Designing Programming Exercises with Computer Assisted Instruction*

Fu Lee Wang[1], and Tak-Lam Wong[2]

[1] Department of Computer Science, City University of Hong Kong,
Kowloon Tong, Hong Kong
flwang@cityu.edu.hk
[2] Department of Computer Science and Engineering,
The Chinese University of Hong Kong,
Shatin, Hong Kong
wongtl@cse.cuhk.edu.hk

Abstract. Teaching of computer programming has created significant difficulties to both teachers and students. Large class size is one of the major barriers to effective instruction. A well-designed pedagogy can make the instruction most effective. This paper will share our experiences of teaching programming courses with large class size. A set of programming exercises have been designed with help of computer assisted instruction. Evaluation has showed that the new pedagogy provide great flexibilities to both teaching and learning of computer programming. The students' academic results have been significantly improved in programming courses.

Keywords: Computer programming, large class size, stepwise learning, teaching and learning, computer assisted learning (CAI).

1 Introduction

Learning computer programming has been known to be difficult for many beginners (Boulay, 1989). A number of challenges have been identified for both teaching and learning programming (Sleeman, 1986). A programming course typically has a large class size. Large class size is one of the major barriers to effective instruction. It is difficult to closely monitor individual student's learning progress. The teachers do not have enough time to interact with all students in a class of hundreds of students within a few hours of lectures and tutorials each week. Teaching and learning computer programming has created significant difficulties to both teacher and students.

It has been showed that computer-assisted instruction (CAI) technology can be a more effective way of teaching introductory programming courses (Anderson &

* The work described in this paper was substantially supported by a grant from City University of Hong Kong (Project No.: 6000144).

Skwarecki, 1986). This paper will share our experience in using CAI technology to teach computer programming with large class size.

Students taking computer programming courses very often come with various backgrounds and ability levels. We have incorporated several teaching strategies in designing our teaching and learning activities for computer programming courses. We have designed programming exercises with different levels of difficulty to fulfil the need of students with various backgrounds and ability levels. We can ensure that each step is learned by stepwise learning (Schulman, 2001). The CAI technology allows us to have a close monitoring of students' learning progress. Moreover, we have designed programming activities in an incremental manner, so that the students gain the knowledge of large application development by implicit learning (Berry, 1997). This experience prepares the students ready to participation in a software development team.

The CAI technology provides great flexibilities for us to render the teaching and learning of computer programming more effectively. The students' academic results have been significantly improved. Students find the learning computer programming become interesting, and their programming skills are enhanced subsequently.

The rest of this paper is organized as follow. Section 2 gives an overview of computer-assisted instruction systems for computer programming. Section 3 describes the development of programming activities with multiple levels of difficulties. Section 4 presents incremental style of programming activities. Section 5 evaluates the new pedagogy. We conclude our work in Section 6.

2 Computer-Assisted Instruction for Computer Programming

Related research has showed that computer-assisted instruction (CAI) technology can be a more effective way of teaching introductory programming courses - for certain populations (Anderson & Skwarecki, 1986). Programming skill has to be acquired through lots of practice (Cheang, Kurnia, Lim & Oon, 2003). With the support of CAI, we are able to provide adequate practices to students.

Instant support to the students is a critical factor to the success of teaching and learning of computer programming. However, it introduces a huge pressure in the resources, and it may not be affordable by some universities. It has been showed that intelligent computer-assisted instruction technology can be a more effective way of teaching introductory computer programming courses (Anderson & Skwarecki, 1986). We have implemented a computer-assisted instruction system to support our teaching of computer programming courses. The detail functionalities, design and implementation can be found in (Choy, Nazir, Poon & Yu, 2005; Yu, Poon & Choy, 2006). Figure 1 shows the Programming Assignment aSsessment System (PASS). The PASS system is a web-based computer-assisted instruction system for computer programming (Choy, Nazir, Poon & Yu, 2005; Yu, Poon & Choy, 2006). The PASS system is a fully automated system to help students to study programming.

The PASS system allows the teachers to setup some programming problems. The teachers provide the input and the corresponding output to each test case. The students then submit their program for testing. The system automatically complies and executes the program submitted. By comparing the outputs generated by the students' program and the expected output provided by the teachers, the system will then provide feedbacks to the students. For example, if a student gets wrong in a certain type of inputs, the system will show the attached annotation provided by the teachers to give some hints of possible mistakes to the student. The instant feedback provided by the system provides concrete assistances to students to revise their programs, and debugging will become more interesting.

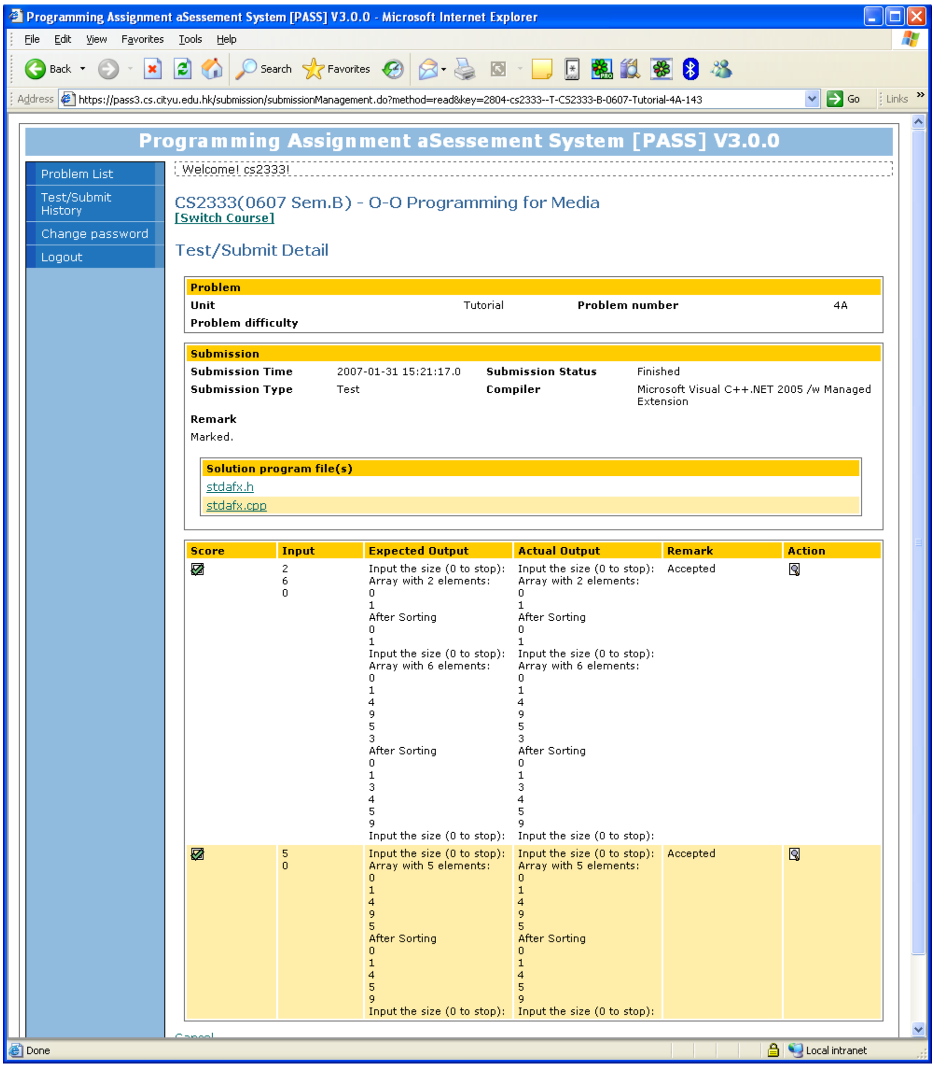

Fig. 1. Programming Assignment aSessement System (PASS)

A number of computer-assisted programming learning systems have been developed. The PASS system has a lot of advantages over the existing systems (Choy et. al., 2007). The program submitted will be tested automatically against a large number of test cases. The system will test whether the program is correct with respect to the selected test cases. When the submitted program is incorrect, PASS will indicate to the student exactly at which position the actual output differs from the correct output. Some teacher may store some predefined comments with some specific patterns of mistakes in the system. The system will show the comment to the student to help the student to debug. This kind of prompt feedback to students was rarely possible before PASS was developed (Choy et. al., 2005).

The PASS system has been used in programming courses since 2004, and it has been evolved to its third version. The system currently support teaching and learning of a number of programming languages, including C, C++, Java and Pascal. The system is highly evaluated by both students and teachers. PASS allows a tailor-made learning pace and style for individual student. It has provided a quick and convenient channel for students to test their work without manual involvement. Instant feedback to students encourages them to enhance their programming skills. The introduction of PASS has made the learning of computer programming more rewarding than before.

In the following sections, we will discuss how teaching strategies are incorporated with the intelligent computer-assisted instruction system.

3 Programming Activity with Multiple Levels of Difficulties

Effective instruction involves working the content to provide stepwise learning which checks along the way to assure that each step is learned (Schulman, 2001). It is important to ensure that students are well-trained in the fundamentals to the extent that they can eventually consider some problems with high-level complexity. We pay extra care to design the teaching and learning activities to incorporate stepwise learning.

Students taking computer programming courses very often come with various backgrounds and ability levels. The PASS system allows us to design exercises with different levels of difficulty to fulfil the need of students with various backgrounds and ability levels.

To illustrate the idea, we take the programming exercise of solving a quadratic equation as a running example (Figure 2). We have created a number of test cases, which are grouped into three levels of difficulty, namely, the beginner level, intermediate level and advanced level (Figure 3). The equations which have two distinct real roots are considered relatively easier; and therefore we classify the corresponding test cases as at the beginner level (Figure 3a). The test cases which correspond to quadratic equations with one repeated root or two complex roots are classified as at the intermediate level (Figure 3b). The exceptional cases (such as those corresponding to the cases when the equations become linear or identities) are classified as at the advanced level (Figure 3c).

A Programming Exercise of Solving a Quadratic Equation

Write a program to solve a quadratic equation. The general form of a quadratic equation is $ax^2 + bx + c = 0$, where a, b, c are real numbers. When $a \neq 0$, the solution of the equation is given by the quadratic formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

In this exercise, the user need to input the value of a, b and c, then the program will output the answers accordingly. We assume that all the inputs are integers, and that the outputs are to be displayed in ascending order with 2 decimal point precision.

Fig. 2. Programming Exercise of "Quadratic Equation"

Test Cases of Quadratic Equation Programming Exercise

(a) Test Cases at the Beginner Level of Difficulty

Input	Expected Output
$a = 1, b = -5, c = 6$	$x = 3, 2$
$a = 2, b = -7, c = -15$	$x = 5, -1.5$
$a = 1, b = 6, c = 8$	$x = -2, -4$

(b) Test Cases at the Intermediate Level of Difficulty

Input	Expected Output
$a = 1, b = -2, c = 1$	$x = 1$
$a = 1, b = 2, c = 5$	$x = -1+2\mathbf{i}, -1-2\mathbf{i}$
$a = 2, b = 12, c = 18$	$x = -3$

(c) Test Cases at the Advanced Level of Difficulty

Input	Expected Output
$a = 0, b = 2, c = 4$	$x = -2$
$a = 0, b = 0, c = 0$	x = any real numbers
$a = 0, b = 0, c = -4$	No solution

Fig. 3. Test cases of quadratic equation programming exercise at different levels of difficulty

The PASS system allows us to tell the students the level of difficulty of each exercise (Figure 4). For the same problem, students can attempt the exercises based on their capability. For example, the less talented students may design a simple program to solve the problem at the beginner level of difficulty (Figure 5a). If they submit their programs to attempt exercises at other levels of difficulty, they will fail in those test cases (Figure 5b). Instead, they must enhance their programs in order to solve the problem at the intermediate level of difficulty.

Unit	Prob. No.	Problem difficulty	Prob. Type
Quadratic Equation	Beginner Level	★★	Practice
Quadratic Equation	Intermediate Level	★★★★	Practice
Quadratic Equation	Advanced Level	★★★★★★★	Practice

Fig. 4. Programming exercises at different levels of difficulty.

No.	Input	Annotation	Expected Output	Actual Output	Result
0	1 -5 6		2.00, 3.00	2.00, 3.00	Accepted
1	2 -7 -15		-1.50, 5.00	-1.50, 5.00	Accepted
2	1 6 8		-4.00, -2.00	-4.00, -2.00	Accepted

(a) A Sample Run of Submission to an Exercise at the Beginner Level of Difficulty

No.	Input	Annotation	Expected Output	Actual Output	Result
0	1 -5 6		2.00, 3.00	2.00, 3.00	Accepted
1	2 -7 -15		-1.50, 5.00	-1.50, 5.00	Accepted
2	1 6 8		-4.00, -2.00	-4.00, -2.00	Accepted
3	1 -2 1		1.00	1.00, 1.00	Wrong Answer
4	1 2 5		-1.00-2.00i, -1.00+2.00i	Runtime error 207 at $00401115 $00401115 $0040123E	Wrong Answer
5	2 12 18		-3.00	-3.00, -3.00	Wrong Answer

(b) A Sample Run of Submission to an Exercise at the Intermediate Level of Difficulty

Fig. 5. Exercises at different levels of difficulty and sample runs of submissions in PASS.

However, the talented students may work directly to solve the problem at the intermediate level of difficulty. They can even try to challenge the exceptional test cases by submitting their programs to solve the problem at the advanced level of difficulty. Eventually, they should come up with a single program which can solve the problem up to a certain level of difficulty. This approach allows the students to regulate their own learning pace. Additionally, we may require students to study the given test cases for each exercise to figure out how we select the test cases so as to learn how to test their programs on their own.

4 Incremental Style of Programming Activity

Traditional programming courses focus on the development of small applications. Without the support of related technology, students usually develop small applications by writing the code solely on their individual effort. The student may become an analyst programmer in the future and may be involved in some large scale projects. Students often find it difficult to manage large software development jobs when they work in the industry. It is very important to provide students with experiences of software development in large scale applications while they are studying.

However, there are practical difficulties to require students to develop a large application. First of all, students' learning motivation drops very fast as the time they have to spend on study increases. If we require the students to code a large application, they are usually unable to see their results before the completion of the whole application. They will lose their interests in programming soon after they started. Lack of motivation is one of the major resistances to learning (Atherton, 1999). As we foresee the need, we consider large application development as an essential part of an advanced programming course. Some special arrangements have to be made to keep the students' learning motivation.

When we design a large application, we may divide the application into several modules. After the student has completed one module of the application, he/she can submit the modules to PASS. Some stubs or test drivers can be provided for testing their individual modules. It is important to reinforce the student's success upon his/her completion of one module. This approach also increases the student's confidence in learning. The intermediate results can keep students' learning motivation constantly high. The students will develop the application in a progressive manner. After the students have completed the entire application, they can submit it to PASS, which will test all the modules together as a single integrated application.

This approach highlights the modularity of computer programs. The students are exposed to programs that are built from modules so that they learn the concept of modularity of program by implicit learning (Berry, 1997). When developing an application, students will have to divide their solutions into modules as functions and classes.

Moreover, we require the students to archive all the files developed in their activities. When designing a programming activity, we intentionally require the students to make use of some modules developed in previous activities. For example, we may require students to develop a program to solve a quadratic inequality (Figure 6) based on the module developed earlier in the programming exercise of solving a quadratic equation (Figure 3).

Similarly, as before, we create exercises at different levels of difficulty (Figure 7). The least talented students can solve the inequality by using their simple programs that solve a quadratic equation, while the talented students can try some challenging test cases such as when the quadratic inequality has one solution or no solution. In this way, students will naturally acquire the concept of code reuse through their own experience of reusing the previously developed code, as concrete experience is important in the learning cycle (Kolb, 1984).

On the other hand, we sometimes ask students to exchange files and develop their applications based on modules written by other students. This gives students some experiences how to collaborate with other students. It prepares the students to work as a team member in a large project development team in the future. Moreover, the students need to ensure that their program depends fully on the program interface, so that their programs can work properly with the modules developed with other students.

A Programming Exercise of Solving a Quadratic Inequality

Write a program to solve a quadratic inequality based on the module you developed earlier in the programming exercise of solving a quadratic equation. In general, a quadratic inequality can be written in one of the following standard forms, where *a*, *b*, *c* are real numbers:

$$ax^2 + bx + c \geq 0$$
$$ax^2 + bx + c > 0$$
$$ax^2 + bx + c \leq 0$$
$$ax^2 + bx + c < 0$$

Suppose that the equation $ax^2 + bx + c = 0$ has two real roots x_1 and x_2, where $x_1 < x_2$. If $a > 0$, the solution sets of the inequalities are, respectively, as follows.

Inequality: $ax^2 + bx + c \geq 0$. Solution: $(-\infty, x_1] \cup [x_2, +\infty)$

Inequality: $ax^2 + bx + c > 0$. Solution: $(-\infty, x_1) \cup (x_2, +\infty)$

Inequality: $ax^2 + bx + c \leq 0$. Solution: $[x_1, x_2]$

Inequality: $ax^2 + bx + c < 0$. Solution: (x_1, x_2)

Fig. 6. A programming exercise based on a previously completed module.

Unit	Prob. No.	Problem difficulty	Prob. Type
Quadratic Inequality	Beginner Level	★★★	Practice
Quadratic Inequality	Intermediate Level	★★★★★	Practice
Quadratic Inequality	Advanced Level	★★★★★★★★	Practice

Fig. 7. Programming exercises of “Quadratic Inequality”.

5 Evaluation of the New Pedagogy

The new pedagogy is first developed for teaching computer programming courses in 2006. There are a number of differences between the new pedagogy with the old one. The most major change is the design of the programming exercises. The current programming activities are designed in incremental manner with multiple levels of difficulty. However, the exercises in the old design are disjoint from each other, and there is only one level in terms of difficulty.

The PASS system is currently used during tutorial class. The teacher assigns some programming exercises to the students. The student selects the difficult level of the exercises to attempt according to their capability. Moreover, some less talent students are unable to complete the tutorial exercises during tutorial session. These students can return home to continue their works. The system will be able to provide assistance to them even after school. As the system stores some patterns of common mistake, the system will give some pre-stored hints to the students if any of the patterns is identified.

The students taking programming courses are assessed by coursework and final examinations. The coursework is usually in the format of programming assignments, and the final examination is in the format of written examinations. We have compared the results of the students before and after the implementation of new pedagogy.

We have selected a typical programming course at the introductory level (Table 1). Because the class size of this course is very large, the statistical information of this course is worthy trusted. On the other hand, the materials of assessment are moderated by peer review to ensure the standard of assessment. No scaling of score has been conducted in this course. The score boundary for each grade has been fixed by the department. As a result, this graded distribution of students is a very important indicator to show the performances of teaching and learning.

Table 1. Statistics of a Computer Programming Course

		Year 2004	Year 2005	Year 2006
Total no. of students		277	253	251
Grade	Score Boundary	% of Students	% of Students	% of Students
A	69.5	7.94%	7.11%	26.00%
B	54.5	16.25%	17.79%	22.40%
C	39.5	35.38%	23.72%	23.60%
D	34.5	9.75%	12.65%	4.80%
F	below 34.5	30.69%	38.74%	23.20%

In years 2004 and 2005, only a small percentage of students got grade “A”, while a large percentage of students failed the course in these two years (Table 1). After the new pedagogy was implemented in year 2006, the percentage of grade “A” students increased dramatically from 7~8% to 26% (Table 1). At the same time, the percentage of failure decreased significantly. As shown in the table, the students’ performances in the programming course increase significantly. This is a strong evidence to show the success of the new pedagogy.

A focus group session has been held with students who enrolled in computer programming courses. A set of interview questions are designed by professionals in education development. The students are interviewed by an independent interviewer and none of the course lecturers were presented. All the students in the focus group believe that the new pedagogy can help them to learn programming courses more effectively. Few responses are extracted as examples:

Student 1: *The programming assignment with different levels is a fresh idea. I can control my learning pace.*
Student 2: *My fellow classmates teach me a lot. They know clearly of my problem.*
Student 3: *Eventually, I can develop a computer game by myself.*
…

The preliminary results of interview show a positive feedback from the students. In order to get a more quantitative measurement for the course structure of programming courses, we have conducted a survey by questionnaires. The questionnaires are designed by professionals in education development in the similar way as (Harding, Kaczynski, & Wood, 2005). The students are asked to score each dimension of the course structure on the scale from 0 to 10, where a score of 10 represents the highest satisfaction, while 0 represents the least satisfaction. 50 students have participated in the survey. The results are summarized as Table 2.

Table 2. Evaluation of Course Design for a Computer Programming Course

Questions	**Average Score**
The PASS system is useful to my study of computer programming.	7.8
The PASS system helps me to have comprehensive testing of program.	8.2
I like the programming activity with different levels of difficulty.	8.1
I like the programming activity with incremental style.	7.3
The peer learning scheme is useful to my study.	7.8
The course design helps me to control my learning pace.	6.8
The course helps me to identify weakness.	7.5
The course encourages collaborations between students.	7.6
The course is effective in learning computer programming.	7.4

In Table 2, we can clearly see that the students are highly satisfied with the course structure. The students are happy with the flexibilities provided by the new pedagogy. They help the students to identify their weakness and control their own learning paces. Therefore, the students can learn computer programming effectively. Summing the above up, the new pedagogy is a good teaching and learning model for computer programming.

6 Conclusion

This paper has shared our experiences in design of teaching and learning activities for computer programming with large class size. By designing exercises at different levels of difficulty, we provide stepwise learning experiences to students, such that they can solve problems pertaining to their corresponding ability levels. Teachers can also define problems in various ways in PASS so as to make students familiar with modules programming and be prepared for large projects. The interviews have shown that new pedagogy is very effective in teaching and learning computer programming. The students' performances in the assessments have further confirmed our findings.

References

1. Anderson, J. R., & Skwarecki, E. (1986). The automated tutoring of introductory computer programming. Communications of the ACM, 29(9):842-849.
2. J.S. Atherton (1999). Resistance to learning: A discussion based on participants in in-service professional training programmes. Journal of Vocational Education and Training 51(1):77-90.
3. D.C. Berry (ed.) (1997). How Implicit is Implicit Learning. Oxford University Press.
4. B. du Boulay (1989), "Some difficulties of learning to program". In E. Soloway and J.C. Spohrer (eds.), Studying the novice programmer. Hillsdale, N.J.: L. Erlbaum Associates.
5. B. Cheang, A. Kurnia, A. Lim, A., and W.-C. Oon (2003), "On automated grading of programming assignments in an academic institution". Computers & Education 41(2):121-131.
6. M. Choy, U. Nazir, C.K. Poon and Y.T. Yu (2005), "Experiences in using an automated system for improving students' learning of computer programming". In Proceedings of the 4th International Conference on Web-based Learning (ICWL 2005). Lecture Notes in Computer Science (LNCS no. 3583), Springer, pp. 267-272.
7. M. Choy, S. Lam, C. K. Poon, F. L Wang, Y. T. Yu and Leo Yuen (2007), "Design and Implementation of an Automated System for Assessment of Computer Programming Assignment". In Proceedings of the 6th International Conference on Web-based Learning (ICWL 2007). Lecture Notes in Computer Science (LNCS no. 4823), Springer, pp. 584-596.
8. Harding, A., Kaczynski, D. & Wood, L.N., (2005), Evaluation of blended learning: analysis of qualitative data, Proceedings of the Symposium of Blended Learning in Science Teaching & Learning, pp. 56-62, The University of Sydney, Australia, 28th-30th September 2005.
9. D.A. Kolb (1984). Experiential Learning: Experience as the Source of Learning and Development. New Jersey: Prentice-Hall.
10. M. Schulman (2001), "Basic understandings for developing learning media for the classroom and beyond". Learning Technology Newsletter 3(1).
11. Sleeman, D., (1986), The Challenges of teaching computer programming. Communication of the ACM, 29(9):840-841.
12. Y.T. Yu, C.K. Poon and M. Choy (2006), "Experiences with PASS: Developing and using a programming assignment assessment system". In Proceedings of the 6th International Conference on Quality Software (QSIC 2006), IEEE Computer Society Press, pp. 360-365.