Asia Pacific Journal of Contemporary Education and Communication Technology (APJCECT)
ISBN: 978 0 9943656 82; ISSN: 2205-6181
Year: 2017, Volume: 3, Issue: 1
www.apiar.org.au

THE DEVELOPMENT OF TEACHING KIT FOR INTRODUCTORY PROGRAMMING IN SCHOOLS

Azizah Suliman[a], Surizal Nazeri[b]
[ab] Universiti Tenaga Nasional, Putrajaya Campus, Malaysia
Corresponding email: azizah@uniten.edu.my

Abstract

As the problem of teaching and learning programming continues in tertiary education, tools and approaches are often proposed and developed. This paper describes the development of a teaching and learning package for introductory programming in schools. The teaching and learning package consist of a teaching module and an embedded board training kit. Lessons plan is also described so the effectiveness of the package as designed can be maximized. The lessons are supposed to invoke students' interest in programming, cooperativeness in finding solutions and hence enhance their learning experience. The teaching module and training package were tested and results were briefly described.

Keywords: Teaching and Learning Programming, Programming in Schools, Embedded Systems

1. Introduction

The problem of teaching and learning of introductory programming are common and it is a global issue. Example of early studies conducted by Evans and Simkin (1989) and the more recent one by Seyal, Siau and Mey (2015), to name the very few, prove that the issue is still relevant and ongoing. Our paper (Suliman & Nazeri, 2016) has discussed the factors that could contribute to the issue at length. From teachers' teaching style, students' learning methods, to the level of difficulties of the subject matter; all are major contributing factors to the problem. Traditional teaching strategies where teachers explain the elements of the programming concepts, show some examples and then leave it to students to solve problems seems to have limited impact to the effectiveness of the lesson. Students study methods of memorizing by reading and looking at solved exercises too are not the ideal way to learn programming (Gomes & Mendes, 2014). With the abstract nature of programming, highly needed skills of problem solving and error prone coding, learning how to program are definitely not an easy feat for beginners.

Several teaching tools and pedagogical approaches have been introduced by researchers over the years. However, the issues of high failure rates in programming subjects in higher learning institutions continued. It is believed no one method or tool for teaching and learning programming would be applicable to all level of students and teachers. How the issue can be tackled also need consideration of the class environment, education system, culture and available resources. This research work is developed based on the Malaysian school curriculum and class environment. The proposed work used existing resources and class style of Malaysian school so it may be easily adopted and used.

The rest of this paper is organized as follows. The next section, which we will refer to as section two and subsequently numbered, reviews related work in teaching and learning

programming and the motivation behind this research work. In section three, we describe the teaching and learning package and its lesson plan. Section four briefly describes the evaluation of the teaching and learning kit. Finally, a brief conclusion is given in section five.

2. Literature Review

Some of the efforts done by the Malaysian government to increase the number of programmers and computer application developers in Malaysia were through the introduction of computer subjects in school. In 2002, Computer Programming courses were introduced at vocational schools in the country (Shirat, 2003). The introduction of this course will provide early exposure to students who are interested to continue their lesson in the field of information technology or computer science. In 2006, the Ministry of Education has introduced the subject of Information Communication and Technology (ICT) as an elective subject in school to learn the basics of computers at the school level (Suliman & Nazeri, 2014). However, till now, the number of students taking up Computer Sciences in tertiary institutions is still on a decline. Interest and competency in programming are relatively still very low.

Previous Literatures have shown that the problem is global and ongoing. Gomes and Mendes (2014) discussed the difficulties faced in teaching and learning programming from teachers' perspective in Portugal. Difficulties may result from students' competence deficiencies, especially in logical thinking and mathematical background and study skills, class size and organization to traditional teaching strategies and the choice of programming language used. De Lira Tavares, De Menezes and De Nevado (2012) presented a pedagogical architectures designed especially to support the process of teaching and learning of programming. The architecture is defined at two levels, the educational strategy and technological resources. Many have developed teaching tools to support the teaching and learning programming, examples of recent literature are (Enzai et al., 2011; Husain, Tarannum & Patil, 2013; Tillmann et al., 2013; Chandramouli, Zahraee & Winer, 2014; Huang et al., 2014; Begosso et al., 2015; Janke, Brune & Wagner, 2015; Malliarakis, Satratzemi&Xinogalos,2016). All the tools created have somewhat helped in achieving the lessons' learning outcomes.

In retrospect of the many literatures available, we designed our teaching and learning package based on the motivation to create interest in the students to learn. Crucial influencing factors such as interest and perception of students toward programming, can be given great deal of focus so the problems of high failure rate may be reduced. If there is no interest, they'll find it hard to understand even a simple concept. At the same time, problem solving skill is another major factor that should be emphasized on as it leads to a better understanding of programming topics later. As such, the teaching module with an accompanying embedded kit was created. The accompanying embedded kit would be a pleasant change from the usual computer monitor that they used to execute their program. Output and input devices such as small LCD monitor, mini keypad, push buttons, LED lights, and buzzers are among the devices used for students to work on. The use of embedded kit in this work is not coincidental, but intentional. The objective was more than to just create interest in programming, but also in embedded system programming. Though, in this teaching kit, we have created a lot of library functions so the students need not bother about configuring the input and output ports of the various different devices, it was our intention for them to be aware of the embedded system. With the buzz of IOT, embedded system programmers are also as highly in need.

3. Teaching Module

The teaching and learning package developed for this research consist of a teaching module and a training kit developed on an embedded board. The programming language used is C. The teaching module is an accompanying book containing lessons on programming fundamentals. The module comprehensively contains introductory topics on C programming, related experiments, examples and exercises to test the students' understanding of the topics covered. The topics that the module carries are aligned with the syllabus and standard specifications

that have been set by the Ministry of Education of Malaysia, for programming subject in school. The already prescribed topics are maintained, the only changes made are the insertion of the introduction to embedded systems. Since the students will be using the training kit on an embedded system board, it is only fitting for them to be introduced to the elements of embedded programming, especially its input and output ports and devices.

The additional topics are necessary as the embedded systems devices such as LED lights, keypad, LCD and others will act as input and output interfaces for the experiments and examples, in replacement of the input and output devices of a PC as commonly used. The module contains about 7 topics as shown in Table 1.

Table 1: Topics Covered in the Teaching Module

Chapter	Topic	Contents
1	Introduction to Programming	• Why Program? • Types of programming and programming languages • Programming process: compilation, execution
2	Basic Problem Solving	• Algorithm • Flowchart • Pseudo-code
3	Fundamentals of C programming	• Program Structure • Variable, Constant and Data Types • Operators and Expression • Input/output
4	Selection Statements	• if, if-else • Nested if-else • switch
5	Iterative Statements	• for • while • do-while
6	Functions	• What is a function? • Function construct: Function prototype, function definition, function call, parameter list
7	Introduction to Embedded Programming	• Introduction to Embedded Systems • Input Output Ports & Devices

At the start of every chapter, the objectives and the learning outcomes of the chapter are specifically stated. The objectives and learning outcomes are written so it covers the three domains as identified by educational psychologist, Benjamin Bloom. The domains are cognitive, psychomotor and affective. The cognitive domain encompasses Fes intellectual or thinking skills, psychomotor encompasses physical skills or performance of actions and affective encompasses attitudes and values. In teaching and learning programming, the first two domains would be most applicable and must be incorporated. Teachers are encouraged to go through the learning outcomes when starting a new topic. Students are encouraged to check and gauge at the end of the lesson if the learning outcomes have been met.

Each topic is accompanied with activities and exercises to test students' understanding. Each subtopic will contain at least three questions that have different answering and solutions technique. Sample of activity and exercises of a topic in Chapter 4 are as shown in Figure 1. Diversity of the questions are meant to test the extent of students' abilities to identify

problems and answer questions based on what they have learned before. It is also to satisfy the different levels of Bloom Taxonomy. The examples in Figure 1 covers from remembering and understanding right to applying. By having exercises at every subtopic, this will also help to re-emphasize all that has been taught and increase students' level of understanding and participation in learning. All of the lab exercises will be executed on the embedded kit.

Theory Exercise

1. When do you use selection statements?
2. What is a condition?
3. Why should you indent the statements within the **if** statement?

Activity 4.1

Step 1: Write this code

```
1.  void main ( )
2.  {
3.   int a = 15;
4.   if((a>=10) (a>=20));
5.        printf("\n Value is between 10 and 20 ");
6.  }
```

Step 2: Add the appropriate relational operator (&& or ||) in code line 4, so that the printf( ) statement in code line 5 will be executed.

Lab Exercise

Lab Exercise 1: Write a program to light up blue LED if Button 4 is pressed.

Lab Exercise 2: Write a program to display your name when Button 2 is pressed.

Figure 1: Sample of Activity and Exercises

4. The Teaching Plan

A proposed teaching plan accompanied the teaching module. This is to recommend to teachers who are using the module the best way to use the teaching and learning package. With a correct teaching style and usage of the module, the intended outcome can be maximized. The teaching plan is organized in the concept of TALK and DO. TALK would be where teachers are expected to explain and teaches the students of the corresponding module's content. DO would be when the students are involved in doing programming oriented tasks that are incorporated in every topic and section of the module. Table 2 shows an example of the teaching plan for selected chapters.

Table 2: Teaching Plan

OPERATIONS OR CONTENT	TRAINING METHODS	TRAINING RESOURCES
Introduction of the Course	TALK	Teaching Plan & Module
Chapter 1: Introduction	TALK	Module and PowerPoint Slides
Quiz and Exercise	DO (Write answer in answer booklet)	Quiz/Exercise Question
Chapter 2: Problem Analysis & Design	TALK	Module and PowerPoint Slides
Quiz and Exercise	DO (Write answer in answer booklet)	Quiz/Exercise Question
Chapter 3: Fundamentals of Embedded Programming	TALK & Demonstration DO: Doing Experiment	Module, PowerPoint Slides and Embedded Kit
Quiz and Exercise	DO (Write answer in answer booklet)	Quiz/Exercise Question
Chapter 4: Selection Statements	TALK & Demonstration DO: Doing Experiment	Module, PowerPoint Slides and Embedded Kit
Quiz and Exercise	DO (Write answer in answer booklet)	Quiz/Exercise Question
Chapter 5: Looping Statements	TALK & Demonstration DO: Doing Experiment	Module, PowerPoint Slides and Embedded Kit
Quiz and Exercise	DO (Write answer in answer booklet)	Quiz/Exercise Question
Chapter 7: Embedded Programming	TALK & Demonstration DO: Doing Experiment	Module, PowerPoint Slides and Embedded Kit
Quiz and Exercise	DO (Write answer in answer booklet)	Quiz/Exercise Question
Project Preparation	DO: Developing a program	Embedded Kit and Module
Project Presentation	DO: Group Presentation	Embedded Kit
TOTAL		

A manual is also prepared to explain to the students the proper way to use the embedded kit. This manual contains a diagram of the installation of power cable and connection between the embedded kit and computer. In addition, the manual guides the user how to connect the jumper that connects the device to the microcontroller. This manual also explains how to use the MikroC PRO for PIC and PICkit2 software. Users need to read this manual before they can start using the training kit. It will help the user to understand the proper use of the training kit to minimize damage.

5. Evaluation of Teaching Module

The teaching and learning package were tested. The detail of the evaluation process was discuss at length in our earlier paper (Suliman & Nazeri, 2014). To briefly describe here, we ran four teaching workshops on four groups of students selected from two types of schools, which are the day schools and vocational schools. Photos of some the sessions in progress are shown in Figure 2. Students from vocational schools who have had basic programming for 8 months where they have been exposed on how to develop programs. Students from the daily schools have yet to be exposed to any formal basic programming in schools, which

means they do not have any programming background. The effectiveness of the teaching and learning package are measured by comparing the performance of the two groups. Will the group without programming background performed at par with the group with programming background?

Figure 2: Teaching and Training Session

Total teaching time depends on the approval given to the school. Lesson plans are broken down into two categories, which are 2 days of teaching and 3 days of teaching. For schools that provides 2 days of teaching and learning, teaching begins at 8 am and end at 5 pm and the total amount of time for teaching session is 15 hours and for the school that provides 3 days, teaching session begins at 8 am and ends at 2 pm. Total number of hours for 3 days is 16 hours.

During the teaching process, students were divided into groups of 2 to 4. Each group is provided with 1 teaching module, 1 embedded kit and 1 set of answer booklet for them to write answers to each exercise session. Students' interest and understanding were assessed through:

i) Survey: The survey collects information on interests, understanding and improvement to the teaching module that can be done. The survey is carried out after the completion of the teaching and learning process. Each student will be given a set of survey questions that they need to answer and return to the instructor.

ii) Exercises: The exercises as found in every sub topic as shown in Figure 1, need to be answered by students in a group. Exercises consist of several questions and students needs to answer the question in the answer booklet provided and each group is required to answer all the questions. At the end of the course, they need to return the answer booklet to the instructor for grading.

iii) Project: Project is to identify whether the students can apply what they have been learned to produce a program using the Embedded Kit. Each group was given time to write a program using the embedded kit that must have input and produce output. Marks are dependent on the amount subtopic that can be included in the program. The more subtopics are used in a program, the more marks they will obtained. Students were made known of the marking criteria. Each group is required to present their project.

6. The Results

The results of the two groups were compared. Their performance in the exercises and projects were graded and awarded grades based on the usual scale used in schools. In general, both groups did equally good. A summary of their performance are given in Figure 3. There are no huge gaps in performance were observed, though the group who had done programming before performed slightly better than the group who had no programming experience. Even though students in Group 1 have the lowest grade of C while in group 2 the lowest was D, 25% of the students from both grade scored the highest grade of A and A+. In their project works, it was noted that the students from Group 2 showed more creativity and complexity in applying programming techniques that they have learned where they incorporated more I/O devices on the embedded kit into their projects and use more program constructs.

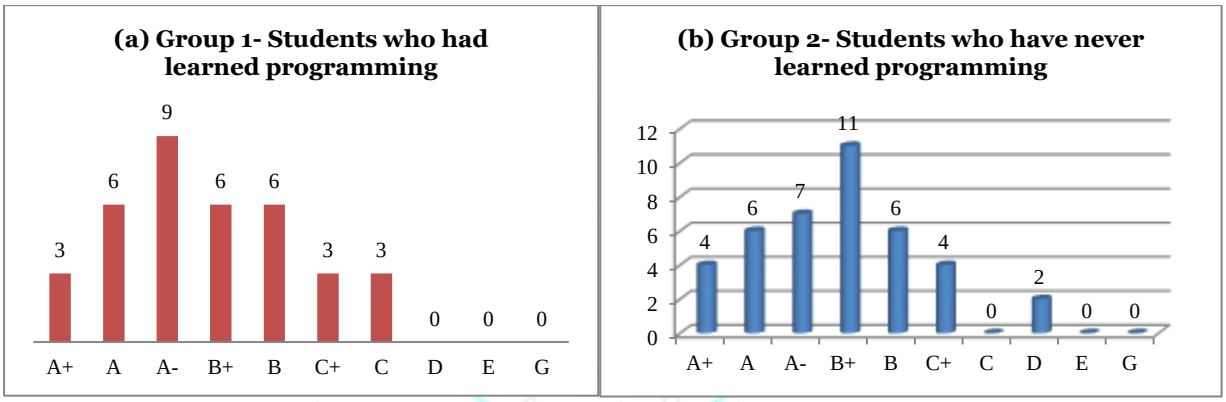

Figure 3: Grades comparisons of the two groups

Conclusions

In most teaching assignments, the success of the teaching methods, modules and instructors are commonly measured through the achievements of students in their exams and tests. Achievements are often measured through the grades achieved and will reflect if the teaching sessions have been effective in achieving its outcomes. The evaluation of the embedded teaching tool used shows a remarkably positive result in terms of the students' assessments. Throughout the teaching session students were notably more interactive with the instructor and their peers. The use of the embedded board for the programming practical sessions managed to invoke their interest and inquisitive nature. Since the output of the embedded kit is very visible, there was also an obvious element of competitiveness between them to produce better output. The class atmosphere was more lively than usual. All these factors helped to motivate them to learn and understand the lessons better.

In general, we were quite pleased with the results achieved. It is in our intention to proceed the testing phase further to increase the number of participating school children. However, the constraints are huge. While conducting the teaching workshops are energy and time consuming efforts, that in itself is not the major obstacle. The most challenging is in getting the permission to have the teaching sessions in school. With the tight teaching schedule already in place at

most schools, to get the permission to interrupt their daily schedule was not easy. Teachers have to re-plan their teaching schedule and assignments. However, we will carry on with our cooperation with the schools and plan to have more workshop session to further test the effectiveness of the teaching kit. We are confident the teaching kit is helpful in creating interest in the school children to like programming and as a consequence do better in their programming courses later in their tertiary education. And we believed that two pronged methods also work in creating the awareness towards embedded system programming.

Acknowledgements

This earlier and vital part of this research project was funded by Malaysian's Ministry of Science, Technology and Innovation (MOSTI) through the E-science fund (06-02-03-SF0185).

References

i. Begosso, L.C., Begosso L.R., Ribeiro, A., dosSantos, R.M.&Begosso, R.H., 2015. The use of Learning Objects for Teaching Computer Programming. *Frontiers in Education Conference (FIE)*, pp. 1-6.

ii. Chandramouli, M., Zahraee, M. &Winer, C., 2014. A Fun-Learning Approach to Programming: An Adaptive Virtual Reality (VR) Platform to Teach Programming to Engineering Students. *IEEE International Conference on Electro/Information Technology*, pp. 581 – 586.

iii. de Lira Tavares, O., de Menezes, C. S. &deNevado, R. A., 2012. *Pedagogical Architectures to Support the Process of Teaching and Learning of Computer Programming*. 2012 Frontiers in Education Conference Proceedings.

iv. Enzai, N. I., Che Kar, S. A., Husni, H. &Ahmed, N., 2011. Harnessing Creativity in Teaching and Learning Introductory Programming. *Research and Development (SCOReD)*, pp. 431 – 435.

v. Evans, G. E. & Simkin, M. G., 1989. What Best Predicts Computer Proficiency? *Communications of the ACM*, 32(11), pp. 1322-1327.

vi. Gomes, A. &Mendes, A., 2014. *A Teacher's View about Introductory Programming Teaching and Learning: Difficulties, Strategies and Motivations*. 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, pp. 1 - 8.

vii. Huang, T. C., Shu, Y., Chang, S. H., Huang, Y. Z., Lee, S. L., Huang, Y. M. and Liu, C. H., 2014. Developing a Self-Regulated Oriented Online Programming Teaching and Learning System. *Teaching, Assessment and Learning (TALE)*, pp. 115-120.

viii. Husain, M., Tarannum, N., Patil, N., 2013. Teaching Programming Course Elective: A New Teaching and Learning Experience. *Innovation and Technology in Education (MITE)*, pp. 275–279.

ix. Janke, E., Brune, P. &Wagner, S., Does Outside-In Teaching Improve the Learning of Object-OrientedProgramming?*Software Engineering*, 2, pp. 408 – 417.

x. Malliarakis, C., Satratzemi, M. &Xinogalos, S.,2016. CMX: The Effects of an Educational MMORPG on Learning and Teaching Computer Programming. *IEEE Transactions on Learning Technologies*, 1, p. 1.

xi. Seyal, A. H., Siau, N. Z. & Mey, Y. S., 2015. Evaluating Students' Perception of Teaching/Learning Computer Programming: A Study in a Bruneian Technological University. *Journal of Education and Vocational Research* (JEVR), 6(2), pp. 25-33.

xii. Shirat, R. H., 2003. *Pmr 2003: 50, 520 Tempat Disediakan Di Smt*. Available at: http://www.tutor.com.my/tutor/dunia.asp?y=2003&dt=1209&sec=Berita&pg=be_01.htm

xiii. Suliman, A. & Nazeri, S., 2014. The Evaluation Of An Embedded System Kit As A C Programming Teaching Tool. *Journal of ICT*, 13, pp. 123-138.

xiv. Suliman, A. & Nazeri, S., 2016. *Factors Influencing the Teaching and Learning of Programming*. 2016 IEEE International Conference on Teaching and Learning in Education (ICTLE2016), pp. 90-95.

xv. Tillmann, N., deHalleux, J., Xie, T., Gulwani, S. & Bishop, J., 2013. *Teaching and Learning Programming and Software Engineering via Interactive Gaming*. 2013 35th International Conference on Software Engineering (ICSE), pp. 1117 - 1126.