

Computer Programming Difficulties at Higher Secondary School: A Content Analysis

Canayah Cuniah
University of Technology, Mauritius
UTM
Mauritius
canayah.cuniah@gmail.com

Shireen Panchoo
University of Technology, Mauritius
UTM
Mauritius
s.panchoo@umail.utm.ac.mu

Abstract—Teaching and learning of computer programming has stimulated increasing awareness at the Higher Secondary School. This paper analyzed main problems in the subject of computer programming which shows that the major difficulties include learning skills, subject complexity and the teaching means and methods. A content analysis is performed to a batch of 60 heterogeneous students who are going to sit for the final year examination. It is a study to identify which concepts of computer programming are found to be difficult. Therefore the result obtained showed that students are not able to answer concepts that require analytical thinking skills. This study is an initial step of an ongoing research to build a conceptual guideline for the teaching and learning of computer programming at secondary education.

Keywords— *Computer programming skills, programming concepts, content analysis.*

I. INTRODUCTION

Computer programming is one of the core modules in computer science. Studies have shown however, that students face cognitive setbacks to apply the basic method of it to solve practical problems [1][9]. Computer programming is a challenging subject to both learners and instructors as it is required to master the language and to structure the overall logical flow of instructions [9]. Logic formulation is introduced in the form of codes which are close to English structures (pseudo-codes) and through flowchart. However, the fact that there is no application software that can generate real-time feedback for bugs while writing the algorithms; the logical errors are not identified either [4]

Among the studies conducted in the teaching and learning of computer programming, very few authors were able to identify a reliable support that would bring the best out of the students to overcome the cognitive as well as teaching mode setbacks. Moreover, lack of resources for practice, results in very poor test scores [2], and therefore, students rely more on the practical sessions done at school to acquire the programming skills also necessitate regular individual attention. In addition the time frame and the number of practical session allocations may not be sufficient to meet up with the expectations of all the learners [9] consequently leaving learners to abide by

prescribed textbooks which in turn lack ample support that could guide a learner to become self dependent to the understanding of the programming concepts which demand logical, analytical skills and conceptual perception to the area of study [4].

Computer programming skills were previously assessed from project-based coursework but recently these were replaced by written papers [3]. A change that might on one side reduce the amount of time spent to documenting the project and the other favoured rote learning where it has been proved that it is important to let students execute their program on computer system so that they can receive real-time feedback on what they have programmed whether logical or syntax errors [1]. Yet, reports have shown a constant decline in the performance of computer science at Higher School level. Moreover, examiners' statements from Cambridge declared that in contrast to candidates who were able to produce good answers, others could not answer the coding questions and unsure of the programming language they were using [3]. The impact of it, hence will directly affect the intentions in making the country a cyber hub with skilled programmers as less students are interested in opting for the subject.

II. LITERATURE REVIEW

Computer programming is difficult and challenging. The setbacks and delusions of it are due to the isolation of several factors which are interspersed. The three insights of computer programming are identified as:

A. Learning Skills

Learning to program requires a lot of skills that form a hierarchy, where a learner has to apply them at any point in time. These aptitudes are to memorize the keywords and apply ambiguous logical concepts to solve problems, unlike other subjects. Furthermore, these intricate skills are also to learn the purpose of every specific keywords and move on to structure the overall logical flow of instructions to unearth a solution. Therefore, the composition of these skills is such that it requires a level of abstraction, creative thinking and complex syntax [1]. Moreover, it has been noted that learners may know the syntax and semantics of instructions, but do not know how to combine them into valid programs [6].

B. Subject Complexity

The transition from abstraction to generating the final outputs is a translation from theoretical knowledge into algorithms design. It requires students to understand the concept and comprehend the characteristics of the notations to be able to solve a problem [9]. Learners consider the tasks too difficult and refrain to proceed further when they encounter the first complexity of the subject. The increase in difficulty leaves them with no other choice than to look for ready-made solutions, due to their limited knowledge of programming concepts [7]. One of the barriers to learning computer programming which have been identified by researchers is also the initial contact with it. The students' interest in the subject depends on how quickly and easily the initiation has been performed. The transmission from content knowledge to pedagogical aspirations determines the motivation of the recipients towards the subject and the degree of commitment even in a mounting level of abstraction [9].

C. Teaching Means and Methods

The skills and complexity of computer programming are not the only stumbling blocks but the teaching means and methods have also been found quite off-putting. An educator might find it hard to cope with mixed learning styles and the reasoning ability of each student. ICT supports to teach and learn computer programming, may vary from different institutions due to either lack of know-how or cost prone maintenance.

Previous studies have shown that only parts of the computer programming were tackled at a time. The students are initially taught the concept of writing codes which are close to English structures. However, the fact that there are no application platforms that can generate real-time feedback for bugs while writing the algorithms; the logical errors are not identified [5].

Moreover, time allocation to practice is hard to address to a class of mixed abilities and of different learning styles [9].

To adopt the technological utilities in the learning processes, learners who are from a poor socio-economic background find it difficult to afford certain resources for practice after school hours which fallout in low performances [2]. As a result, the students look for resources available at school. Furthermore, the amount of time allocated to complete the course contents may not be adequate to attain the goals. The lack of guidance and time will hence, result in relying on the prescribed textbooks [9].

III. METHODOLOGY

This research is conducted to a group of 60 heterogeneous students, using the content analysis approach and the data is obtained are analyzed. The batch consists of 37 boys and 23 girls aged from 17-18 years of age from the computer science department in an urban school.

As the study focuses in finding the difficulties of computer programming, a question paper is set containing 2 problem-based questions where the students have to manually answer within an exact duration. Each section of the questions assesses the level of understanding of a specific concept in computer programming. The first part of the paper tests the students' abilities in writing pseudo codes and the second half to program the pseudo codes using java as a programming language. The number of marks to each part of the questions is based on the level of complexity as shown in table 1. The total number of marks is seventy and the duration of the paper is two hours.

Table 1: Evaluation Mark Table

<i>Computer Programming concepts</i>	<i>Number of Marks</i>
Syntax	3
Use of data-types	5
Declaring variables	4
Nested IF	6
Loop	12
1D Array	13
2D Array	15
Logical Flow	12
Total	70

A Mark scheme is provided as a guide to a panel of four computer science teachers during the marking process. The table 1 is an abstract as the mark scheme is a document which contains the number of marks allocated for each relevant answer. A question-wise analysis is done after correction of scripts to determine the level achievement per question by each student. The different computer programming concepts tested are explained as follows:

Syntax:

It is a set of specific keywords that are used to command an instruction. These keywords vary from pseudo codes to Java programming language.

The students are initially taught to use pseudo codes to solve a problem. However, the difficulty comes when having to program using Java as the keywords are different from pseudo codes. Moreover, it becomes more confusing to the students when writing the codes on paper.

Data-type and declaration of variable:

A data type tells the program the form of data being accepted by a variable.

Declaration of variable in pseudo codes are shown without bearing specific syntax. However, in java it is compulsory to declare a variable with from a list of specific data types. Consequently the students are reluctant as to which data type to use while solving for a problem in Java programming language.

Nested Ifs and Loop

Nested ifs are conditional statements whereas loops are the execution of statements several times until a condition is met.

These concepts require logical thinking skills as there can be various ways which can be implemented within a program. Therefore, the students are unsure and confused as to when and where these concepts are applied so as to solve and interpret logically.

Array

It is a concept which can represent several memory locations within a computer bearing the same variable name.

Both 1Dimension (1D) and 2Dimension (2D) arrays are quite ambiguous concepts to the students as they are not able to code it properly. Many of them are hesitant as to where to include the array within their program.

Logical Flow

The flow while writing a program produces the required output.

A list of statements correctly written and logically placed generates relevant output to a problem. From the evaluation process it has been noted that many students have the tendency to write statements without a proper course hence, resulting to illogical results.

General Observation:

Firstly the switch from using pseudo code to coding using programming language creates confusion in the mind of the students. Secondly, having to memorise the different keywords along with concepts to be used are problematic to the students. Finally, writing codes on paper is problematic as the students have on one hand to memorise all the programming concepts and syntax while on the other have to think logically.

IV. RESULT AND ANALYSIS

An evaluation process is carried out to project how many students are scoring maximum marks in each concept. The test assesses the students' ability to provide good answers based on each problem stated in the question by applying proper use of the computer programming concepts. A bar-chart is used to show how many students were able to score maximum marks in each of the concept as shown in Fig 1.

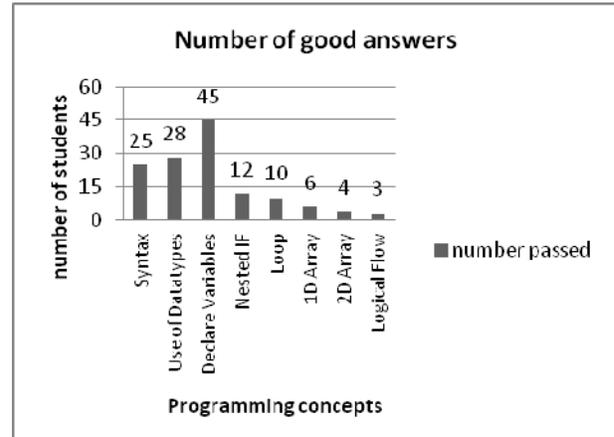


Figure 1: Number of good answers per concept

As the computer programming concepts gets more analytical and logical fewer students are able to score maximum marks. The batch of students has prior knowledge on computer programming. However, they are reluctant to the application of those concepts when it comes to solve problem-based questions. The number of students who got maximum marks in each concept is shown in Fig 1.

Students are able to show an understanding in memorizing the syntax and the use of data-types. However, as the total number of students who sat for the paper is sixty therefore, it is relatively low regarding the fact that the students had prior knowledge and understanding. This is due to the complexity of the subject itself and the application of ambiguous concepts [10].

Declaring and assigning values to variables are relatively impressive as forty-five out of sixty students have a proper control over it. However, as the concepts require more cognitive aptitudes the number of good answers tends to decrease drastically [6].

Concepts like conditional statements; loop and arrays have shown to be complex for the students to master. Besides, the logical formulation in the students' solutions is found to be very alarming where only three of them manage to achieve. During practice when the concepts get more complex the students tend to go for ready-made solutions and apply them to the actual problem.

The concept on 1D Array, only six students have managed to answer correctly and four students in 2D Array. The initial contact with difficult concepts makes it hard for the students to grasp as it is dependable on the students' interest in the subject [10]. Moreover, having to switch from learning of pseudo codes to writing codes in programming languages along with the mounting level of abstraction the students are less motivated in opting for the subject.

V. CONCLUSION AND FUTURE WORK

This research paper presents the difficulties secondary students are facing in learning computer programming. The setbacks are not only in the learning but also the subject itself and the strategies used to deliver. However, the study did not explore modes of teaching and assessments methods for computer programming. As an ongoing research, the creation of an educational framework is being devised that will act as a guide to both learners and teachers. The framework proposes solutions to the teaching and learning of computer programming. The initial phase of the model is an intelligent agent. It is being developed to help the students obtained real time feedback while coding java statements. The agent is going to provide customised feedbacks about each statement

which is going to help students to program on their own while correcting the syntax errors.

REFERENCES

- [1] Azizah Suliman, R. H. M. O., 2011. A Preliminary Study on Teaching Programming at Malaysian School. *ICOCI*, Issue 082, pp. 220-224.
- [2] Barry J, 2006. The Effect of socio-economic status on academic achievement, Kansas: Wichita State University.
- [3] Cambridge, 2013. *teacher support*. [Online] Available at: <http://cie.co.uk> [Accessed february 2013].
- [4] Carlile O, 2005. Rearranging the mental furniture: Reflective practice of the Teacher. *Scientific*, pp. 32-40.
- [5] Craig and Sarlo, 2012. *Improving Adolescent Literacy*. New York: Inc Larchmont.
- [6] Essi Lahtinen, K. A.-M. H.-M. J., 2005. A Study of the Difficulties of Novice Programmers. *IitiCSE*, pp. 14-18.
- [7] Jakub Swacha, 2010. New Concepts for Teaching Computer Programming to Future Information Technology Engineers. *MEMSTECH*, pp. 188-191.
- [8] Nelson and Rice, 2000. Introduction to Algorithms and Problem Solving. *30th ASSEE/IEEE Frontiers in Education Conference*.
- [9] Rongfand Gao, 2011. Reforming to Improve the Teaching Quality of Computer Programming Language. *6th International Conference on Computer Science & Education*, pp. 1267-1269.
- [10] Said Hadjerrouit, 2007. A blended Learning Model in Java Programming: A Design-Based Research Approach, Norway: Adger University College, Kristiansand.