FRGS/1/2024/SSI09/UTM/02/4].

REFERENCES

Agostino, R. D., & Pearson, E. S. (1973). Tests for departure from normality. Empirical results for the distributions of b 2 and $\sqrt{}$ b. *Biometrika*, *60*(3), 613–622. DOI: 10.2307/2335012

Al Salman, A., Al-Wakee, S., Mandurah, M., Aloraifi, Y., & Al-Mubarak, A. (2013). *The document of computer curriculum and information technology at the secondary level*. Ministry of Education.

AlHumoud, S., Al-Khalifa, H. S., Al-Razgan, M., & Alfaries, A. (2014). Using app Inventor and LEGO mindstorm NXT in a summer camp to attract high school girls to computing fields. In *2014 IEEE Global Engineering Education Conference (EDUCON)*, 173–177. DOI: 10.1109/EDUCON.2014.6826086

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

Altin, H., & Pedaste, M. (2013). Learning approaches to applying robotics in science education. *Journal of Baltic Science Education*, *12*(3), 365–377. DOI: 10.33225/jbse/13.12.365

Álvarez-Herrero, J. F. (2020). Computational thinking in early childhood education, beyond floor robots. *Education in the Knowledge Society*, *21*. Advance online publication. DOI: 10.14201/eks.22366

Alyahya, D., & Alotaibi, A. (2019). Computational thinking skills and its impact on TIMSS achievement: An instructional design approach. *Issues and Trends in Educational Technology*, *7*(1). Advance online publication. DOI: 10.2458/azu_itet_v7i1_alyahya

Anastasiadou, S., & Karakos, A. (2011). The beliefs of electrical and computer engineering students' regarding computer programming. *International Journal of Technology, Knowledge and Society*, *7*(1), 37–51. DOI: 10.18848/1832-3669/CGP/v07i01/56170

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. DOI: 10.1145/1929887.1929905

Basawapatna, A., Repenning, A., Koh, K. H., & Nickerson, H. (2013). The zones of proximal flow: Guiding students through a space of computational thinking skills and challenges. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research* (pp. 67–74). DOI: 10.1145/2493394.2493404

Baser, M. (2013). Attitude, gender and achievement in computer programming. *Middle East Journal of Scientific Research*, *14*(2), 248–255.

Basogain, X., Olabe, M. Á., Olabe, J. C., & Rico, M. J. (2018). Computational thinking in pre-university blended learning classrooms. *Computers in Human Behavior*, *80*, 412–419. DOI: 10.1016/j.chb.2017.04.058

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning*, *11*(1), 13. Advance online publication. DOI: 10.1186/s41039-016-0036-2 PMID: 30613246

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. DOI: 10.1016/j.compedu.2013.10.020

Bilgic, K., & Dogusoy, B. (2023). Exploring secondary school students' computational thinking experiences enriched with block-based programming activities: An action research. *Education and Information Technologies*, *28*(8), 10359–10384. Advance online publication. DOI: 10.1007/s10639-023-11583-1

Brandell, J. R. (2010). *Theory & practice in clinical social work* (2nd ed.). Sage Publications, Inc.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (pp. 1–25). https://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Çalişkan, E. (2020). The effects of robotics programming on secondary school students' problem-solving skills. *World Journal on Educational Technology: Current Issues*, *12*(4), 217–230. DOI: 10.18844/wjet.v12i4.5143

Carter, L. (2006). Why students with an apparent aptitude for computer science don't choose to major in computer science. *Association for Computing Machinery Special Interest Group on Computer Science Education Bulletin*, *38*(1), 27–31. DOI: 10.1145/1121341.1121352

Çetin, M., & Demircan, H. Ö. (2020). Empowering technology and engineering for STEM education through programming robots: A systematic literature review. *Early Child Development and Care*, *190*(9), 1323–1335. DOI: 10.1080/03004430.2018.1534844

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. DOI: 10.1016/j.compedu.2017.03.001

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

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Routledge.

Creswell, J. W., & Zhang, W. (2009). The application of mixed methods designs to trauma research. *Journal of Traumatic Stress*, *22*(6), 612–621. DOI: 10.1002/jts.20479 PMID: 19960518

Dann, W. P., Cooper, S., & Pausch, R. (2006). *Learning to program with Alice*. Prentice-Hall.

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education*, *46*(3), 277–296. DOI: 10.1080/15391523.2014.888272

Denning, P. J., & McGettrick, A. (2005). Recentering computer science. *Communications of the ACM*, *48*(11), 15–19. DOI: 10.1145/1096000.1096018

Dijkstra, E. W. (1972). The humble programmer. *Communications of the ACM*, *15*(10), 859–866. DOI: 10.1145/355604.361591

Erdogan, Y., Bayram, S., & Deniz, L. (2008). Factors that influence academic achievement and attitudes in web based education. *International Journal of Instruction*, *1*(1), 31–47. https://files.eric.ed.gov/fulltext/ED524139.pdf

Ferreira, M., Duarte, J., Campos, S., Chaves, C., Felizardo, S., & Cardoso, A. P. (2017). Emotional skills and promotion of school success: preliminary data of an intervention programme with parents. In Bekirogullari, Z., Minas, M. Y., & Thambusamy, R. X. (Eds.), *Health and Health Psychology - icH&Hpsy 2017, European Proceedings of Social and Behavioural Sciences* (*Vol. 30*, pp. 310–317). Future Academy., DOI: 10.15405/epsbs.2017.09.30

García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, *80*, 407–411. DOI: 10.1016/j.chb.2017.12.005

Gomes, A., Santos, Á., & Mendes, A. J. (2012). A study on students' behaviours and attitudes toward learning to program. In *Proceedings of the 17th Association for Computing Machinery Annual Conference on Innovation and Technology in Computer Science Education* (pp. 132–137). DOI: 10.1145/2325296.2325331

Government of Saudi Arabia. (2016). Saudi Vision 2030. https://www.vision2030.gov.sa/en/

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

Holowka, P. (2020). Teaching robotics during COVID-19: Machine learning, simulation, and AWS DeepRacer. *17th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2020)*, 227–234. https://files.eric.ed.gov/fulltext/ED626743.pdf

Jenkins, R. (2002). In the present tense: Time, identification and human nature. *Anthropological Theory*, *2*(3), 267–280. DOI: 10.1177/1463499602002003799

Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *Association for Computing Machinery Transactions on Computing Education*, *14*(1), 1–20. DOI: 10.1145/2576874

Kaloti-Hallak, F., Armoni, M., & Ben-Ari, M. (2015). Students' attitudes and motivation during robotics activities. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 102–110). DOI: 10.1145/2818314.2818317

Karatrantou, A., & Panagiotakopoulos, C. (2008). Algorithm, pseudo-code and lego mindstorms programming. In *Proceedings of International Conference on Simulation and Programming for Autonomous Robots/Teaching with Robotics* (pp. 70–79). Didactic Approaches and Experiences., https://www.dei.unipd.it/~emg/downloads/SIMPAR08-WorkshopProceedings/TeachingWithRobotics/karatrantou.pdf

Karatrantou, A., & Panagiotakopoulos, C. (2012). Educational robotics and teaching introductory programming within an interdisciplinary framework. In Jimoyiannis, A. (Ed.), *Research on e-Learning and ICT in Education* (pp. 197–210). Springer., DOI: 10.1007/978-1-4614-1083-6_15

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, *72*, 558–569. DOI: 10.1016/j.chb.2017.01.005

Kubilinskiene, S., Zilinskiene, I., Dagiene, V., & Sinkevičius, V. (2017). Applying robotics in school education: A systematic review. *Baltic Journal of Modern Computing*, *5*(1), 50–69. DOI: 10.22364/bjmc.2017.5.1.04

Liu, J., Lin, C.-H., Potter, P., Hasson, E. P., Barnett, Z. D., & Singleton, M. (2013). Going mobile with app inventor for android: a one-week computing workshop for K-12 teachers. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13)*, 433–438. DOI: 10.1145/2445196.2445324

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. DOI: 10.1016/j.chb.2014.09.012

Mason, R., & Cooper, G. (2014). Introductory programming courses in Australia and New Zealand in 2013 - Trends and reasons. In *Proceedings of the Sixteenth Australasian Computing Education Conference* (*Vol. 148*, pp. 139–147). https://doi.org/DOI: 10.5555/2667490.2667507

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B., Laxer, C., Thomas, L., Utting, I., & Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *Association for Computing Machinery Special Interest Group on Computer Science Education Bulletin*, *33*(4), 125–180. DOI: 10.1145/572133.572137

Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *Revista de Educación a Distancia (RED), 46*, 1–23. https://www.um.es/ead/red/46

Munson, A., Moskal, B., Harriger, A., Lauriski-Karriker, T., & Heersink, D. (2011). Computing at the high school level: Changing what teachers and students know and believe. *Computers & Education*, *57*(2), 1836–1849. DOI: 10.1016/j.compedu.2011.03.005

Nofitasari, A., Yuana, R. A., & Maryono, D. (2017). The use of RoboMind application in problem based learning model to enhance the student's understanding on the conceptual programming algorithm. *Indonesian Journal of Informatics Education*, *1*(1), 9. DOI: 10.20961/ijie.v1i1.4170

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. DOI: 10.1007/s11423-019-09708-w

Nugent, G., Barker, B., Grandgenett, N., & Adamchuk, V. I. (2010). Impact of robotics and geospatial technology interventions on youth STEM learning and attitudes. *Journal of Research on Technology in Education*, *42*(4), 391–408. DOI: 10.1080/15391523.2010.10782557

Nusayr, A., & Da Silva, R. (2019). The use of educational robots to engage the youth in computer science: A case study. In *Proceedings 2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)* (pp. 477–481). DOI: 10.1109/LARS-SBR-WRE48964.2019.00090

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

Román-González, M. (2015). Computational thinking test: Design guidelines and content validation. *EDULEARN15 Proceedings*, 2436–2444. International Academy of Technology, Education and Development. DOI: 10.13140/RG.2.1.4203.4329

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

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*(1), 42. Advance online publication. DOI: 10.1186/s41239-017-0080-z

Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). Quasi-experimental designs that either lack a control group or lack pretest observations on the outcome. In *Experimental and quasi-experimental designs for generalized causal inference* (2nd ed., pp. 103–134). Cengage Learning.

Shim, J., Kwon, D., & Lee, W. (2017). The effects of a robot game environment on computer programming education for elementary school students. *IEEE Transactions on Education*, *60*(2), 164–172. DOI: 10.1109/TE.2016.2622227

Spolaôr, N., & Benitti, F. B. V. (2017). Robotics applications grounded in learning theories on tertiary education: A systematic review. *Computers & Education*, *112*, 97–107. DOI: 10.1016/j.compedu.2017.05.001

Ucgul, M., & Cagiltay, K. (2014). Design and development issues for educational robotics training camps. *International Journal of Technology and Design Education*, *24*(2), 203–222. DOI: 10.1007/s10798-013-9253-9

Voogt, J., McKenney, S., Kali, Y., Breuleux, A., Cober, R. D., Slotta, J., Eylon, B. S., Itow, R., Könings, K., Laferrière, T., Linn, M. C., Markauskaite, L., Reeve, R., Sagy, O., So, H. J., Svihla, V., Tan, E., & Matuk, C. (2014). In Polman, J. L., Kyza, E. A., O'Neill, D. K., Tabak, I., Penuel, W. R., Jurow, A. S., O'Connor, K., Lee, T., & D'Amico, L. (Eds.), *Learning and becoming in practice: The International Conference of the Learning Sciences (ICLS) 2014* (*Vol. 1*, pp. 1–14). International Society of the Learning Sciences., https://www.isls.org/icls/2014/downloads/ICLS%202014%20Volume%201%20(PDF)-wCover.pdf

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

Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717–3725. DOI: 10.1098/rsta.2008.0118

Wing, J. (2016). Computational thinking, 10 years later. Microsoft Research Blog. https://www.microsoft.com/en-us/research/blog/computational-thinking-10-years-later/

Witherspoon, E. B., & Schunn, C. D. (2019). Teachers' goals predict computational thinking gainsin robotics. *Information and Learning Science*, *120*(5/6), 308–326. DOI: 10.1108/ILS-05-2018-0035

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *Association for Computing Machinery Transactions on Computing Education*, *14*(1), 1–16. DOI: 10.1145/2576872

Yang, W., Ng, D. T. K., & Su, J. (2023). The impact of story-inspired programming on preschool children's computational thinking: A multi-group experiment. *Thinking Skills and Creativity*, *47*, 101218. DOI: 10.1016/j.tsc.2022.101218

Yuana, R. A., & Maryono, D. (2016). RoboMind utilization to improve student motivation and concept in learning programming. *Proceedings of the ICTTE FKIP UNS*, *2015*, 962–966. https://download.garuda.kemdikbud.go.id/article.php?article=431194&val=9209&title=Robomind%20Utilization%20to%20Improve%20Student%20Motivation%20and%20Concept%20in%20Learning%20Programming

Zhong, B., Luo, W., Li, H., Zhang, Q., Liu, X., Li, W., & Li, Y. (2020). Knowledge, attitudes, and practices towards COVID-19 among Chinese residents during the rapid rise period of the COVID-19 outbreak: A quick online cross-sectional survey. *International Journal of Biological Sciences*, *16*(10), 1745–1752. DOI: 10.7150/ijbs.45221 PMID: 32226294

Mohd Nihra Haruzuan Mohd Said PhD (University of Waikato) is an Associate Professor at Universiti Teknologi Malaysia. He is currently the Information Technology Manager at the Center for Advancement in Digital and Flexible Learning (UTMCDex). His interests lie at the intersection of educational technology, e-learning, online learning, and information and communications technologies in education and academic leadership. He has 15 years of experience in the field of education and technology in the United Kingdom, New Zealand, Australia, and Singapore. His contact information is: School of Education, Universiti Teknologi Malaysia, 81310 Johor, Malaysia, nihra@utm .my. For more information, please visit the following webpage: https://people.utm.my/nihra/.

Wafa Ahmad Alalawi is a lecturer at Princess Nourah Bint Abdulrahman University. She has a master's degree in instructional technology and a bachelor's degree in computer science (King Saud University in Riyadh). Her areas of expertise include educational technology, e-learning, online learning, robotics, and information and communications technologies in education. She has 12 years of experience teaching computer science in secondary schools in Saudi Arabia and three years at Princess Nourah Bint Abdulrahman University. Her contact information is: School of Education, Princess Nourah Bint Abdulrahman University, 84428 Riyadh, Saudi Arabia, waalalawi@pnu.edu.sa.

Mohd Fadzil Bin Abdul Hanid holds a PhD (Universiti Teknologi Malaysia) and is a lecturer at Universiti Teknologi Malaysia. He teaches in the Department of Advanced Learning Technology (ALT). His expertise is educational technology, and he can be contacted via email at: mohdfadzilabdulhanid@utm.my.