



## Computer Problem Solver (CoPS) Method for Teaching First Year Computer Programming Course

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

### ABSTRACT

Poor rates of result among Computer Science students at Universiti Teknologi MARA (UiTM) Pahang in computer programming course has caused the performance, competency level as well as the student learning outcome be at par, thus affecting the KPI's performance of the Computer Science Department. In order to overcome this problem, an improvisation of the teaching and learning method is introduced and implemented. This paper presents the introduction of the teaching and learning method called CoPS (Computer Problem Solver) which has been used and implemented for teaching computer programming course among computer science students in their first year at UiTM Pahang. There are four batches of students starting from 2009 until 2011 that are involved in this study. Analysis shows that there is significant improvement in the students' performance after using CoPS method the first computer programming course.

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

### Introduction

Computer programming course is one of the most important courses in computer sciences and computer engineering study (Mancy et al., 2004). However, the first programming course has always been the major stumbling block for many students in computer science study and it is known for the notorious poor pass rates (Proulx, 2000; Mancy et al., 2004; Bennedsen et al., 2007). Jenkins et al. (2000) states that learning programming is not as easy as teaching programming.

Experience shows that learning programming in higher institutions is far from easy as the students continue to struggle even when writing the simplest computer program. The main concern of the educators or the instructors is how to make the learning of the basic concepts of computer programming become less complicated and easy to be comprehended by the students. Greca et al. (2003) states that teaching and learning are two complex related issues in computer programming. Good teaching approach can improve learning, and improved learning can make teaching more effective as student become more successful. Thus both factors determine the success rate in introductory programming course.

Many studies have been carried out and numerous methods have been designed to help students of computer sciences in improving the skills in learning and writing computer programming. These methods include object oriented approach (Kolling et al., 1996), programming pattern design (Proulx, 2000), pair programming (Gehring, 2003; Thomas et al., 2003), hypertext based system (Kay et al., 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 become lost when asked to write the simplest program eventhough 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 are face difficulties in writing the program is the lack of problem solving skills.

Studying computer programming requires students to think visually and critically. Think visually is concerned about what to be solved. Whereas think critically is addressing the issues of 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 at Universiti Teknologi MARA (UiTM) Pahang.

### Methodology

This section discusses the data collection, teaching and learning methodology when using CoPS method and the experiment that has been conducted.

## Data Collection

The CoPS method is implemented to the first semester students, who enrolled in the Diploma in Computer Science at UiTM Pahang. The CoPS method has been revised starting from December 2009 and has been fully implemented in two academic sessions in semesters June 2010 and January 2011. While in semester June 2009, the CoPS method had not been implemented yet and the method of teaching programming for that semester was by following the text book, manual and standard way of computer programming and problem solving. However, the data in semester June 2009 were also collected in this study to compare the performance of the students before and after the implementation of the CoPS method. The data collected in this study are gathered from the instructor who implemented the CoPS method in teaching programming course for first year students.

Table 1: Implementation of the CoPS Method.

Semester	Number of students	Implementation of CoPS Method
Jun 2009	60	No
December 2009	30	<i>Research, revised and improvised the teaching method available to be adapted in teaching a computer programming. CoPS method is introduced to the student.</i>
Jun 2010	88	Yes
January 2011	20	Yes
<b>Total</b>	<b>198</b>	

Table 1 shows the enrollment for the course for the four semesters of study and the implementation of the CoPS method for teaching the first programming course starting from semesters June 2009, December 2009, June 2010 until January 2011.

## Teaching and Learning Method

CoPS is a method of teaching computer programming that combines a 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 the problem based learning technique, the students are exposed to the problems that are relevant and contextual to the real world situations. Furthermore, the 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 learning and helping group mates 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 improvised as well as the combination of the cooperative learning and the problem based learning question, we have come up with the improvised teaching and learning method to be used for teaching the first year computer programming course for computer science students at 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 with 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 CoPS method, some of the sequences have been changed. The sequence of computer problem solving in CoPS method are as follow:

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 had become step (4) in CoPS method, while step (4) and step (5) in the computer programming steps had become step (5) and step (3) respectively. The idea behind this method is that, before designing the steps solution or algorithm for the particular problem, the student should understand, 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 to give the suitable names for identifiers or variables for the program.

The analogy of the process of the CoPS is like preparing for a delicacy. For instance, if one wants to bake a cake, the type or flavour of a cake to be baked must be identified first. Hence the needed ingredients to produce the cake will be much easier to be identified. 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 2 below presents the steps for computer problems solving used in CoPS method and the steps in computer programming phase.

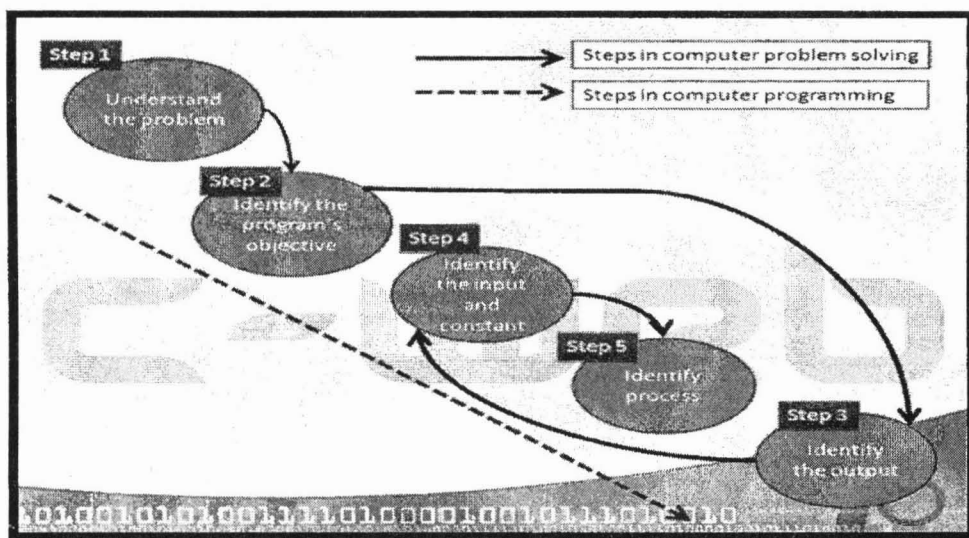


Figure 2: Steps in computer problem solving and computer programming

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

Table 2: Explanation of the Steps in Computer Problem Solving for 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 in the problem solving and transfers it to 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 à Process à Output). The formal process of identification is usually done in sequence which will actually block the visualization capability of the students, hence will slow the process of solving a problem for computer programming which is actually the critical thinking of the students.

In the CoPS method, it is assumed that if the students cannot visualize, they cannot think critically, and if they cannot think critically, they cannot come up with the correct solution. Therefore, the CoPS method improvises the formal sequence of the computer problem solving for computer programming. The students should visualize and think of the outputs (what the outcome should look like), and then identify the possible inputs and lastly formulate the processes from the inputs and outputs obtained (Output à Input à Process).

The interchange involved between the three phases and replaced by a new sequence of steps which is from (Input à Process à Output) to (Output à Input à 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 for executing 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.

## Experiment

In the experiment phase for this study, there are four steps involve in the CoPS method which are:

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 requirement of the problem including the output, input, process and algorithm or steps of the program by writing down the answer in KAP. Figure 2 shows a basic card for programming (KAP) that is used to solve problem for computer programming. The final step in the CoPS method is the process of transferring the information gathered in KAP into a source code.

<b>Example Problem:</b>		
If you are given the radius of a circle, you can calculate the area of that circle. Write a complete C++ program to calculate and displays the area of the circle. Assume the value of PI is 3.142		
<b>Output Variables:</b>  Area	<b>Input Variables:</b> radius  <b>Constant Values:</b> PI = 3.142	<b>Processing Items:</b> area  <b>Formula / Process</b> area = PI * radius * radius  <b>Steps / Solution Algorithm</b> 1. Get input (from keyboard) Read value of radius 2. Calculate area (in CPU and Memory) area = PI * radius * radius 3. Display output (to screen console) Print area

Figure 2: Example of computer programming problem

**Findings**

There are four batches of semester of studies involved in this study. Data are collected from semester Jun 2009 until January 2011. The distributions of the students' performance are shown in Table 3 and Figure 3.

Table 3: The Students' Performance for Four Semesters

Semester Of Studies	June 2009			December 2009			June 2010			January 2011		
	#	%	Cum. %	#	%	Cum. %	#	%	Cum. %	#	%	Cum. %
A+	0	0.00	0.00	0	0.00	0.00	5	5.68	5.68	1	5.00	5
A	1	1.67	1.67	0	0.00	0.00	15	17.05	22.73	2	10.00	15
A-	1	1.67	3.33	3	10.00	10.00	12	13.64	36.36	2	10.00	25
B+	3	5.00	8.33	8	26.67	36.67	5	5.68	42.05	3	15.00	40
B	4	6.67	15.00	5	16.67	53.33	5	5.68	47.73	6	30.00	70
B-	7	11.67	26.67	7	23.33	76.67	18	20.45	68.18	1	5.00	75
C+	9	15.00	41.67	4	13.33	90.00	10	11.36	79.55	4	20.00	95
C	13	21.67	63.33	2	6.67	96.67	11	12.50	92.05	1	5.00	100
C-	3	5.00	68.33	0	0.00	96.67	2	2.27	94.32	0	0.00	100
D+	7	11.67	80.00	0	0.00	96.67	3	3.41	97.73	0	0.00	100
D	7	11.67	91.67	1	3.33	100.0	2	2.27	100.00	0	0.00	100
E	5	8.33	100.0	0	0.00	100.0	0	0.00	100.00	0	0.00	100
<b>Total Students</b>	<b>60</b>	<b>100.0</b>	<b>100.0</b>	<b>30</b>	<b>100.0</b>	<b>100.0</b>	<b>88</b>	<b>100.0</b>	<b>100.00</b>	<b>20</b>	<b>100.00</b>	<b>100.00</b>

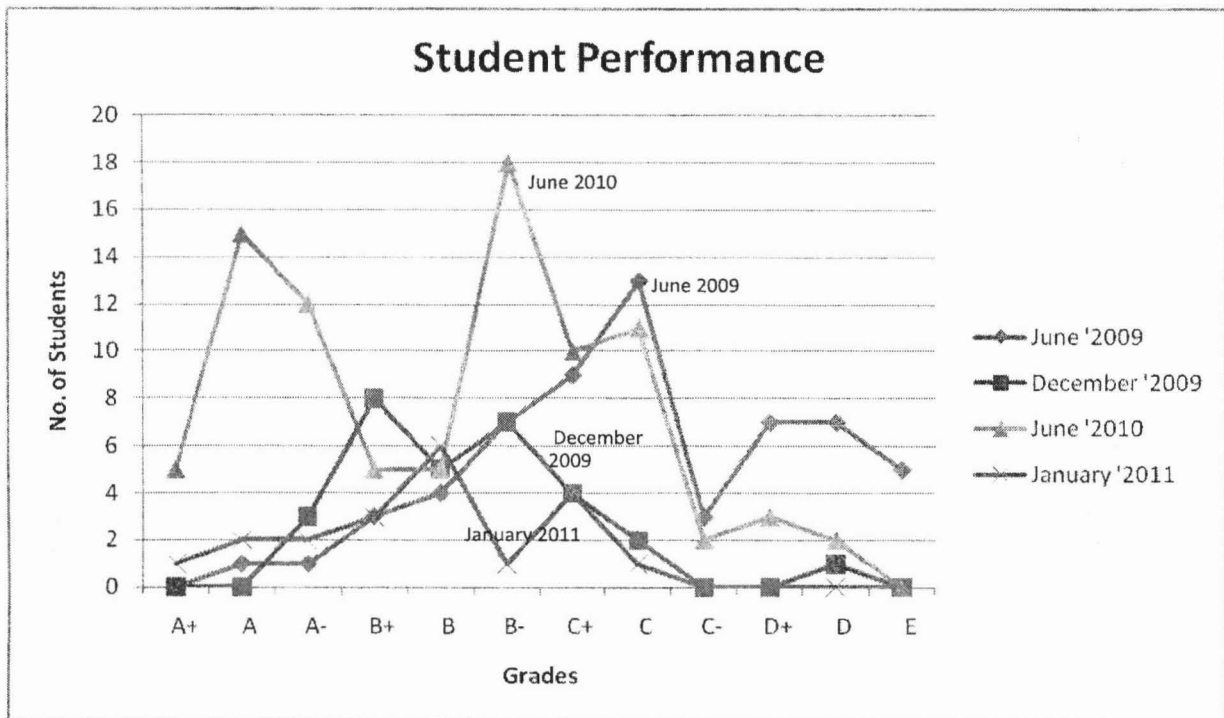


Figure 3: Students performance according to grades for four semesters (June 2009 – Jan 2011)

Figure 3 presents the students' result for four semesters which are January 2009, December 2009, June 2010 and January 2011. The line graph for semester June 2009 shows that the curve is skewed slightly to the right which signifies that the number of students who obtained poor grades is more than the good grades. However there are two students who obtained grades A and A- respectively and there are 23 students who did not reach the minimum passing grades.

While for semester December 2009, there is improvement on the good grade obtained by the students as compared to the previous semester although there is still one failed student recorded. The line graph for semester December 2009 shows the binomial distribution that indicates the proportion of students who obtained good and poor grades almost equivalent.



The data in semester June 2010 shows that there is significant improvement in the number of students who obtained good grades. There are five students with A+, 15 students with A, and 12 students with A-. However, it is reported that seven students failed to obtain the minimum passing mark. Nevertheless, as compared to the previous semester, there is positive improvement on the students' grade for semester June 2010.

Whereas for semester January 2011, the line graph shows that there is no student reported to fall below the minimum passing grades. The line graph for semester January 2011 shows that the curve is positively skewed and it signified that there are more students who obtained good grades as compared to the minimum grade.

Figure 4 and Figure 5 present the percentage trend of pass and fail for the introductory programming course. The line graph in Figure 4 shows the increment of percentage of students who passed the course. In semester June 2009 the percentage is 63.33 percent. While in semesters December 2009 and June 2010, in which the CoPS method is introduced to the students, the percentage of students who passed the course is 96.67 percent and 92.05 percent respectively. Percentage of passes in semester June is slightly dropped about 4.78 percent compared to semester December 2009. However, the percentage is considered increase more than 50 percent as compared to semester Jun 2009. In semester January 2011 the number of students who passed the course is reported at 100 percent.

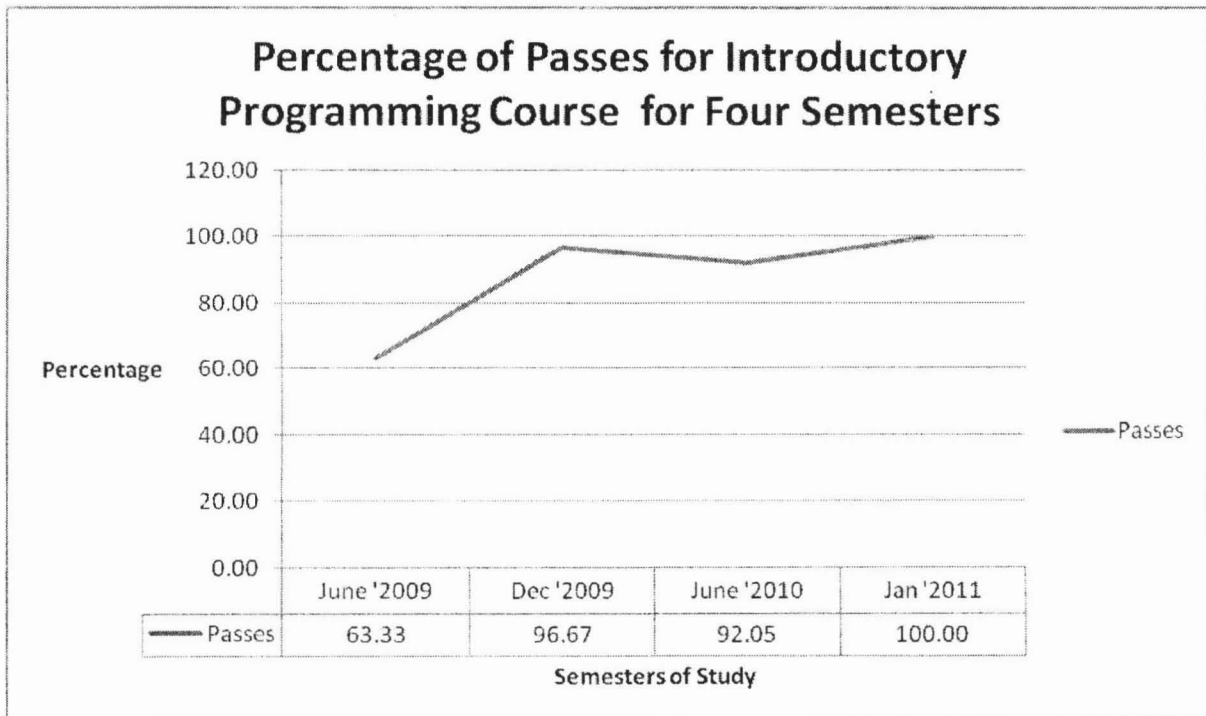


Figure 4: Percentage of students who obtained the minimum passing grade (June 2009 – January 2011)

In Figure 5, the trend of the data shows the decline in the percentage of the students who. In semester June 2009, the percentage of students who failed the course is 36.67 percent and this is not meeting the KPI of the department that required at most only 30 percent of failure rate. However in semester December 2009, the failure rate dropped more than 91 percent compared to previous semester, which only 3.3 percent reported to have failed. While in semester June 2010, the percentage of failure rate is 7.95 percent which increased more than 100 percent as compared to semester December 2009 but the rate is still in satisfactory conditions. In semester January 2011 in which the CoPS is fully implemented, the percentage of failure is zero percent.

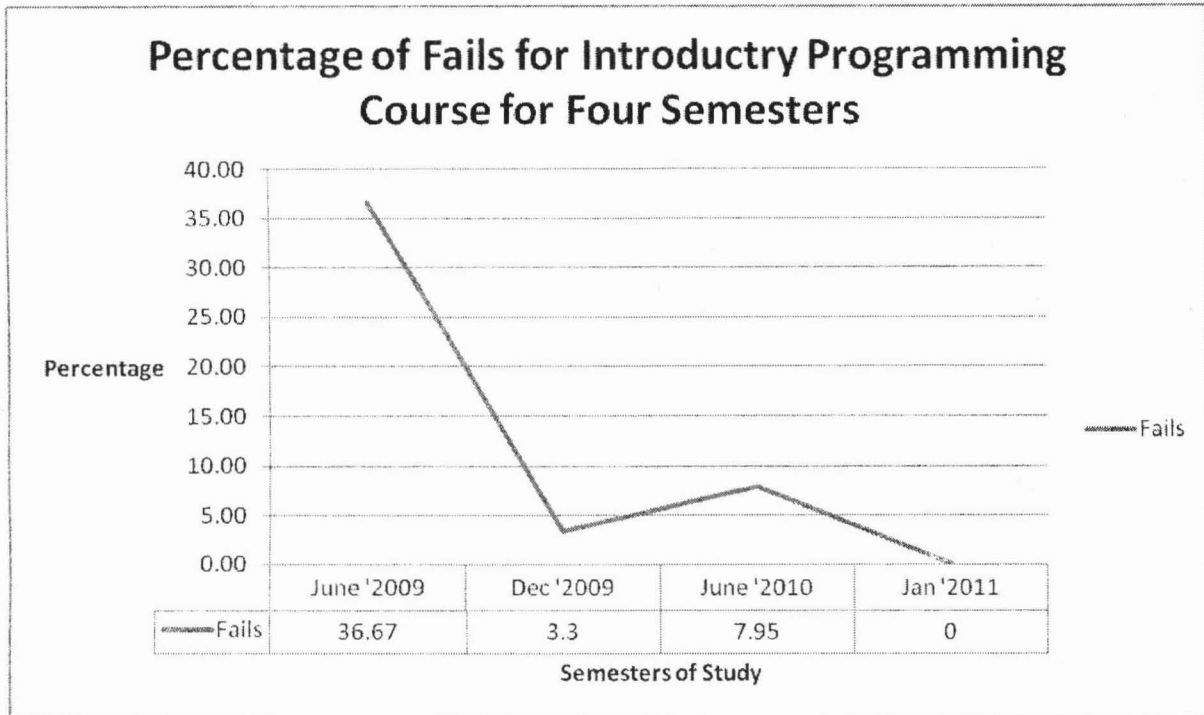


Figure 5: Percentage of students who did not obtained the minimum passing grade (Jun 2009 – January 2011)

## Conclusion

The introduction and implementation of the CoPS method for teaching computer programming and computer problem solving in the first year of study for students who enrolled in Diploma in Computer Science at UiTM Pahang has showed positive impact to the passion and capability of students in solving a computer problem and writing a computer program. Employing the CoPS method in teaching computer programming at UiTM Pahang has not only decreased the percentage of failure rate but also increased the number of good grade students, as the CoPS helps students to solve the computer problem and enhance their capability in writing a computer program. The CoPS method has also enforced the ability of the students to visualize the outcome of the computer program before embarking in the other steps of computer programming phases. This study suggests that the CoPS method which has been implemented to improve the teaching and learning method in the first programming course has helped students in enhancing their problem solving skills and ability to understand as well as to practice the creative way of solving a computer programming problem.

## References

- 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 I*

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 Computer Science Education: Vol 28(1). Pp.83-87. ACM, New York, USA.
- Mancy, R & Reid, N. (2004). *Aspects of Cognitive Style and Programming*. 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 Education: 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, HASLINDA NORADZAN, NURSYAHIDAH ALIAS, NOR ZALINA ISMAIL, MOHD NORAFIZAL ABD AZIZ, MOHD IKHSAN MD RAUS, 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, haslindanoradzan@pahang.uitm.edu.my, syahidah@pahang.uitm.edu.my, nza1601@pahang.uitm.edu.my, mnorafizal@pahang.uitm.edu.my, mohdikhsan@pahang.uitm.edu.my, eizan@pahang.uitm.edu.my, zazaleena@pahang.uitm.edu.my, fazlinmarini@pahang.uitm.edu.my, sitinurbaya@kedah.uitm.edu.my, fiza\_hz@pahang.uitm.edu.my.