

## **Understanding the Common Mistakes made by Fundamentals of Computer Problem Solving (CSC128) Students of Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia on Repetition and Functions Topic**

Wan Anisha Wan Mohammad, Naemah Abdul Wahab and Azlina Mohd Mydin  
*wanan122@uitm.edu.my, naema586@uitm.edu.my, azlin143@uitm.edu.my*

Jabatan Sains Komputer & Matematik (JSKM), Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia

### **Introduction**

Computer programming is the process of writing and designing a computer programs while a computer program is a set of instructions to solve problems (Anisha et. al., 2019). Learning computer programs need to start from the basic level before anyone can go to a higher level. Thus, it is very important for non-programmers to understand the basic concepts of programming before they are able to solve a complicated one.

Rogalski and Samurcay (1990) summarise in their study that the process of acquiring and developing computer programming knowledge includes numerous cognitive activities and mental representations associated with program design, program comprehension, program debugging, program modifying and lastly program documentation. They added that programmers should be able to construct the theoretical understanding into schemas and plans using basic control structures such as conditional statements, loops, functions and others.

Other than that, a study by Saeli et. al. (2011) mentioned that Govender (2006) indicated three main parts of learning to program are the data, instruction and syntax. Data refers to the variables and data types of a computer language while instruction is the control structures (conditional statements, loops and others) and subroutines in the programming. Syntax represents the rules and vocabulary of a programming language.

## **Fundamental of Computer Problem Solving (CSC128) for Novice Programmer**

As mentioned in our previous paper, Fundamentals of Computer Problem Solving (CSC128) is a course specifically for beginners or those who are new to programming. This course only covers few chapters which involved the basic concept of computer problem solving.

Since this course is being taken by the non-IT students or the novice programmers, they will only learn the fundamental part of programming using the C++ programming language. The topics involve the introduction to programming where students will learn few terms on programming and understand the process in developing a program which is by using the Program Development Life Cycle. Before moving to the control structures, students must also understand the basic elements in programming such as how to declare variables and constant, data types, input, and output as well as operation statement. The control structures will enable the students to write programs using sequential, selection and repetition. Finally, functions are being introduced to the students so that they will be able to construct the program into smaller pieces.

## **Common Mistakes made by Students in CSC128 Course by Chapters**

CSC128 is the fundamental course on programming taken by the engineering students. Since the students do not have any programming background, it is very crucial for the lecturers to make sure that the students understand what they will be learning to make the learning process more interesting in order to gather the students' attention.

Learning is a process of understanding or acquiring knowledge. However, it is normal if the students make some mistakes during the learning process. Below are the common mistakes that have been identified during the learning process of this course. In this part, we will show the common mistakes done by most of the student on the repetition control structure and functions topic.

### ***Repetition control structure***

In repetition control structure, it allows a program to be repeated based on certain condition. Thus, students must be able to understand the Loop Control Variables or LCV and how the loop works. Table 1 shows the common mistakes done in Repetition Control Structure.

Table 1: Common Mistakes in Repetition Control Structure

<b>Chapter 4</b>	<b>Common Mistakes</b>
Flow Chart on Repetition	✘ Most of the students do not have problem in writing programs on repetition. However, when it comes to drawing flowchart which involves repetition, students tend to make mistakes whereby they do not show how the repetition happens in the flowchart. In the flowchart, repetition is being shown by using the flow line symbol that goes back to check the condition. Once the condition is TRUE, the program will be repeated but once the program is FALSE, the program will skipped all the statements.
Writing Initial Value	✘ Before any condition in a repetition is being checked especially for a WHILE loop, an initial value should be entered or initialized first. Some of the students do not give input or initialize values to the variables and this will unable the condition to be read as the value is not given yet. One important thing that student must understand here is the Loop Control Variable which is the starting value that must be given to a variable.
Infinite Loop	✘ Infinite loop happens when there is no counter which controls the loop condition. This is another important concept that the student must understand in Loop Control Variable which is the function of the counter is to control the number of time the loop is being repeated.
Wrong syntax	✘ Student should understand the syntax on writing the three types of loop which is the while loop, do...while loop and for loop. Because lack of practices, some of the students is confused by the different syntax for these three types of control structure and will make mistakes when using it.
Tracing	✘ Tracing a loop needs a lot of patience. As long as the condition is TRUE, a loop will continue looping and when it is FALSE, a loop will stop. When tracing, students must identify the types of loop used and be able to trace the loop especially when it involves nested loop where in nested loop, the outer loop takes control of the inner loop. The outer loop will run once, follow with the inner loop which will run until the condition is FALSE. The program will then go back to the outer loop and will repeat back the steps as long as the outer loop is TRUE. Program will only stop until the outer loop is FALSE.

All the mistakes discussed above can be avoided if the students do a lot of practices. In any programming lesson, doing a lot of exercise will help the students to understand much better.

### **Function**

In real world situation where programs are written by team of programmers, function is being done as each of the programmers only does their part. The advantage of functions is that it will separate the programs into smaller one. However, since student are used to write programs in only one main function from chapter 1 to chapter 4, they are not used to function and it is quite hard for them to understand the concept of function. Since there are many ways to do function, Table 2 below are the common mistakes done in Functions.

Table 2: Common Mistakes in Functions

<b>Chapter 5</b>	<b>Common Mistakes</b>
Understanding the concept of function	✘ Before starting to do a function, students must be able to understand the concept of function and why it needs to be done. In function, what is important is to understand the three function elements which are function prototype, function definition and function call.
Function prototype	✘ In function prototype, what student must know is function prototype is only written if the function definition is done below the main function. If a function definition is done above the main function, a function prototype is not needed.
Function definition	✘ In English, definition is also known as meanings. Students are confused with this term so they tried to give the definition of function instead of writing the correct function definition.
Function call	✘ A function needs to be called in order for it to be useful. However, there are many ways to call a function. Some of the students did not call the function while some of them do not understand the few ways of calling a function.
Function with/without return statement	✘ Sometimes, students are not sure to start a function definition with either void or other function types. They also must understand that a function which starts with void does not have to return a value while a function that starts with other function types MUST return a value.
Global and local variables	✘ In functions, variables are declared either globally or locally. Some students make mistakes especially when it comes to declaring a local variable as they do not know that when using local variables, parameter passing is needed. The students must also understand the concept and ways of passing a parameter to a function.
Passing parameter -by value and by reference	✘ This is very important when students used local variables in their functions. Student must know that a function must only return one value at one time. The concept of passing parameter is very crucial that if a function needs to return more the one value, passed by reference is needed in a program.

To understand this topic, lecturers should expose the students with real world situation so that they can understand the topic better. Students must also try to solve many problems which involve functions so that they can be used to it.

## **Conclusion**

Learning programming is fun if the students know how to solve problems given to them. However, to make them understand more, students need to do a lot practices instead of just reading the books. To make programming more interesting, lecturers should play an important role so that students will love programming. It is also hoped that this paper will help the students to avoid all the common mistakes that they did when learning programming especially in their assessment.

## **References:**

- Anisha, W., Azlina, M. (2019). *Introduction to C++ Programming* (2nd ed.), Oxford Fajar
- Govender, I. (2006). *Learning to Program, Learning to Teach Programming: Pre-and In-service Teachers' Experiences of an Object-oriented Language* (Doctoral dissertation, University of South Africa).
- Rogalski, J., & Samurçay, R. (1990). Acquisition of programming knowledge and skills. In *Psychology of programming* (pp. 157-174). Academic Press.
- Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching programming in Secondary school: A pedagogical content knowledge perspective. *Informatics in education*, 10(1), 73-88.