The Influence of Short Term Centralized Programming Training Course on Computer Science Students

Yanhong Yang, Zhongxiang Ding
School of Information Engineering
Beijing Institute of Graphic Communication
Beijing, China
Email: yangyanhong@bigc.edu.cn

Abstract-The short term centralized programming training course contains 10 cases questions that is assigned for the first year computer science students after finished their related theoretical courses. In this study, the process of the course is presented including the goal, the content and the assessment. We analyzed and showed the impact of this course on students. The results show that through this training, students can recall the knowledge that they learned and skillfully applied to the solution of the problems. The course is vital for them to future self learning and established self confidence.

Keywords: centralized programming; course design; logic training; task driven

I. INTRODUCTION

In Beijing Institute of graphic communication, every year nearly 50 students are recruited studying computer science. For computer science students the ability of programming is essential for the following courses in rest campus life [1]. Based on our experience, programming courses like Java for one semester only have 30 hours in lab, which is not enough for students master programming skills.

Without doubt, practicing programming in the lab is an important part in Computer Science Curriculum [4]. Programming training is a sophisticated process (what to write, how to read compiler messages, how to design test cases, etc) [2]. Similar work in [5] discussed the general design and implementation of this kind of course. But they didn't show the course details nor omit existing problems.

That is the reason we introduce short term centralized programming training course. In order to enhance the logic of students' program design, this course is designed to improve the logic of students' program design. This course contains 10 isolated questions for first year students before summer vacation. In this paper, we will show the goal of course and how to design this course. Furthermore, the teaching techniques and analysis are also discussed.

Through our survey, most students can improve their ability on analysis questions, design algorithm, write code and debug. When they figure out the questions, they enhance self-confidence. For teachers, it is flexible and strenuous for preparation work to choose suitable questions and make them more attractive and interesting.

II. COURSE GOAL

The goals of the centralized programming course are to improve the students' programming ability and guide them to study actively. In many cases, students have learned the basic grammar of program language, but there are still big problems in the logic of program design. When students solve practical problems, they are unable to translate their own solutions into computer programs. In the centralized programming training course, some examples of programming are added to the course. The examples enable students to learn the programming methods and improve their programming skills.

The course has taken into account the students' learning status, and has established the following five project training objectives:

1) Learn to read the code, master the correct programming learning method;

2) Learn to use various algorithms to describe languages, such as pseudocode and flowchart;

3) Learn the methods and techniques of structured programming and master Top-down design methods;

4) To improve the ability of programming, master the proficient debugging methods and good program design style;

5) Be familiar with the use of various common algorithms

III. COURSE CONTENT

Program design is a highly practical course, and the teaching content of the course is abstract. Students usually cannot understand knowledge through language and words, but through practice they can gain better understanding.

Therefore, the course uses the task-driven teaching method, which enables learners to acquire knowledge and skills in the process of accomplishing specific tasks. It is the task of combining teachers' effort that the course needs to set up a variety of difficult questions for the students to solve.
It is not easy to set up the questions for some reasons. First, the answers to those questions cannot be easily found on the Internet. If students find answers easily through Internet, they may not think about how to solve those questions on their own. The purpose of the course is difficult to reach. Secondly, the difficulty of these questions should be

reasonable. Students should be able to use the knowledge they have learned to solve those questions instead of being stumped by them. Finally, the knowledge what students learned in solving questions should be helpful for their future courses.

Based on the above considerations, Table I shows the tasks, knowledge points and difficulties assigned in the teaching of the 2018 course:

TABLE I.   COURSE CONTENTS

Number	Title	Knowledge points	Difficulty
1	Matrix rotation	Two-dimensional arrays, Matrices	1
2	Select a sort problem	Arrays, Sorting	1
3	String matching	String, Loop	2
4	Train to buy tickets	Arrays, Loops, Judgments	2
5	Count words, characters	String, ASCII code	3
6	Print a positive integer	Integer division, Number operation	3
7	Palindrome number	Circulation, Judgment	3
8	Spell check	Array, String matching	4
9	Integer merge	Arrays, Number operations	4
10	Wheel track	Simulation, Number operation	5

IV.   COURSE ASSESSMENT

In order to accurately evaluate the student's performance, the course will be graded from the following aspects：

1.   Attendance-20%

According to the students ' attendance status, they will get a corresponding score. This score reflects whether the student is seriously studying and can finish the course task on time.

2.   Code-40%

The student's code score depends on the correctness, robustness, and performance of the code. Nearly 100 test cases for each task will be used to test the program. The student's code score will be determined according to the test results.

3.   Report-40%

The content of the report should include three parts: program analysis, programming process and course summary. The program analysis is to record the students' ideas in solving the problem. The programming process includes the student's code, the interpretation of the code, etc. The course summary is used to describe the student's harvest after solving the problem.

V.   COURSE TECHNICALS

For highly practical courses, the use of centralized teaching tends to achieve better results than decentralized teaching. With the help of Course Grading teaching platform [3], students' programs can be judged by online platform. Students can quickly know the correctness of the program through platform feedback, thereby improving their learning interest and programming ability. This teaching platform not only improves the timeliness and objectivity of the course examination, but also saves the time and energy of the teachers' grade work.

This course locates in the computer room. This can achieve the combination of teaching and practice. When students have problems, they can discuss with each other. Students will learn new knowledge from other students so as to improve the teaching effect. In summary, the course needs to do the following to better improve the quality of teaching:

1)   Keep preparing carefully. The course will adopt corresponding teaching methods according to the content of the textbook and the students' learning situation. Using a heuristic method to arouse students ' inquiry consciousness Design Inquiry question, train students ' divergent thinking ability and scientific attitude, use evaluation and incentive mechanism scientifically, arouse students ' interest in study.

2)   More cases analysis and strengthening of practical requirements, focus on student programming thinking training to guide students to learn more than the knowledge of the comparison, summary, so that students grasp of the knowledge of a more solid, deeper people.

3)   The use of classroom questioning, so that students more actively to speak out their own ideas and participate in the course of learning in the classroom. Help students to get more experience from the commissioning of the machine and make it more interested in the program design.

4)   The introduction of some websites and books and other materials guides students to expand learning range.

VI.   RESULTS AND ANALYSIS

A.   Student Structure

In 2017, the number of computer science students is 48. The structure of boys and girls is illustrated in Fig.1. There is no obvious difference between boys and girls. Some female students can come up with better ideas than boys. We can even find that female students work harder than male students.

B.   Achievement Analysis

In Fig.2, the distribution of the achievement is showed that 32students got the full mark. Nearly 66.7% students reach the full mark this result is much better than what we expected. It is inevitable that there happened situation of sharing code.

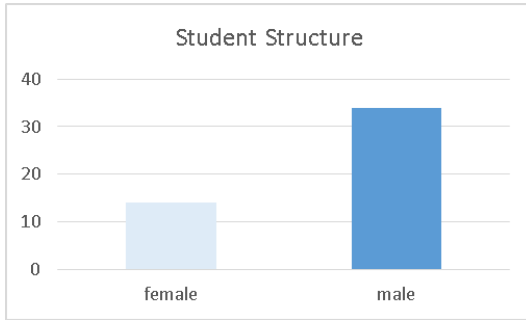We set the passing score is 80 that 83% students can gain credits of the course.
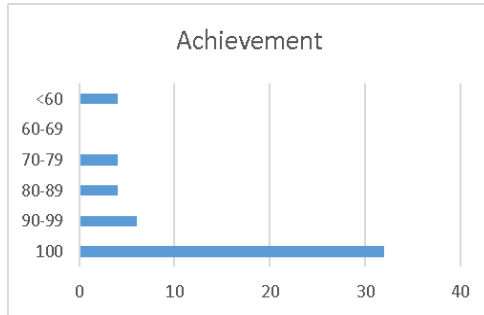
Figure 1 Student structure

Figure 2 Distribution of achievement

C. Feedback of Questionnaire

We use questionnaire to get general feedback from the students. There are 4 questions in questionnaire focus on the course not on teachers. The format of the questionnaire is showed in Fig.3.

Question 1 seventy-eight percent of the students feel helpful and thirty-two percent of the students feel not helpful. It can be easily concluded that students are also in favor of the importance of this training.

Question 2 forty-five percent of the students feel normal. Only nine percent of the students feel easy. The conclusion is that the difficulty of the subject is moderate. Of course, some excellent students find them easy, but this course is oriented toward a standardized examination.

Question 3 twenty-eight percent of the students did not like spending long time in lab on programming. This course is still a laborious course. Especially when they met difficulty, students are required to find their own ways to get the answer.

Question 4 half of them agree, and the other half disagree it. This is a difficult to evaluate the subjective feeling. But through our follow-up observation on students, the students who have completed the course seriously have made great progress.

D. Feedback of Interview

We also interviewed some students about their feedback. We collected three students' views on this course.

Questionnaire of Centralized Programming Training
1. How do you evaluate this course?
   ○Helpful ○ Not Helpful ○ Don't know
2. How do you evaluate the difficulty of the questions?
   ○ Hard ○ A little ○ Normal ○ Easy
3. Do you like the feeling of continuous days of programming?
   ○ Dislike ○ Normal ○ Like
4. Does the completion of the course help you build self-confidence?
   ○ Yes ○ No

Figure 3 Questionnaire of centralized programming training

View 1: "This course is helpful for us to learn basic algorithm and master programming ability. By constantly overcome difficulties and solve problems, we gained the way of thinking in programming and had a good understanding of programming language. But because of the lack of knowledge of algorithm, it took long time to understand."

View 2: "The first problem of climbing stairs belongs to the classic problem, it can be solved by many methods, and a lot of code can be obtained on the Internet. However, there is no current code for the second climbing stairs. Adding small conditions on the classic problem or making small changes will often be right. The idea of the whole problem has a great upgrade. If the first question does not have its own thinking and solution, the second questions have no ready-made code, so we have to start from understanding the first question and think further, this is a good call to the enthusiasm, and let the students think deeply about the classic problem, There is a more profound understanding of the methods used."

View 3: "There are some problems that are easier to do than them seems when we do it. It is difficult to convert steps or code that a computer can execute. We need to consider all the situations that the problem will encounter and deal with each situation accordingly.

Logic is the most common problem in programming, and it is also the most difficult problem to find. The logical problem of the program is often not thoughtful and careful when the code is written. It is also a troublesome thing to find the wrong place in the program.

Through the course, we have improved our programming ability and ability to solve mistakes."

VII. CONCLUSIONS

Programming ability is important for computer students. Although students learn basic knowledge in programming course, the difficulty of topic is more focused on language self. And the difficulty of subsequent data structure and algorithm will suddenly increase.

This course is an important supplement and plays a connecting role. Through our observation of the students, the students' satisfaction of this course is better, and students'

…

feedback is also rewarding. Although for teachers, the preparation of this course is a painstaking effort. Teachers need to seriously think about the topic, because the convenience of the Internet also makes it easy for students to be suspected of plagiarism.

ACKNOWLEDGMENT

REFERENCES

[1] Martin F, Ndoye A. Using Learning Analytics to Assess Student Learning in Online Courses.[J]. Journal of university teaching and learning practice, 2016, 13(3).*)*

[2] Berglund A, Eckerdal A. Learning Practice and Theory in Programming Education: Students' Lived Experience[C]. International Conference on Learning and Teaching in Computing and Engineering. IEEE, 2015:180-186.

[3] Matsuura S. Improving the marking of programming exercises using a marking support tool and subject design[C]. annual conference on computers, 2007: 76-80

[4] "Computer Science Curriculum," Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM), IEEE Computer Society2013.

[5] Yue S. Project practice and exploration to enhance programming logic in short term[J]. Chemical management, 2013(2):89-90.