Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

Development of Robot to Improve Learning of Programming Skills among Students

Ahmad Sobri Hashim[1], Aminu Aminu Muazu[1, 2 *], Mohammad Ateff Md Yusof[1], Noreen Izza Arshad[1]

[1]Computer & Information Sciences Department, Faculty of Science and Information Technology Universiti Teknologi PETRONAS, Malaysia.
[2]Computer Sciences Department, Faculty of Natural and Applied Science Umaru Musa Yar'adua University Katsina, Nigeria.

*Corresponding Author: Aminu Aminu Muazu

DOI: https://doi.org/ 10.52866/ijcsm.2023.02.03.001
Received March 2023; Accepted May 2023 ; Available online June 2023

ABSTRACT: The Fourth Industrial Revolution (IR4.0) has shifted the mindsets of engineering students on the importance of IT skills for current and future engineering related jobs. Nowadays, programming is the most fundamental skill that needs to be learnt by the students using mBot technology. The mBot technology is consider as programmable educational robot designed for beginners to learn basic programming concepts which can be assessed and evaluated via bloom's taxonomy framework. It can be a daunting task to learn programming, especially to new students who do not have any prior experience in coding. Average and low performing students are lacking algorithmic skills, where they could not visualize how the programming concepts work. Therefore, this paper presents the effectiveness of using robot to improve students' learning of the programming concepts. In designing the learning modules, bloom's taxonomy model and problem-based learning are adopted using mBot. Moreover, a low-cost and pre-programmed line follower robot has been used to demonstrate the outputs from the programs written in a more interactive manner. For the evaluation, pre-test and post-test of Quasi experimental design have been applied involving 40 students who were identified and categorized as average and low performing groups in the course. The findings show that a significant improvement has been observed from students' performance and motivation. As such, the students' performance is measured based on two phases of experiments. Whereas the students' motivation is measured based on four motivation attributes: self-efficacy, active learning strategy, programming learning value and performance goal.

Keywords: Engineering Education, Technology Education, Fundamentals of Programming, Problem-based Learning, Robotic Intervention, Motivation.

1. INTRODUCTION

PROGRAMMING skills are very important for university graduates, as this skill is highly demanded by the job market nowadays. Most of the companies from various fields in developing countries like Malaysia embark on Information Technology (IT) initiatives to ensure their sustainability and relevancy of business in this digital era [1]. These initiatives were influenced by the current global trend of automation called the Fourth Industrial Revolution (IR4.0). Noticeably that most practical and educational institutions and daily life use the internet in almost every activity [2],[3]. Consequently, the development of any software application like mobile app in android operating system can estimate it time and cost needed for the whole development, even though there are some predicting approaches that can provide efficient effort of software development estimation, for instance the work of [4]. Buzz words such as big data and data analytics, Internet of Things (IoT), cloud computing and cyber physical systems emerged out of this industrial revolution. More and more small and medium enterprises (SMEs) start to automate their business processes [5]. This is a different case with engineering, where this industry has started this digital initiative since last couple of decades. IR4.0 has become a new wave for this industry to implement more sophisticated technologies in producing more and better products and services [6]. Unsurprisingly, engineering graduates with IT skills are more in demand in the industry nowadays [7].

Programming is a compulsory course that needs to be taken by engineering students from most, if not all of the public and private universities in Malaysia. This course introduces fundamental concepts of programming, such as variables, decision, repetition, array, function, structure and many more. The work of [8], [9] made a comparison between skills required for IR4.0 as compared to engineering skills required back in 2015. It was found that

*Corresponding author: aminu.aminu@umyu.edu.ng

programming is one of the seven skills required today which was not really required 5 years ago. The other required skills include analytical thinking and innovation, active learning and learning strategies, creativity, originality and initiative and a few more [10]. This course becomes a feeder for the other industry demand courses, such as data analytics, IoT and robotics, to name a few. Coşkun et al., [11] even added that the programming course that should be taken by engineering students should not only cover introductory concepts but also the advanced concepts applied for data science, artificial intelligence as well as robotics. Having said that, it is crucial for students to master this course. In general, learning programming will nurture and sharpen students' critical thinking in solving computing related problems, train them to focus on every single detail to ensure the written programs successfully working and promote lifelong learning among students.

Learning programming, especially for first timers, is challenging [12]. A preliminary study was conducted to investigate the problems faced by students in understanding programming. This study involved 136 undergraduate engineering students of three private universities in Malaysia. It was found that most of the students lacked algorithmic skills, where they had problems designing a working algorithm (program) to solve given problems. There are many factors contributed to this problem including teaching approach practiced by lecturers was not interactive and interesting enough, students lacked prior knowledge required for learning programming such as Mathematics and lacked motivation when learning difficult lessons. This finding is also supported by studies conducted by [13]–[15]. Most of the universities in Malaysia teach structured programming to undergraduate engineering students, using either C or C++ language. Compilers such as Dev C++ are being used to demonstrate programming concepts, and this method was found to be less interactive, which resulted in the students not being able to visualize how the programming concepts work and thus feeling demotivated [16].

Our goals, then, are to improve students' comprehension of programming concepts by sharpening their algorithmic skills, and to increase their motivation in learning programming. In traditional programming course activities, students are required to design the algorithm and write the program in the compiler. Students will then observe how the program works by looking at the output displayed in the output screen. Some parts of the algorithm could not be observed from the output screen. Thus, learning programming using a low-cost pre-programmed robot, namely mBot, is introduced to fill up the gaps to achieve the goals. The application of mBot is simple, easy to maintain, and is practical as it supports a hands-on approach. Students will be able to comprehend the programming concepts by observing the movement of the robot [17]. mBot is an educational robot designed for students to learn programming and robotics. The robot consists of both electronic and mechanical components that work together to make the mBot function. The electronic components of mBot include an Arduino-compatible mCore controller board that controls all the other components. The ultrasonic sensor detects obstacles, the line follower sensor follows lines, and the light sensor detects light levels. Additionally, the buzzer produces sound, while the RGB LED produces different colors of light. The infrared receiver and transmitter can be used for remote control, and the Bluetooth module enables the mBot to be controlled from a smartphone or tablet. The battery holder powers the mBot. In terms of mechanical components, the mBot includes a chassis, wheels, motors, caster wheel, screwdriver, screws, and nuts. These components allow the mBot to move around and interact with its environment. Students can assemble and disassemble the mBot using these mechanical components, while also learning about the different functions of each component. By using both the electronic and mechanical components of the mBot, students can develop a better understanding of programming and robotics while also having fun building and controlling their own robot. Overall, the combination of electronic and mechanical components in the mBot creates a powerful tool for learning and experimentation in the field of robotics [17].

Students with excellent achievement are assumed to have most of the necessary capabilities and attributes to visualize how the programming concepts work. Therefore, this study will focus on improving only average and low performing students on basic programming concepts. This paper will report the effectiveness of intervention tools applied in improving their understanding of the teaching programming concepts and motivational attributes toward learning programming. mBot has its own programming software called mBlock, where it implements block programming, just like Scratch and MIT App Inventor. A block programming environment is higher in the abstraction levels as compared to high level programming languages, such as C and C++ [18]. Block programming environment is easy to work with and there is less conceptual burden to carry, however, the downside is that it offers lesser programming capabilities. Nevertheless, this tool is sufficient to ensure the goals and objectives are achieved [19]. Students will be assessed in terms of their algorithmic skills and not programming skills, using certain programming language. This intervention aims to benefit the targeted group so that they will not be left behind from appreciating and mastering programming which will be very useful for their future career.

The remaining of the paper is organized in the following sections: related works, methodology, results & analysis, then finally a conclusion.

2. RELATED WORKS

In Malaysia, more and more jobs in job market nowadays require IT skills to fulfill the requirements of digital initiatives influenced by the IR4.0. Tertiary education institutions have also taken quick action to address industry needs, by introducing IT courses to various study programs including engineering, administrative sciences, finance, medical, etc. and one of the courses is programming [20]. Due to its importance, educationists have come up with innovative approaches in delivering programming courses, and one of them is by using current trendy technology, which is robot. There have been quite a number of researchers reporting on the utilization of various interventions and approaches in teaching and learning programming such as gamification [21]. These interventions and approaches have evidenced positive impacts to the students' understanding and motivation. However, this paper will only be focusing on robotic intervention. Robots have been widely applied as assistive tools for teaching and learning programming to various levels of studies in the developed countries for the past 15 years. Among robots that have been used as intervention tools are Lego, NAO, Cozmo, mBot, KitiBot and SparkFun [22].

Ref. [23] investigated the use of mobile robots in teaching programming for IT Engineering students at Kecskemét College, Hungary with the aim to solve problems related to difficulties of abstract thinking in which caused the decrease of programming self-concept and motivation. This investigation involved 149 students using programmable robots, such as Lego NXT, RCX and Surveyor. Quasi Experimental Design and questionnaire instruments were applied to measure programming skills and knowledge, programming self-concept, as well as attitudes toward programming. These three elements were identified as the contributing factors contributing to students' ability to master programming. The result of this study showed that the use of programmable mobile robots improved self-concept and attitudes toward programming. However, it did not improve students' programming skills.

Similar study was conducted by [24] where Lego Mindstorms technology was applied to teach programming to Greek secondary school students. This study involved 100 participants, where the assessment was divided into two parts: robotics (basic mechanical and electronics) and concepts of programming. For the robotic part, participants were given tasks, such as assembling Lego parts and operating the assembled robot using built-in programs, while assessment of programming concepts involved the ability of the participants to design programs that apply certain concepts to fulfill the tasks given. It was found that the participants responded well in robotic assessment parts, where more than 50% of participants gave correct responses on the robotic related tasks. However, less than 50% of total respondents successfully delivered programming related tasks, such as writing input and output commands, and writing programs that involve decision and repetition structures. Finding of this study did not support the general benefits offered by Lego robotics for education as reported by [25] in their systematic literature review study. The analysis showed that most of the studies benefited with the interactive programming platform to program the robot which is using visual programming environment called LabView.

In general, the idea of applying robot to teach programming aims to enhance the interactivity of traditional ways of teaching and learning programming, from using input/output screen to getting inputs and displaying outputs from sensors and robot movements. Therefore, the study conducted by [26] applied virtual robotics programming to develop computational thinking skills involving 364 middle school students from four schools. It was to support the teaching and learning of computational thinking at middle school level of United States. Pre-test and post-test of Quasi Experimental Design were applied to measure the effectiveness of the program. The experiment was designed based on four computational thinking constructs: developing algorithms, evaluating algorithms, developing abstractions, and documentation and process. A small, significant improvement was reported after students underwent the virtual robotics programming program, as compared to the students in non-robotics group.

Another angle of capability of robotic intervention could also be observed from the study that has been conducted by [27] to increase the proportion of women and reduce gender imbalance in Computer Science field using robot. Two robots (NAO and Cozmo) have been used as intervention tool in a workshop involving fourteen female students aged between 15 to 18 years old. Questionnaire was distributed before and after intervention. Result showed the participants have stronger interest in Computer Science field after undergoing the workshop.

Four studies were previously discussed on the adoption of robotics in teaching and learning programming at secondary education level and there were a lot more. As far as tertiary education is concerned, there was a smaller number of studies available. [28] Has also applied an innovative mobile robot to improve learning of programming languages among undergraduate engineering students. A low-cost, pre-programmed robot was applied as an intervention tool for students to improve their academic performance, as well as motivation towards learning programming. Like the study conducted by [24] where the robotic implementation involved two parts: robotics and concepts of programming. However, different findings were reported, where the students gave positive response on robotics part, and they also showed positive improvement in their academic performance. The authors highlighted that students who were involved in the experiment felt more engaged the moment they started assembling the robot. As a result, they could visualize the link between the programs that they designed and the robot that they assembled better. The students' motivation was then measured using ARCS model, and a very high degree of motivation was found in all ARCS measured parameters.

In Munich University of Applied Sciences, robots have also been applied to support collaborative learning among students in practical session [29]. The study used Cozmo robot that could be programmed using professional programming, Python as well as a programming language for beginners which is Scratch. The study also involved 90 participants (both engineering and non-engineering) who were working collaboratively in groups of 3 to 5 people to solve small projects consisting of eight tasks, and the lecturer acted only as a coach. As reported, significant improvement has been achieved for overall course objectives due to the user-centered learning environment promoted by collaborative learning using Cozmo. However, the result from experiment was not really supporting the claim where the instrument used was only questionnaire to measure constructivist process which comprised of six characteristics: active (73.3%), self-regulated (42.2%), emotional (70.4%), situational (37.8%), social (88.1%) and constructive (65.2%). Two characteristics received ratings of less than 50% which are self-regulated and situation. The constructs for both characteristics were the ability to conduct self-regulated learning as well as ability to effectively work and communicate and team. Therefore, achievement of the intervention could be argued. Besides, this study also lacked empirical evidence that can prove the effectiveness of the tool that improves students' learning as compared to existing applied approaches and tools.

Based on the related works reviewed, robots have been widely used to teach programming in secondary schools and tertiary level in the developed countries for the past 15 years. The aim was to introduce the students to the technologies and skills required today. However, different situations happened in the developing countries. There are two studies that have been conducted by two different researchers on the issues and challenges faced by undergraduate students in developing countries in learning programming. [30] Reported the contributing factors include poor pedagogical methods, poor study methods of learners, lack of problem-solving abilities and low self-efficacy. Similar findings were also highlighted by [31] from Nigerian context. Both studies also recommended a few innovations that could be practiced improving teaching and learning programming at undergraduate level. However, most of the recommendations were on teaching and learning approach in general such as to actively involve the students during practical sessions, use real-world examples as case studies, and help in creating student comforts and few others. Assistive tool was not discussed in detail as another innovative solution to overcome the issues faced by undergraduate students in developing countries. In Malaysia, robotic has been introduced to primary and secondary school students for informal education as one of extra-curricular activities. The aim is to nurture interest among students to pursue tertiary education in STEM related courses [32]. Due to its effectiveness, this tool is also seen as an innovative and beneficial approach to assist tertiary education students who are struggling in learning programming to better understand programming concepts and acquire the right programming skills to prepare them for challenging job market.

As far as research direction is concerned, literature survey conducted found very few studies focused on improving teaching and learning programming using robotics, especially in tertiary education. Even though, it has been conducted recently in the work of [33], but basically for children with visual impairment. Among available studies, most of the related works focused very much on the use of robot only, without looking into teaching and learning aspects, such as learning outcomes and pedagogy. On top of that, the selection of participants in the reviewed works was also random, which comprises of excellent, medium and low achievers. It can be argued that excellent students can appreciate robot intervention more, as compared to medium and low achievers. If the analysis is done based on mean score, the score obtained by excellent students will balance up the score obtained by medium and low achievers.

3. METHODOLOGY

The learning module for learning programming concepts using mBot has been developed. It is to guide and ensure the learning process could be run effectively using Bloom's Taxonomy model. Four programming concepts are included in the module: Introduction & Basic Programming Structure, Decision Structure, Repetition Structure and Function. All these modules are designed based on Bloom's Taxonomy model [34], where this model is being applied by most of the higher learning education institutions in Malaysia. Each programming concept will apply a different robot design based on the activities planned. This module applies to the first and second level of Bloom's Taxonomy cognitive domains, which are understanding and application. Problem-based learning was also applied in designing the learning module. At the end of each module, it is expected that students will be able to explain the programming concept taught and apply necessary programming algorithms to solve the given problems using mBot. The map between Bloom's Taxonomy cognitive domains and the intended learning outcomes that wanted to be addressed at the end of the intervention is presented in Table 1.

Table 1. - Map between bloom taxonomy cognitive domains and intended learning outcomes

Cognitive Domain	Intended Learning Outcome
Understanding	To describe programming concepts including basic programming structure, variable, constant, data type, decision, repetition and function.
Applying	To apply suitable programming concepts to solve robotic related problems.

3.1 ASSEMBLING ROBOT

The goal of this project is to improve students' comprehension of programming concepts. It requires an easy-to-use, small scale, pre-programmed robot so that students do not have to spend so much effort on the mechanical and electronic parts of the robot (mBot). The main role of mBot is to present the output of written programs through output devices, such as various robot movement, buzzer, LED, and others, based on applied programming concept. More time and effort should be spent on the learning of programming concepts instead. For this study, 20 sets of mBot were used, with a total cost of USD 2,000. A session was required to assemble the mBots to prepare for the experiment. It took two hours to assemble and to ensure all devices of all the mBot sets were functioning correctly.

3.2 PARTICIPANTS & TIMELINE

This study involved 40 second year undergraduate students from mechanical engineering at a top private university in Malaysia who were taking the Structured Programming & Database System course.

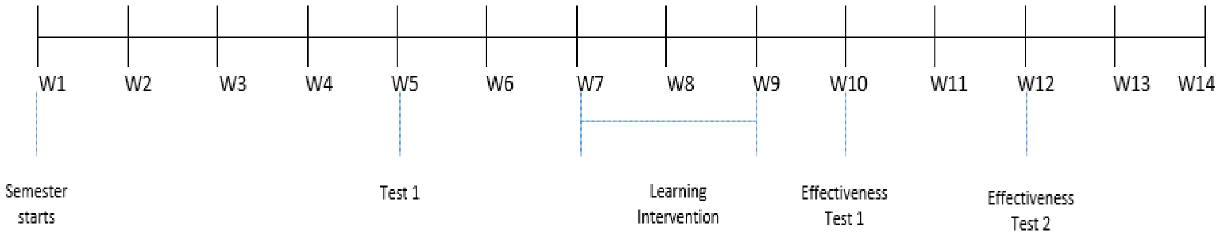

FIGURE 1. Timeline of Experiment.

Figure 1 illustrates the timeline of the execution of this study. The university implements 14 weeks semester system, where the academic activities including lectures, labs and assessments need to be completed in 14 weeks' time. This study aims to improve students' performance. To ensure this study produces reliable and credible findings, the implementation of robotic intervention was embedded in their formal study period. After Test 1 took place in week 5, 40 students who obtained grade B and below were selected for this study. They will undergo three weeks of intervention, where they learnt basic programming concepts using mBot. Finally, two series of effectiveness evaluations were conducted in week 10 and week 12, respectively.

As mentioned, there were four modules of programming concepts designed which are Introduction and Basic Programming, Decision, Repetition and Function. These modules were chosen due to the importance of these modules to be mastered before the students learn more advanced topics such as Array, Vector and so on. In other words, if the students could not master those modules, it will be hard for them to understand the rest of modules in the syllabus. The implementation of the intervention is described in Table 2. The intervention took 3 hours per week. In every module, the instructor will relate prior knowledge that the participants have with the lessons taught during intervention. All modules used for the intervention have already been learnt by participants in week 2 until week 4 during their formal academic sessions. Thus, the implemented intervention functioned as revision for them to prepare for more advanced topics.

Table 2. Implementation of intervention

Week	Module	Implementation
7	Introduction & Basic Programming Structure	Students were introduced to electronic and mechanical components of mBot as well as basic structure of how to program the mBot. The activities included in this module are:
		Observe all built-in capabilities of mBot.
		Design sequential algorithms based on the given problems.
		Translate the algorithms in the form of codes and observe the codes work based on the outputs presented by the robot.
		Applying certain mathematical and logical operation in solving the problems such as arithmetic operation, relational and logical operations, compound operations, and so on.
	Decision Structure	Students were explained on decision structure of mBot programming. Among the activities done in this module are:
		Students to explain how decision structure works, logical and logical operators as well as single and multiple alternatives.
		Design decision algorithms based on the given problems. For example, turn the robot to the right if obstacle is detected.
		Translate the algorithms into working programs.
8	Repetition Structure	Students were explained on repetition structure of mBot programming. Among the activities carried out in this module are:
		Students explain their prior knowledge on repetition structure. Students will also

		need to explain the components repetition structure for mBot. Decision repetition algorithms based on the given problems. For example, turn the robot a certain number of times, change robot moving speeds by certain formula, etc. Translate the algorithm into working programs.
9	Function	Students were explained on divide and conquer concept in problem solving. Students were also explained on the concepts of user defined function for mBot. The activities included in the module are: Students explain the concepts and components of user defined function. Divide the main problem into multiple sub related problems and design algorithms to solve all sub problems. Translate the algorithms into a working program.

3.3 EXPERIMENTAL SETUP & QUESTIONNAIRE

This study implemented experimental research design, which involved two series of effectiveness evaluations, and ended with a questionnaire on motivation.

In the first effectiveness evaluation, two tests were conducted, namely the algorithm test and Test 1 retake. Both tests applied pre and post quasi experimental design, where students were required to answer test questions before and after the intervention. For the algorithm test, students were instructed to design algorithms based on the case study given. Different case studies were given to the students before and after the intervention. Students were then expected to draw a flowchart to represent the algorithm. As mentioned earlier, the participants were selected from the result of Test 1 taken by the students in week 5. After three weeks of intervention, students retook the test with the same questions. From there, the improvement in students' performance could be measured. The students' answers for both algorithm and Test 1 retake were assessed by an independent lecturer.

The second effectiveness test was conducted in week 12. This test was actually the same Test 2 that was taken by all students for that particular semester. The only difference was that the marks for participants of this study experiment were recorded. The marks were then compared with their results of Test 1.

Finally, the debriefing questionnaire was distributed to measure the students' motivation to learn programming after they went through the robotic intervention. The questionnaire was derived and adopted from a study conducted by Tuan, et al. [35], where it measured motivation based on four main attributes: self-efficacy, active learning strategy, programming learning values and performance goal. Table 3 describes each of Tuan et al.'s motivational attribute. Likert scale of 1 (strongly disagree) to 5 (strongly agree) was also applied.

Table 3. Tuan et al.'s motivational model [32]

Attribute	Description
Self-efficacy	Students believe in their own ability to perform well in programming tasks.
Active learning strategy	Students take an active role in learning activities.
Programming learning values	Students acquire important programming learning values such as the concepts of programming.
Performance goal	Students understand their goal in learning programming.

The questionnaire has gone through content validity checking by a panel of experts. All suggested modifications and improvements were taken into consideration to ensure the clarity of the sentences and the appropriateness based on the context. Besides, pilot test was also conducted involving 12 random undergraduate students to evaluate the internal consistency of the questions included in the questionnaire. Result showed the questionnaire passed the reliability threshold (Cronbach's alpha = 0.833). Therefore, the questionnaire was considered suitable to be applied in this study [36].

3.4 DATA ANALYSIS

This study applied quantitative research methods; thus, statistical analyses were performed to analyze the data. Descriptive analysis and t-test were performed for both effectiveness tests as shown in Tables 5 and 7. Meanwhile, the data of questionnaire to measure motivation was analyzed using only descriptive analysis.

4. RESULTS & ANALYSIS

The ultimate goal is to improve students' understanding of programming, and this can be easily reflected from their tests or examination results. This study involved two series of experiments to measure improvement in the participants' understanding on fundamental programming concepts, and a survey to measure their motivation to learn programming after undergoing the three weeks of intervention using mBot.

4.1 EFFECTIVENESS TEST 1

There were two tests conducted for Effectiveness Test 1: Algorithm Test and Test 1 Retake. The algorithm test applied quasi experimental design. During pre-test, which was conducted in week 6, participants were given a problem and they were required to design an algorithm in the form of flowchart to solve the problem in 20 minutes. The participants' answers were then marked by an independent lecturer, and the marks were recorded in SPSS software. The algorithm was assessed based on the right flowchart symbols used and the right steps in solving the problem. After they underwent three weeks of intervention using mBot, a post-test was conducted. For the post-test, different questions with similar levels of difficulty as pre-test were given to the participants. 20 minutes were given to them, and the same independent lecturer was assigned to mark the answers. Table 4 presents the marks obtained by the participants for this algorithm test.

4.1.1 Algorithm Test

Based on the conducted pre-test for algorithm test, 77.5% (31) of the participants scored less than 4 marks out of the total of 10 marks, 15.0% (6) obtained 4 to 7 marks and 7.5% (3) obtained more than 7 marks. A significant improvement has been recorded, where the number participants who obtained below than 4 marks reduced to 17.5% (7), 4 to 7 marks increased to 62.5% (25) and 20.0% (8) obtained more than 7 marks.

Table 4. Students' marks for algorithm test

Phase	Mark		
	<4	4 to 7	>7
Pre-test	77.5%	15.0%	7.5%
Post-test	17.5%	62.5%	20.0%

This data has been further analyzed using a t-test. Before conducting the t-test, the assumption of normality of data for both pre-test and post-test marks, and homogeneity of variances have been verified using Shapiro-Wilk normality test and Levene's test respectively. Table 5 presents the results of both tests as well as the t-test. The results for both tests verified the assumption of the normality and homogeneity of data.

Sig. value (0.00) of t-test rejected the null hypothesis and accepted the alternative hypothesis. The mean mark obtained by students for post-test was 5.85, while mean mark for pre-test was 3.25. It was found that there is significant difference between students' marks of algorithm test before and after undergoing the mBot intervention, where students performed better in post-test as compared to pre-test.

Table 5. Results of Shapiro Wilk Normality Test and Levene's Homogeneity Test for algorithm test

Shapiro-Wilk Normality Test			
Phase	***Statistics***	***Degree of freedom***	***Sig.***
Pre-test	0.926	40	0.00
Post-test	0.911	40	0.00

Levene's Homogeneity Test		
	F	***Sig***
Equal variances assumed (A)	2.87	0.01
Equal variances not assumed (B)		

t-test for Equality of Means							
	T	df	(1)	(2)	(3)	(4)	
						Lower	***Upper***
(A)	7.01	61.0	0.00	1.47	1.16	5.32	10.61
(B)	6.89	65.12	0.00	1.47	1.11	5.73	10.47

(1) Sig. (2-tailed)
(2) Mean difference

(3) Standard error difference
(4) 95% confidence interval of the difference

4.1.2 Retake Test 1

Students took Test 1 in week 5 of the semester. According to Table 6, the test recorded 40.0% (16) of the participants scored less than 40 marks, 60.0% (24) scored between 40 and 65 marks. After three weeks of intervention, participants were asked to retake the test. A significant improvement could be observed from participants' performance, where the number of students who obtained less than 50 decreased to 3 (7.5%), 18 (45.0%) participants obtained 40 to 65 marks and 47.5% (19) obtained more than grade B (65 marks).

Table 6. Students' marks for test 1 retake

Phase	Mark		
	<40%	40% to 65%	>65%
Test 1	40.0%	60.0%	0.0%
Retake	7.5%	45.0%	47.5%

The same analysis step as the algorithm test has been performed, where Shapiro-Wilk and Levene's test were executed to verify normality and homogeneity of data. T-test (shown in Table 7) was then executed to indicate the difference between participants' performance of Test 1 and Retake, as an analysis of means equality. In Table 7, the Sig. value presented indicates the significant difference between Test 1 and Test 1 Retake which was taken by participants after the three weeks' intervention using mBot. The average mark obtained by participants for Test 1 was 52.0%. A significant improvement was recorded from Test 1 retake, where the mean mark obtained by participants was 72.5%.

Table 7. t-test for Test 1 retake

	t-test for Equality of Means						
	T	df	(1)	(2)	(3)	(4)	
						Lower	Upper
(A)	12.6	47.4	0.00	13.7	2.66	10.28	19.19
(B)	14.7	50.2	0.00	13.7	2.94	10.91	18.57

(1) Sig. (2-tailed)
(2) Mean difference
(3) Standard error difference
(4) 95% confidence interval of the difference
(A) Equal variances assumed
(B) Equal variances not assumed

4.2 EFFECTIVENESS TEST 2

Participants' performance once again was measured based on marks that they obtained from Test 2, which was held in Week 12. The marks that they obtained in Test 2 were then compared with their achievement in Test 1. Table 8 summarizes the finding. The percentage of participants who obtained less than 40% and between 40% and 65% have reduced from 40% (16) to 12.5% (5), and from 60% (36) to 30% (18) respectively. The percentage of participants scored more than 65% also increased from 7.5% (3) to 57.5% (23).

Table 8. Students' marks for effectiveness test 2

Phase	Mark		
	<40%	40% to 65%	>65%
Test 1	40.0%	60.0%	7.5%
Test 2	12.5%	30.0%	57.5%

A T-test was also performed, where the Sig. value (0.00) was less than 0.05, which concludes significant difference between marks obtained by participants in Test 1 and Test 2. This analysis was further elaborated with mean difference, where the mean mark for Test 2 was higher than Test 1, at 73% and 52% respectively.

Analyses of both effectiveness tests reflected improvement in participants' performance after undergoing the mBot intervention.

4.3 MOTIVATION

Preliminary study was conducted and one of the factors that contributed to low performance among students was their own attitude when they felt it was difficult to grasp the programming concepts, which made them feel unmotivated. A questionnaire instrument was applied to measure participants' motivation after they went through intervention using mBot. Four motivational attributes were measured: self-efficacy, active learning strategy, programming learning value and performance goal. Table 9 presents the mean rate rated by participants. All measured motivational attributes were rated more than average scale (3.00). It can be concluded that most of the respondents agreed that at the end of the intervention, they believe in their own ability to perform well in programming tasks, they have grasped basic programming concepts, they understand their goal in learning programming and they enjoy the activities that they did in learning programming using mBot.

Table 9. Mean Rate for measure of motivational attributes.

Value	Mean	SD
Self-efficacy	4.03	0.73
Active learning strategy	3.97	0.71
Programming learning value	4.14	0.76
Performance goal	4.07	0.71

5. CONCLUSION

This study aimed to measure the effectiveness of robot intervention in improving the learning of fundamentals of programming, and to enhance students' motivation towards learning programming. The effectiveness of the intervention has been measured through two series of tests, which are Effectiveness Test 1 and Effectiveness Test 2. The results from Effectiveness Test 1 depicted a significant improvement on Algorithm Test and Retake Test 1 after undergoing the intervention. Meanwhile, a significant improvement was also recorded in students' marks in Test 2, as compared to Test 1. All assessments were conducted in accordance with the formal semester timeline. In terms of motivation, positive feedback was given by the students on all four measurements, namely self-efficacy, active learning strategy, programming learning value and performance goal. Due to the scope of this study being only limited to mechanical engineering students from one university, the finding of this study therefore could not be generalized. However, it is sufficient to prove the effectiveness of this intervention to improve learning of programming among average and low performing students. As part of our future work, we are currently investigating a new innovation to generalize our finding to the students at Malaysian universities.

Funding

None

ACKNOWLEDGEMENT

On behalf of our research team, I would like to express my special thanks of gratitude to MURATA Science Foundation, Ministry of Higher Education Malaysia and UTP Foundation for funding this research in order to innovate new approaches of teaching and learning engineering and technology related courses.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] L. S. Istiyowati, Z. Syahrial, and S. Muslim, "Readiness for the Implementation of Ubiquitous Learning in Programming Course in Higher Education," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 18, 2021, doi: 10.3991/ijim.v15i18.24547.

[2] Mohammed Thakir Shaamood, "Internet applications in libraries and information centers," *Iraqi Journal for Computer Science and Mathematics*, 2020, doi: 10.52866/ijcsm.2020.01.02.003.

[3] A. Aminu Muazu and I. Umar Audi, "Network Configuration by Utilizing Cisco Technologies with Proper Segmentation of Broadcast Domain in FNAS-UMYUK Nigeria," *Journal of Network Security and Data Mining*, vol. 4, no. 1, pp. 1–13, 2021, doi: 10.5281/zenodo.4776375.

[4] Zainab Rustum Mohsin, "Investigating the Use of an Adaptive Neuro-Fuzzy Inference System in Software Development Effort Estimation," *Iraqi Journal for Computer Science and Mathematics*, 2021, doi: 10.52866/ijcsm.2021.02.02.003.

[5] M. A. Sait and M. Anshari, "Industrial Revolution 4.0: A New Challenge to Brunei Darussalam's Unemployment Issue," *International Journal of Asian Business and Information Management*, vol. 12, no. 4, 2021, doi: 10.4018/IJABIM.20211001.oa3.

[6] Naptsoksch, B. (2022). Smart robot using in smart homes. Wasit Journal of Computer and Mathematics Science, 1(4), 87–93. https://doi.org/10.31185/wjcm.84

[7] A. Aminu Muazu, B. Aminu, and A. Kamaluddin, "DESIGN AND IMPLEMENTATION OF ANDROID QUIZ APP-IN HIGHER INSTITUTION OF LEARNING: UMARU MUSA YAR'ADUA UNIVERSITY KATSINA, NIGERIA," in *AICTTRA 2018 Proceeding* , 2018, pp. 67–71.

[8] F. Mohd Kamaruzaman, R. Hamid, A. A. Mutalib, and M. S. Rasul, "Comparison of engineering skills with IR 4.0 skills," *International journal of online and biomedical engineering*, vol. 15, no. 10, pp. 15–28, 2019, doi: 10.3991/ijoe.v15i10.10879.

[9] A. Akkaya and Y. Akpinar, "Experiential serious-game design for development of knowledge of object-oriented programming and computational thinking skills," *Computer Science Education*, 2022, doi: 10.1080/08993408.2022.2044673.

[10] Qasim, A. M., Aljanabi, M., Kasim, S., Ismail, M. A., & Gusman, T. (2022). Study the field of view influence on the monchromatic and polychromatic image quality of a human eye. JOIV: International Journal on Informatics Visualization, 6(1-2), 151-154.

[11] S. Coşkun, Y. Kayıkcı, and E. Gençay, "Adapting Engineering Education to Industry 4.0 Vision," *Technologies (Basel)*, vol. 7, no. 1, p. 10, Jan. 2019, doi: 10.3390/technologies7010010.

[12] A. Anggrawan, C. Satria, Mayadi, and N. G. A. Dasriani, "Reciprocity Effect between Cognitive Style and Mixed Learning Method on Computer Programming Skill," *Journal of Computer Science*, vol. 17, no. 9, 2021, doi: 10.3844/jcssp.2021.814.824.

[13] F. J. Gallego-Durán, R. Satorre-Cuerda, P. Compañ-Rosique, C. J. Villagrá-Arnedo, R. Molina-Carmona, and F. Llorens-Largo, "A low-level approach to improve programming learning," *Univers Access Inf Soc*, vol. 20, no. 3, 2021, doi: 10.1007/s10209-020-00775-y.

[14] M. Piteira and C. Costa, "Learning computer programming: Study of difficulties in learning programming," in *ACM International Conference Proceeding Series*, 2013. doi: 10.1145/2503859.2503871.

[15] V. F. Martins, I. de Almeida Souza Concilio, and M. de Paiva Guimarães, "Problem based learning associated to the development of games for programming teaching," *Computer Applications in Engineering Education*, vol. 26, no. 5, 2018, doi: 10.1002/cae.21968.

[16] D. T. Butterworth, "Teaching C / C ++ Programming with Lego Mindstorms," *3rd International Conference on Robotics in Education*, 2012.

[17] C. Vandevelde, J. Saldien, C. Ciocci, and B. Vanderborght, "Overview of technologies for building robots in the classroom," in *Proceedings of the 4th International Conference on Robotics in Education*, 2013.

[18] E. Adetiba *et al.*, "Evolution of Artificial Intelligence Programming Languages - a Systematic Literature Review," *Journal of Computer Science*, vol. 17, no. 11, 2021, doi: 10.3844/JCSSP.2021.1157.1171.

[19] S. N. H. Mohamad, A. Patel, R. Latih, Q. Qassim, L. Na, and Y. Tew, "Block-based programming approach: Challenges and benefits," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, ICEEI 2011*, 2011. doi: 10.1109/ICEEI.2011.6021507.

[20] R. C. Ferguson, P. D. Leidig, and J. H. Reynolds, "Including a programming course in general education: Are we doing enough?," in *2014 Proceedings of the Information Systems Educators Conference, ISECON 2014*, 2014.

[21] A. Rojas-López, E. G. Rincón-Flores, J. Mena, F. J. García-Peñalvo, and M. S. Ramírez-Montoya, "Engagement in the course of programming in higher education through the use of gamification," *Univers Access Inf Soc*, vol. 18, no. 3, 2019, doi: 10.1007/s10209-019-00680-z.

[22] P. Vostinar and N. Klimova, "Experience with Using Robots for Teaching Programming," in *Proceedings of the International Conference on Information and Digital Technologies 2019, IDT 2019*, 2019. doi: 10.1109/DT.2019.8813310.

[23] A. Pásztor, R. Pap-Szigeti, and E. Török, "Mobile Robots in Teaching Programming for IT Engineers and its Effects," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 11, 2013, doi: 10.14569/ijacsa.2013.041123.

[24] M. Sartatzemi, V. Dagdilelis, and K. Kagani, "Teaching programming with robots: A case study on Greek secondary education," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2005, vol. 3746 LNCS. doi: 10.1007/11573036_47.

[25] I. M. L. Souza, W. L. Andrade, L. M. R. Sampaio, and A. L. S. O. Araujo, "A systematic review on the use of LEGO® robotics in education," in *Proceedings - Frontiers in Education Conference, FIE*, 2019, vol. 2018-October. doi: 10.1109/FIE.2018.8658751.

[26] E. B. Witherspoon, R. M. Higashi, C. D. Schunn, E. C. Baehr, and R. Shoop, "Developing computational thinking through a virtual robotics programming curriculum," *ACM Transactions on Computing Education*, vol. 18, no. 1, 2017, doi: 10.1145/3104982.

[27] L. Keller and I. John, "How can computer science faculties increase the proportion of women in computer science by using robots?," in *IEEE Global Engineering Education Conference, EDUCON*, 2019, vol. April-2019. doi: 10.1109/EDUCON.2019.8725212.

[28] O. O. Ortiz, J. A. P. Franco, P. M. A. Garau, and R. H. Martin, "Innovative mobile robot method: Improving the learning of programming languages in engineering degrees," *IEEE Transactions on Education*, vol. 60, no. 2, 2017, doi: 10.1109/TE.2016.2608779.

[29] L. Brehm, H. Guenzel, O. Hinz, A. Humpe, and H. Martius, "Collaborative learning with COZMO to teach programming in scratch and python," in *IEEE Global Engineering Education Conference, EDUCON*, 2019, vol. April-2019. doi: 10.1109/EDUCON.2019.8725037.

[30] J. O. Oroma, H. Wanga, and F. Ngumbuke, "CHALLENGES OF TEACHING AND LEARNING COMPUTER PROGRAMMING IN DEVELOPING COUNTRIES: LESSONS FROM TUMAINI UNIVERSITY," in *INTED2012: INTERNATIONAL TECHNOLOGY, EDUCATION AND DEVELOPMENT CONFERENCE*, 2012.

[31] S. Dasuki and A. Quaye, "Undergraduate students' failure in programming courses in institutions of higher education in developing countries: A Nigerian perspective," *Electronic Journal of Information Systems in Developing Countries*, vol. 76, no. 1, 2016, doi: 10.1002/j.1681-4835.2016.tb00559.x.

[32] P. Y. Jiea, T. C. Chuan, and ..., "Promoting students' interest in STEM education through robotics competition-based learning: case exemplars and the way forward," *Learning Science and …*, vol. 0832, no. 14, 2019.

[33] S. Psycharis, P. Theodorou, Y. Spanidis, and P. Kydonakis, "Teaching Programming Skills to Blind and Visually Impaired Learners," in *Lecture Notes in Networks and Systems*, 2022, vol. 349 LNNS. doi: 10.1007/978-3-030-90677-1_26.

[34] M. Garg, D. Dhariwal, and C. Newlands, "Providing national level teaching to OMFS specialty trainees in a virtual classroom setting using learning theories of education," *British Journal of Oral and Maxillofacial Surgery*, vol. 60, no. 1. 2022. doi: 10.1016/j.bjoms.2021.02.017.

[35] H. L. Tuan, C. C. Chin, and S. H. Shieh, "The development of a questionnaire to measure students' motivation towards science learning," *Int J Sci Educ*, vol. 27, no. 6, pp. 639–654, May 2005, doi: 10.1080/0950069042000323737.

[36] Uma. Sekaran and Roger. Bougie, *Research methods for business: a skill-building approach*. Wiley, 2013.