UNIVERSITI TEKNOLOGI MARA

COMPASS

Compilation of Pahang Scholars' Synergy

Year 2011

ISSN 1985-9937

Computer Problem Solver (CoPS): A Better Understanding of Problem Solving in Computer Programming

Roslan Sadjirin, Ikhsan Md Raus, Haslinda Noradzan, Nursyahidah Alias, Nor Zalina Ismail, Mohd Norafizal Abd Aziz, Muhd Eizan Shafiq Abd Aziz, Zazaleena Zakariah, Fazlin Marini Hussain, Siti Nurbaya Ismail, Norhafizah Hashim

ABSTRACT

The introduction course to computer programming is considered one of the most difficult courses by Computer Science students in Universiti Teknologi MARA Pahang (UiTM). The students' results for every semester in the Computer Programming course is always average. In order to enhance understanding of programming concepts and the skills in problem solving, an improvisation of the teaching and learning method is introduced and implemented on the first year students of UiTM Pahang. Therefore, this paper presents the introduction of the teaching and learning method called Computer Problem Solver (CoPS) for the teaching and learning of the Computer Programming course to the first year computer science students in their of UiTM Pahang.

Keyword: *Teaching, Learning, Computer Problem Solving, Programming Competency*

Introduction

The Computer Programming course is one of the most important courses in computer sciences and computer engineering studies (Mancy and Reid, 2004). However, the first programming course has always been a major stumbling block for many students in computer science studies and it is known for its notoriously poor pass rate (Proulx, 2000; Mancy and Reid, 2004; Bennedsen and Carpesen., 2007). Jenkins and Davy (2000) argue that learning programming is not as easy as teaching it. While Bennedsen and Carpesen (2005) state that the most crucial part of the first programming course is the process of instructing the students about the systematic approach of the development of computer programs. However Bennedsen *et. al*, (2005) further explain that this important part is not addressed in textbooks.

Experience show that learning programming at the tertiary level is not easy as students continue to struggle even when writing the simplest computer programs. The main concern of educators or instructors is how to make the learning of the basic concepts of computer programming less complicated and easier to comprehend for the students. Greca, *et.al,* (2003) state that in a computer programming course, teaching and learning are two complex issues that are related. A good teaching approach can improve learning, and improved learning can make teaching more effective as students become more successful. Thus, both factors determine the success rate in an introductory programming course.

Many studies have been carried out and numerous methods have been designed to help students of computer sciences in improving their skills in learning and writing computer programming. These methods include object oriented approach (Kolling and Rosenberg, 1996), programming pattern design (Proulx, 2000), pair programming (Gehringer, 2003; Thomas, *et.al,* 2003), hypertext based system (Kay and Kummerfeld, 1994), information processing model (Mancy *et.al,* 2004), game-based approach (Rajaravivarma, 2005), and model-driven or instructional design (Caspersen, *et.al,* 2007).

However, the studies on teaching to write a computer program are still an active research and it seems that there is no appropriate method that can be implemented or adapted in teaching programming course. Proulx (2000) explained that even bright students can get lost when asked to write the simplest program even though the standard ways or phases of writing a computer program start from analysis, followed by design, and then implementation and eventually the testing phase. One of the reasons why the students face difficulties in writing the program is the lack of problem solving skills.

Studying computer programming requires students to think visually and critically. Thinking visually is concerned with what is needed to be solved. Whereas thinking critically is; addressing the issues on how to solve the programming problem and how to do it. Therefore, this paper presents the improvisation method of teaching the first computer programming course to the first year students of Diploma in Computer Science of Universiti Teknologi MARA (UiTM) Pahang.

Computer Problem Solver (CoPS)

This section explains the concept of the Computer Problem Solver (CoPS) method and some techniques that are used in the method.

Teaching and Learning Method

CoPS is a method of teaching computer programming that combines problem based learning (PBL) and cooperative learning techniques with the help of Basic Card for Programming. The next section discusses the PBL, cooperative learning and Basic Card for Programming.

Problem Based Learning and Cooperative Learning

Problem based learning is one of the techniques in teaching and learning method. In problem based learning, the students are exposed to the problems that are relevant and contextual to the real world situations. Furthermore, cooperative learning is a learning technique in which a small group of students, usually two to four students, with different abilities is formed and each member of the group is responsible for the learning and helping group members to learn in an informal way (Kementerian Pengajian Tinggi Malaysia, 2006).

Therefore, with the help of the KAP (Basic Card for Programming), which has been designed and with the combination of the cooperative learning and the problem based learning in question, we have come up with the improvised teaching and learning method to be used in teaching the first year Computer Programming course to Computer Science students in UiTM Pahang.

Basic Card for Programming

Basic Card for Programming is based on the idea of programming development phase. It is well known that the standard ways of computer problem solving are the same as the steps in computer programming which are:

Step (1): Understand the problem
Step (2): Identify the program's objective
Step (3): Identify the input and constant
Step (4): Identify the process; and lastly
Step (5): Identify the output.

However, in the CoPS method, some of the sequences are changed. The sequence of computer problem solving in this method are as follows:

Step (1): Understand the problem
Step (2): Identify the program's objective
Step (3): Identify the output
Step (4): Identify the input and constant; and lastly
Step (5): Identify the process

Note that step (3) in the computer programming steps becomes step (4) in the CoPS method, while step (4) and step (5) in the computer programming steps becomes step (5) and step (3) respectively in the CoPS method. The idea behind this method is that, before designing the steps' solution or algorithm for a particular problem, the student should understand, and be able to visualize and imagine what are the outputs or outcomes of the program. After knowing the outputs, the students should investigate and understand the required inputs in order to produce the correct outputs. Eventually, after the outputs and the inputs have been identified, the students will be able to formulate the processes or formulae, as well as give the suitable names for identifiers or variables for the program.

The analogy for the process of the CoPS is like preparing delicacy. For instance, if one wants to bake a cake, the type or flavour of the cake must be identified first. Hence the required ingredients to produce the cake will be much easier to identify. Next, after the type of cake and the ingredients have been identified, only then will the process and steps of producing the cake can be formulated in a simple and structured way.

Figure 1 below presents the steps for computer problem solving used in CoPS method and the steps in computer programming phase.

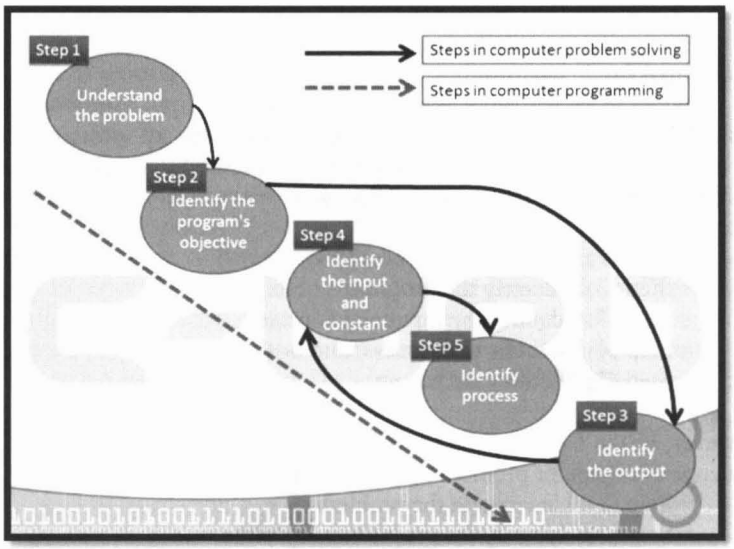

Figure 1: Steps in computer problem solving and computer programming

The lined arrow in Figure 1 shows the steps in computer problem solving used in the CoPS, whereas the dotted arrow in Figure 1 shows the steps in computer programming. Table 2 further explains the steps in computer problem solving for CoPS method as shown in Figure 1.

Table 2: Explanation of the Steps in Computer Problem Solving for the CoPS Method and the Steps in Computer Programming

Steps / Phase	Explanation
Understand the problem	Student needs to visualize overall requirements of the problem (what)
Identify the program's objective	Student needs to understand the precise requirement of the problem (what)
Identify the output	Student needs to determine the output produced by the program (what)
Identify the input and constant	Student needs to define the input and the constant of the program (what)
Identify process	Student needs to identify steps for the problem solving and transfers it into an algorithm (what and how)

Some alterations have been done in the ways of solving a computer problem for computer programming. This improvisation is done to enhance the critical thinking and visualization capability of students in computer problem solving. In the standard way of problem analysis, students are required to identify input, and then process and eventually the output (Input ⟶ Process⟶ Output).The formal process of identification is usually done in sequence which requires critical thinking by the students, hence will slow down the process of solving a problem in computer programming.

In the CoPS method, if the students are unable to visualize, it is assumed that they cannot think critically, they will not be able to come up with the correct solution. Therefore, the CoPS method improvises the formal sequence of the computer problem solving method for computer programming. The students should visualise and think of the outputs (what the outcome look

like), then identify the possible inputs. Lastly, formulate the processes from the inputs and outputs obtained (Input ⟶ Process ⟶ Output).

The interchange involved between the three phases and replaced by a new sequence of steps is as follows; (Input ⟶ Process ⟶ Output) to (Output ⟶ Input ⟶ Process). It is presumed that after determining the precise requirements of the problem in the program's objective, the students can easily recognize the desired output of the program. The next step is identifying the input and constant value that are involved to execute the process. The output requirement obtained in the previous step will help the students to discover the input and constant values that will produce the output. Lastly, the student will be able to come up with the process which consists of formulae and algorithm of the program.

Methods of Computer Problem Solver (CoPS)

There are four steps involved in the CoPS method:

1. Lecture session
2. Prepare PBL based programming tutorial for students
3. Use the basic card for programming (KAP) to solve problem for computer programming to specify output, input and process as well as to design algorithm for the program
4. Transfer the designed algorithm into source code according to steps in computer programming.

After conducting the lecture session, in a group of two to four, students are given programming questions based on a real world problem. Students are required to analyse the requirements of the problem including the output, input, process and algorithm or steps of the program by writing down the answer in KAP.

Figure 1 shows a basic card for programming (KAP) that is used to solve problems for computer programming. The final step in the CoPS method is the process of transferring the information gathered in KAP into a source code as shown in Figure 2.

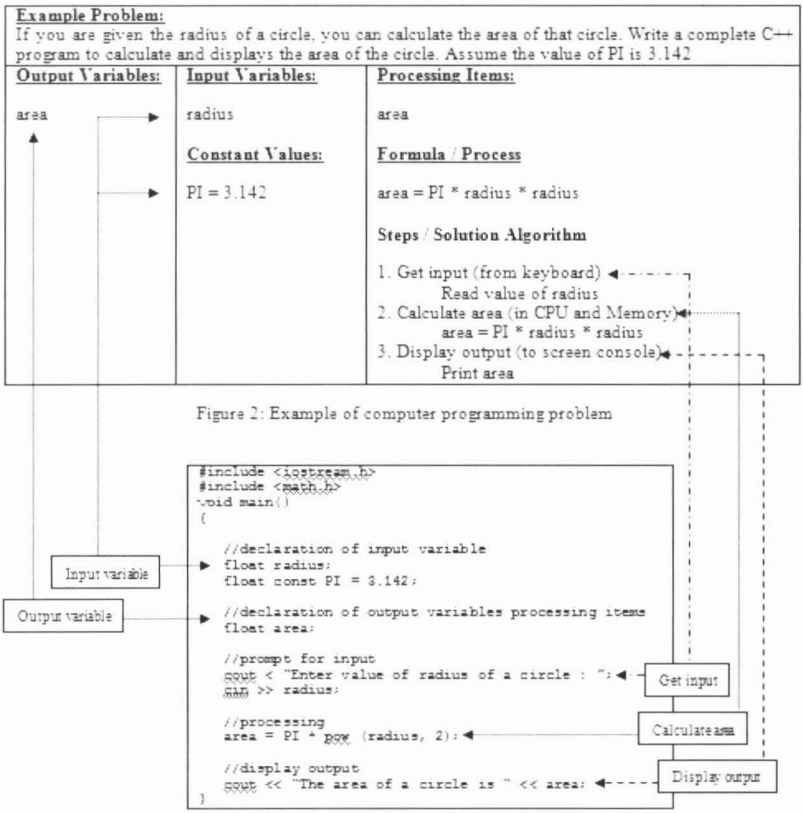

Figure 2: Example of computer programming problem

Figure 2: Example of computer programming source code in C++

Summary

The introduction and implementation of the CoPS method for teaching computer programming and computer problem solving to the first year students have instilled passion in the students as well as enhanced their capability in solving a computer problem and writing a computer program. Employing the CoPS method in teaching the computer programming can not only decrease the percentage of failure rates but also increase the number of high achieving students, as the CoPS helps students solve the computer problem and enhances their technical skills in writing a computer program. The CoPS method can also enforce the ability of the students to visualise the outcome of the computer program before embarking in the next steps of the computer programming phases.

This study suggests that the CoPS method can improve the teaching and learning method for the first year programming course and help students enhance their problem solving skills and ability to understand as well as practice the creative way of solving a computer programming problem.

References

Bennedsen, J., Caspersen, M. E, (2005). *Revealing the Programming Process*. Proceedings of the 36th SISGCSE Technical Symposium on Computer Science Education (SIGCSE'05): Vol. 37(1), pp. 186-190. ACM, New York, USA.

Bennedsen, J., Caspersen, M. E., (2007). *Failure Rates in Introductory Programming*. The ACM SIGCSE Bulletin: Vol. 39 (2), pp.32-36. ACM, New York, USA.

Caspersen, M. E., Bennedsen, J., Larsen, K. D. (2007). *Mental Models and Programming Aptitude*. Proceedings of the 12th annual SIGCSE conference on Innovation and Technology in Computer Science Education (ITiCSE'07): Vol. 39(3), pp. 206-210. ACM, New York, USA.

Gehringer, E. F. (2003). *A Pair-Programming Experiment in a Non-Programming Course*. Proceeding OOPSLA '03 Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications: pp. 187-190. ACM, New York, USA.

Greca, A., Jovanovic, V., Harris, J. (2003). *Enhancing Learning Success In The Introductory Programming Course*. 33'd ASEE IEEE Frontiers In Education Conference: pp. TC4-15-TC4-21. IEEE.

Jenkins, T, Davy, J. (2000). *Dealing With Diversity In Introductory Programming*. 8th Annual Conference on the Teaching of Computing. LTSN Centre For Information And Computer Science. Edinburgh.

Kay, J, Kummerfeld, R. J. (1994). *An Individualised Course for the C Programming Language*. Proceeding of World Wide Web Conference Series.

Kementerian Pengajian Tinggi Malaysia. (2006). *Modul Kursus: Asas Pengajaran dan Pembelajaran Pensyarah Baru IPTA*. Pusat Penerbitan Universiti, UiTM.

Kolling, M., Rosenberg, J., (1996). *An Object-Oriented Program Development Environment For The First Programming Course*. Proceedings of the 27th SIGCSE Technical Symposium on Com puter Science Education: Vol 28(1). pp.83-87. ACM, New York, USA.

Mancy, R, Reid, N. (2004). *Aspects of Cognitive Style and Program ming*. Proceedings of the Sixteenth Annual Workshop of the Psychology of Programming Interest Group: pp.1-9. Carlow, Ireland.

Proulx, V. K. (2000). *Programming Patterns and Design Patterns in the Introductory Computer Science Course*. Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Educa tion: 32(1), pp. 80-84. ACM, New York, USA.

Rajaravivarma, R. (2005). *A Games-Based Approach for Teaching the Introductory Programming Course*. The ACM SIGCSE Bulletin. Vol 37 (4), pp. 98-102. ACM, New York, USA.

Thomas, L., Ratcliffe, M. & Robertson, A . (2003). *Code Warriors and Code-a-Phobes: A Study in Attitude and Pair Programming*. Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education: Vol. 35(1), pp. 363-367. ACM, New York, USA.

ROSLAN SADJIRIN, MOHD IKHSAN MD RAUS, HASLINDA NORADZAN, NURSYAHIDAH ALIAS, NOR ZALINA ISMAIL, MOHD NORAFIZAL ABD AZIZ, MUHD EIZAN SHAFIQ ABD AZIZ, ZAZALEENA ZAKARIAH, FAZLIN MARINI HUSSAIN, SITI NURBAYA ISMAIL, NORHAFIZAH HASHIM, Fakulti Sains Komputer Dan Matematik UiTM Pahang, Universiti Teknologi MARA Malaysia, roslancs@pahang.uitm.edu.my.