Suggested Questions for Non-Computer Science Students' Assessments Using Bloom's Taxonomy in Programming Context.

Rozita Kadar, Saiful Nizam Warris and Syarifah Adilah Mohamed Yusoff
rozita231@uitm.edu.my, saifulwar@uitm.edu.my, syarifah.adilah@uitm.edu.my

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

Introduction

Introduction to programming language is a course that offered to several programmes in UiTM. This course will emphasize on the introduction of computer problem solving with the use of C++ programming language to illustrate the solution. The aim is to develop students' problem-solving strategies, techniques and analytical skills that can be applied to computer field or in other areas as well as to use in their subsequent course work and professional development. Students are learned to identify a problem, design appropriate solution and solve the problem in computerized way.

At the beginning, students are introduced to the basic elements in programming. At this stage, students able to declare variables, applying mathematical operation and able to construct a simple programming statement. Then, student will be exposed to the used of selection control structure and repetition control structure. The implementation of functions and array are also discussing in this course. At the end of semester, students are able to write a complete programming by applying all the elements that are discussed in the course.

Construction of Questions based on Bloom's Taxonomy

One way to measure students' skills in mastery of programming is through assessments. This article proposes the role of Bloom's Taxonomy in the preparation of student assessments. Bloom's Taxonomy was created by Benjamin Bloom (1956) which proposed the six different levels of understanding which are: remember, understand, apply, analyse, evaluate, and create. This section discusses the six level of understanding proposed by Benjamin Bloom and the summary of Bloom's Taxonomy that explores by Thompson et al. (2008) is discussed below:

a) **Remember**

- Retrieving relevant knowledge from long-term memory
- Includes recognising and recalling.

Example of assessment terms:

- *Identifying a particular construct in a piece of code.*
- *Recognising the implementation of a subject area concept.*
- *Recognising the appropriate description for a subject area concept or terms.*
- *Recalling any material explicitly covered in the teaching programme. eg. conceptual definition.*

b) **Understand**

- Constructing meaning from instructional messages, including oral, written, and graphical communications.
- Includes interpreting, exemplifying, classifying, summarising, inferring, comparing, and explaining.

Example of assessment terms:

- *Translating an algorithm from one form of*
- *representation to another form.*
- *Explaining a concept or an algorithm.*
- *Presenting an example of concept or an algorithm.*

c) **Apply**

- Carrying out or using a procedure in a given situation.
- includes executing and implementing.

Example of assessment terms:

- *The process and algorithm are known to the learner and both are applied to a problem that is familiar, but that has not been solved previously in the same context.*
- *The process and algorithm are known to the learner, and both are applied to an unfamiliar problem.*

d) Analyse

- breaking material into its constituent parts and determining how the parts relate to one another and to an overall structure or purpose.
- Includes differentiating, organising, and attributing.

Example of assessment terms:

- *Breaking a programming task into its component*
- *parts (classes, components, etc.).*
- *Organising component parts to achieve an overall objective.*
- *Identifying critical components of a development.*
- *Identifying unimportant components or requirements.*

e) Evaluate

- Making judgements based on criteria and standards.
- Includes checking and critiquing.

Example of assessment terms:

- *Determining whether a piece of code satisfies the requirements through defining an appropriate testing strategy.*
- *Critiquing the quality of a piece of code based on coding standards or performance criteria.*

f) Create

- Putting elements together to form a coherent or functional whole; reorganising elements into a new pattern or structure.
- Includes generating, planning, and producing.

Example of assessment terms:

- *Coming up with a new alternative algorithm or hypothesising that a new combination of algorithms will solve a problem.*
- *Devising an alternative process or strategy for solving a problem; or complex programming tasks, this might include dividing the task into smaller chunks to which they can apply known algorithms and processes.*
- *Constructing a code segment or program either from an invented algorithm or through the application of known algorithms in a combination that is new to the students.*

Suggested Questions based on Bloom's Taxonomy

As non-computer science students, there are some important things that they need to learn in programming. This section proposes the features of programming language that the students need to be expert while learning programming which are divided into sections: introduction to programming; basic elements of programming; selection and repetition control structures; function; as well as array. From these sections, this article proposed the questions that should be assessed on students. Bloom's Taxonomy is used as a guideline in preparing the questions. The suggested questions are listed below:

a) Introduction to Programming

- Define/explain the terms: program/programming, source code/ source file, integer, floating point, object code, assembler, compiler, interpreter, types of programming design approach, types of error, algorithm.
- Briefly explain the importance/advantages of programming.
- Identify the difference/distinguish between machine language and high-level language; assembler, compiler and interpreter.

- Write/insert comments in a program, how the compiler responds to the comments.
- Identify/briefly explain the steps in Program Development Life Cycle (PDLC).
- Illustrate/Write a pseudocode or draw a flowchart based on a problem situation and vice versa.
- Write/Insert indentation in a given program segment.

b) Basic Element of Programming

- Identify the rules of naming identifier and determine the validity of identifier, how an identifier is related to reserve word.
- Write a declaration statements and assignment statements for variables and constants to accomplish the given sentences.
- Evaluate/trace the mathematical expression based on certain input – basic operations (+.-./.*.%), function pow(), sqrt(), etc.
- Postfix increment and prefix decrement - a++, a--, ++a, --a., unary and binary operators, compound assignment.
- Given a problem situation and transform to C++ program segment.
- Given a program and find/label syntax errors.
- Convert algebraic equation into C++ statement.
- Construct/evaluate Boolean expression using relational & logical operator.
- Write a formatting statement to display decimal number.

c) Control Structure: Selection and Repetition

i. Selection Control Structure

- Discuss the types of selection control structure
- Transform/convert into IF..ELSE/SWITCH..CASE statement based on the given problem situation.
- Trace program segment and show the output based on the given input.
- Write a complete program that include the combination of selection & repetition control structure.

ii. Repetition Control Structure

- Discuss the types of repetition control structure
- Identify Loop Control Variable (LCV), starting value, loop condition, updating condition.
- The difference/distinguish between while and do..while
- Trace a program segment and show the output based on the given input.
- Write program segment base on the given problem situation.
- Rewrite using another loop structure based on the given loop.
- Find the number of loops.
- How to avoid infinite loop from the given program segment.
- Write a complete program that include the combination of selection and repetition control structure.

d) Function

- Discuss the types of function and its elements: function prototype, function call, function definition.
- User-defined function-strcpy(), strlen(), etc.
- Identify the types of parameter: actual and formal
- Identify the types of variable: local and global
- Identify the types of parameter passing: by values and by references
- Write a function definition based on the given problem situation including receives and return values.
- Trace a program segment including receives and return value and show the output.
- Write a complete program by constructing functions including the combination of selection and repetition control structure.

e) Array

- Write a declaration and assignment statement to an array.
- Write a program segment to input and display data in array.
- Find the last index number or value in array.
- Trace program segment and show the output based on the given input.

- Write a complete program by constructing arrays including the combination of function, selection and repetition control structure.

Conclusion

This article has discussed the role of Bloom's Taxonomy as a guideline in the preparation of assessments in the context of programming. It is a very important to look thoroughly on the cognitive processes while preparing the assessments and needs to be taken seriously so that students' ability to master the programming knowledge can be measured more efficiently. With this study, it is hoped that educators will benefit from it by generating more discussion and more an important is the quality of assessments can meet current needs.

References:

Bloom, B. S. (1956). *Taxonomy of educational objectives*. Vol. 1: Cognitive domain. New York: McKay, 20-24.

Mohamad,W, A & Mydin, A. (2019). *Introduction to C++ Programming(2nd.)*. Selangor: Oxford Fajar Uni. Press.

Thompson, E., Luxton-Reilly, A., Whalley, J. L., Hu, M., & Robbins, P. (2008, January). *Bloom's taxonomy for CS assessment*. In Proceedings of the tenth conference on Australasian computing education-Volume 78 (pp. 155-161).