Research and Exploration on CDIO-based Experimental Teaching of C Language Programming

Jiangwei Zhou[a*], Wen Li[b]

School of Freshmen, Xi'an Technological University, Xi'an, Shaanxi, China

[a]email: zhoujiangwei@xatu.edu.cn, [b]email:liwen@vip.qq.com

Abstract. Under the background of new engineering science, in order to play the role of local universities in supporting the regional economic development and industrial transformation and upgrading, the experimental teaching of C language programming faces new challenges and opportunities, and further reforms should be promoted in order to meet the needs of cultivating composite engineering talents. Aiming at the current C language programming course experimental teaching cases with low level differentiation, insufficient fun and low comprehensiveness, we designed a series of experimental cases with certain level differentiation, increased the fun of the cases, and designed a comprehensive experimental teaching case of breaking through barriers. The design concept and realization scheme of the case are studied and explored in detail. This research practice is guided by the concept of CDIO, effectively combine theoretical knowledge and programming practice, so as to improve students' practical ability and cultivate their computational thinking and innovative thinking ability.

Keywords: Programming; Experimental teaching; CDIO

1. Introduction

The goal of the new engineering reform is to cultivate composite new engineering talents with strong practical ability and innovation ability, and the curriculum teaching of local colleges and universities must adapt to the needs of regional economic and industrial development [1-3], cultivate students' ability to solve complex engineering problems. The experimental teaching is an important way to cultivate students' practical ability and innovation ability. Students' foundation is uneven [8], so teaching needs to be differentiated, but the previous experimental teaching mainly focuses on such aspects as students' mastery and consolidation of learned knowledge, less consideration of students' knowledge mastery, students' learning interest and other circumstances, and the lack of comprehensive experimental cases, resulting in failing in anticipated effects [4-6].

The paper takes the experimental teaching of C language course as an example. A series of experimental cases with certain level differentiation and a comprehensive experimental teaching case of breaking through barriers are designed based on analyzing the current problems of experimental teaching, combining the difference of students' knowledge base. The case design follows the conception, design, realization and operation concept of CDIO [7]. It is based on students' mastery of what they have learned, guides students to focus on the cultivation of computational thinking ability and innovation ability and cultivates students' ability to solve complex engineering problems through comprehensive experiments.

2. Existing Problems in Experimental Teaching of C programming.

2.1 Complex Knowledge Points and Abundant Difficult Points. C language, as the first computer programming language class course, covers data types, expressions, control structures, arrays, functions, pointers and so on. The first stage of the knowledge is relatively simple but there are many knowledge points, and the later stage of the knowledge is difficult to understand and master. Although the early knowledge points seem simple but need to be understood on the basis of a lot of practice. If not digesting in a timely manner, after the gradual accumulation of the problem, with the later knowledge points of the difficulty of a significant increase in the subsequent learning, students will have big problem with following study.

2.2 Lack of Differentiation in Experimental Cases and Comprehensive Cases. First, although there is different setting of questions based on difficulty degree, the proper questions with appropriate difficulty degree haven't been assigned to the right students. Which result in low interest and participation among some students? Second, Current experimental cases are designed based on knowledge points which can't guide students to develop the ability of analyzing the complex real engineering problems with what they have learned.

2.3 The Gap between the Teaching of Experimental Cases and Real Life. The traditional experimental case design only focuses on the integration of knowledge points, and the practical application of the combination is not enough to guide students to use what they have learned to solve the practical problems around them, and it is difficult to stimulate students' interest in learning.

2.4 The Knowledge Base Difference Among Students. The same teaching setup are assigned to students at different levels, resulting in mediocre teaching and learning.

3. The Mode of Experimental Teaching: "Emphasize the Foundation, Strengthen Practice, and Stress Innovation."

3.1 Emphasize Foundation. The basic knowledge that the course needs to master mainly includes: loop structures, arrays, functions, chain lists, files and so on. Emphasizing the basics is not to stress rote memorization, but the experimental cases are skillfully designed to organically integrate the knowledge points into the experimental cases, so that students can experience the essence of the experiments at the same time, and through repeated training students will be able to firmly grasp the knowledge points.

3.2 Strengthen Practice. Practice is throughout the course of study. Students practice in the process of study and study in the process of practice. Through the learning-practice-learning-practice of the continuous cycle of iteration, students can firmly grasp the knowledge points, and improve the hands-on skills.

3.3 Stress Innovation. Experimental cases are limited, but students' creative thinking ability is unlimited, so the design of experimental cases should not confine students' creative thinking. Experimental cases are used by teaching of C language course, but they should not be limited to the teaching content of C language course. All the correct solutions proposed by students should be accepted and should be encouraged.

4. Layered and Progressive Experimental Case Design

According to the degree of difficulty and complexity, the experimental cases are divided into four levels: basic, improvement, expansion and practical training. Each type according to the knowledge unit module is composed of breakthrough experiments, and each breakthrough experiment consists of a number of experimental cases composed of many levels. Different students can select one or more of these four types of experimental cases for training at different stages, and each experimental level can also be completed at different stages to increase interest and flexibility.

4.1 Basic Experimental Cases. In principle, the basic experimental cases are designed for all students, requiring all students to fully grasp the knowledge points they have learned. In the design of the experimental cases at this level, the main consideration is to cover the knowledge points involved in the syllabus, and an experimental case is usually designed around a single knowledge point. The experimental cases designed cover if statements, for loops, one-dimensional arrays, two-dimensional arrays, structures, chain lists, files and other programming knowledge required mastering for C language. The difficulty is controlled within the scope of majority of students' acceptance.

4.2 Improved Experimental Cases. Complexity and difficulty in improvement-type experimental cases have increased compared with basic-type experimental cases. They require students to be able to synthesize and apply a number of knowledge points to solve real-world problems on the basis of mastering what they have learned. Experiments at this level are more attractive to students with better foundation and more comprehensive mastery. Moreover, these experiments are set up as barrier experiments, the complexity and difficulty of the topic increases in

turn. The breakthrough experiments stimulate students' interest and learning enthusiasm. Successful breakthroughs allow students to enjoy the joy brought to them by the experiment, and the sense of accomplishment and honor of completing the breakthroughs further increase their desire for knowledge.

4.3 Extensive Experimental Cases. The prototypes of extended experimental topics come from some interesting computer grade examination topics and competition topics, and the difficulty is adjusted appropriately after certain processing design and treatment. The complexity and difficulty of this type of experiment is high, requiring students to have strong program design ability and programming literacy. When designing algorithms, students not only need to analyze and design the experimental cases in detail and choose appropriate algorithms, but also need to consider the quality of the algorithms, including time complexity and space complexity. These experiments are mainly for computer science majors and students who intend to participate in programming competitions, and develop their ability to use programming knowledge to solve problems with higher complexity.

4.4 Practical Experimental Cases. The practical experimental case is a comprehensive experimental case and is a simple version of the student performance management system. Student data include student number, name and the results of the three courses of advanced mathematics, college English, and C programming. It is a breakthrough experimental project, a total of 23 steps designed to include the input, statistics, analysis and output of the student performance data and other experimental content. The structure is shown in the following figure 1.

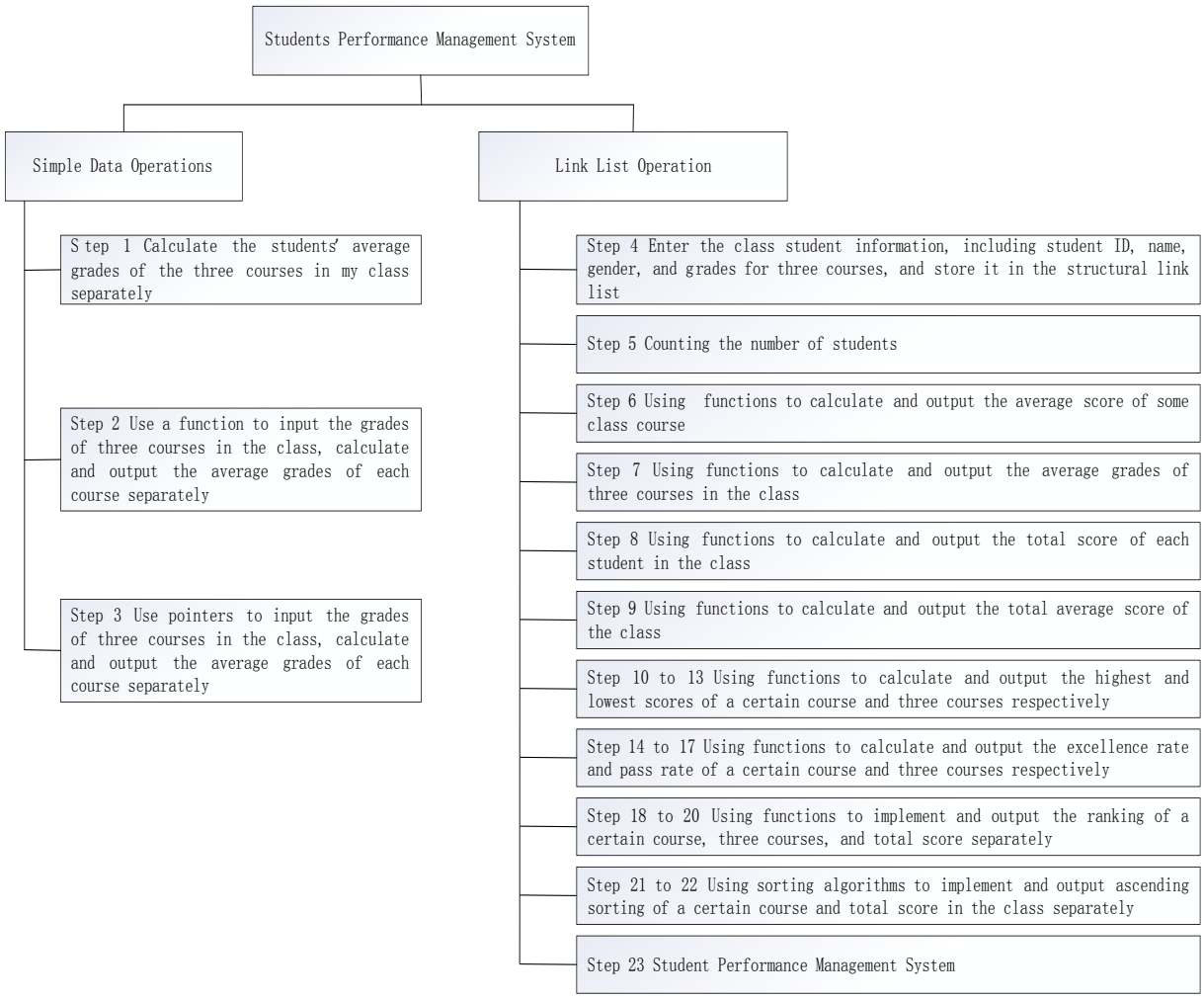

Figure 1. The structure of Practical Experimental Cases

Among them, steps 1~3 is mainly about simple manipulation of data, and step 4~23 is mainly about the operation of the chain table. Step 1 is the use of arrays; step 2 is mainly about functions and their parameter passing; step 3 is mainly about pointers as function parameters; Step 4 is about input

and storage of data, creation of the chain table; Step 5 is about traversal of the chain table; Step 6~17 is about the counting and analyzing of students' grades of the classes; Step 18~ 20 is the calculation of grade ranking; the Step 21st~22nd is the sorting of the chain table. Step 23rd is the display level, where all the program codes of the previous levels are displayed, so that the students can intuitively feel all the program codes realized in this practical training, and so that they can have a greater sense of satisfaction and acquisition.

The knowledge points examined in the case include data types, functions, chain lists, sorting and so on. Case design takes full account of the students' first contact with programming methods to solve more complex engineering problems in the actual situation. The barrier designs use the simple ones as the introduction, and gradually increase the difficulty and complexity. Each barrier combines the knowledge points also from the beginning to the high level, and gradually guides the students to learn how to decompose the complex engineering problems, modular design, so that the problem is decomposed into a number of sub-modules, sub-modules then further subdivided. Finally students form a solution to the whole engineering problem.

5. Conclusions

Based on years of teaching practice, this paper analyzes the current problems of C language experimental teaching, puts forward the experimental teaching mode of "emphasize the foundation, strengthen practice, and stress innovation", and designs a hierarchical and progressive experimental teaching case base based on the teaching concept of CDIO, as well as exploring and practicing the experimental teaching. The conception, design, realization and operation of CDIO have been generally recognized by students, and the teaching effect has improved. Students' classroom participation and concentration have been improved; students' computational thinking ability, program design ability, programming practice and hands-on ability, and students' motivation and interest in learning have been drastically improved.

Acknowledgement

This Research was supported by the Xi'an Technological University under the grant 20JGY34.

References

[1] Gu Peihua, Gu, Peihua. New Engineering and New Paradigm: Concept, Framework, and Implementation Path [J]. Higher Engineering Education Research, 2017 (6): 13 DOI: CNKI: SUN: GDGJ.0.2017-06-002.

[2] Zhong Denghua. The Connotation and Action of New Engineering Construction [J]. Research on Higher Engineering Education, 2017 (3): 6.

[3] Lu Jin, Ren Hongliang. Exploration of the Reform of Program Design Experimental Teaching System under the OBE Concept [J]. Computer Education, 2020, 000 (007): 157-161.

[4] Yuan Liyong, Zhou Jiaqing, Ni Yinghua. Exploration of Multiple Mixed Experimental Teaching Model for Programming Courses [J]. Computer Education, 2021, 000 (005): 101-104.

[5] Zheng Zhong, Wang Lei, Tan Lixiang, Bai Xuefei, Liu Yong. A Hierarchical Collaborative Experimental Teaching Model for Computer Programming [J]. Laboratory Research and Exploration, 2022, 41 (5): 188-191.

[6] Qian Yong, Liu Yao, Wei Huaixin, et al. Exploring the Ways to Cultivate the Innovative Ability of Science and Engineering College Students [J]. University Education, 2023 (3): 126-129.

[7] Gu Peihua, Shen Minfen, Li Shengping, et al. From CDIO to EIP-CDIO: Exploration of Engineering Education and Talent Cultivation Models at Shantou University [J]. Higher Engineering Education Research, 2008 (1): 9. DOI: 10.3969/j.issn.1001-4233.2008.01.003.

[8] Gao Bin, Wang Hongfeng. Research on the Teaching Reform of Computer Basic Courses for Non Computer Majors in the Context of New Engineering [J]. University Education, 2022 (12): 109-112.