

The Construction of Golden Courses of Computer Basic Courses under the Background of Emerging Engineering Education

Weigang Guo^{1, a *}, Jianqin Xie^{1, b}, Xiaohua Li^{1, c} and Qiuming Lin^{1, d}

¹ School of Electronic and Information Engineering of Foshan University, China

^awgguo@qq.com, ^b34603770@qq.com, ^c34603770@qq.com, ^dqmlin8809@qq.com

Keywords: Emerging engineering education; Basic computer courses; Golden courses; Teaching content; Teaching method

Abstract. At present, China's colleges and universities are carrying out Emerging Engineering Education construction. One of the most important contents is to strengthen the intersection and integration of computer technology and various engineering specialties. In order to cultivate students' computational thinking and information literacy, the classroom has become a very important link. This paper studies and practices the methods used in the classroom teaching process of computer basic course from three aspects. First is integrating the new information technology with the characteristics of the major to build the classroom teaching content. Second is paying attention to the operation logic of classroom cases and cultivate students' overall view of solving problems. Third is improving the classroom teaching mode, and build a pleasant, efficient and personalized classroom. And teachers encourage students to participate in classroom communication, improve students' ability to analyze and solve problems with computers. From the aspects of content, operation method and implementation technology, this paper explores a method to construct an "Golden Courses" of computer basic courses, and achieved good results.

Introduction

In 2017, the Ministry of Education launched the research on the development of "Emerging Engineering Education" and issued the "guide to the construction of Emerging Engineering Education"[1]. At present, Emerging Engineering Education has become a hot spot in the field of higher education and the main direction of education reform and development in the new era. Cross integration is one of the core characteristics of "Emerging Engineering Education", the source of innovation, and an important means to cultivate innovation ability[2][3]. The cross integration of computer technology and different specialties is one of the important forms, which plays a particularly prominent role in the construction of "Emerging Engineering Education". As a public compulsory course in Colleges and universities, the basic computer education of non computer major plays a very important role. Under this background, we should establish the computer basic teaching curriculum system for the "Emerging Engineering Education" and the ability training system corresponding to the curriculum system. In the aspect of ability training system, it focuses on the construction of teaching resources, teaching practice and teaching quality, so as to cultivate students' ability to solve problems by using computational thinking.

On the other hand, due to the differences of students' major and computer foundation in primary and secondary schools, there are great differences in students' interests and expectations for this course. Some students have not really realized the importance of Computational Thinking for their future career. This leads to their lack of time and energy, learning difficulties, and even giving up the course. Therefore, how to make students better understand and master the theoretical and logical content of computer basic courses in the classroom, and cultivate their own computing thinking, has become a problem that teachers of computer basic courses need to seriously think and explore.

As an ordinary teacher, he seldom presides over the grand reform. The classroom is the stage that every teacher can control, is directly facing the students, and is the final implementation point of the grand teaching reform. Invigorating and improving the classroom will directly help to improve the quality of teaching and the level of the whole school. Therefore, to build a "Golden Course", that is, by creating a harmonious and pleasant classroom atmosphere, to organically combine classroom teaching content with students' existing cognition and future needs, and to pursue efficient classroom and personalized classroom, is the feasible path for every teacher to participate in the education and teaching reform.

For a long time, colleges and universities have paid too much attention to the grand reform and not to the classroom. Even if they pay attention to the classroom, they seldom put forward specific implementation models for specific courses. Since 2018, the Ministry of education has emphasized the construction of "China Golden Courses", which is a first-class course[4][5]. This paper takes the classroom teaching of "University Computer Foundation" and "Programming Foundation" as an example to talk about our exploration and practice of building a "Golden Course".

Integrating New Information Technology with Specialty Characteristics and Constructing Classroom Teaching Content

The overall teaching content of the course is determined by the curriculum syllabus, and teachers mainly focus on the syllabus in class. However, there are a lot of majors in Colleges and universities. As a public compulsory course, it is impossible to make a syllabus of computer basic courses for each major. Generally, the university computer foundation curriculum is consistent. As a course teacher, it is necessary to combine the professional characteristics in the classroom and practice, so that students gradually realize the combination point and application value of computing thinking and this major, cultivate students' interest in learning, and identify with the course[6]. Here are some of our practices.

Mechanical Engineering: Integrating computer-aided design, Internet of things technology, intelligent manufacturing, robotics and other related contents to pave the way for professional learning.

Electrical engineering: Integrating computer control technology, Internet of things technology, sensor technology, automatic fault diagnosis technology and other related contents.

Civil Engineering: Integrating computer-aided design, engineering cost information system, building information modeling and other relevant contents.

Environmental Engineering: Integrating sensor technology, Internet of things technology, mobile communication technology, artificial intelligence technology and other related technologies and knowledge.

Chemical Engineering: Integrating technology and knowledge related to structural drawing software, data analysis and scientific drawing software, literature retrieval and management software, etc.

Food Engineering: Integrating big data technology, Internet of things technology, image recognition technology, food safety supervision technology under the information environment and other relevant contents.

Photoelectric Engineering: Integrating digital image processing technology, digital video processing technology, artificial intelligence technology and other related contents.

Industrial Engineering: integration of industrial Internet of things, intelligent manufacturing, industrial big data and other relevant contents.

In the process of classroom teaching, teachers should introduce the new technology to students in combination with their scientific research projects, cultivate students' quality of using information technology reasonably and correctly, mainly including data awareness, information security awareness, information ethics, consciousness of observing disciplines and laws, self reflection awareness and innovation awareness, etc.

Paying Attention to the Operation Logic of Classroom Cases and Cultivate Students' Overall View of Solving Problems

Information technology education in primary and secondary schools is mostly limited to the general application of some commonly used software. The logic of completing a work or experiment task is not clear, and the technical principles contained in it are not understood. This problem, on the one hand, requires students to experience and think and explore themselves in the process of practice. On the other hand, it requires teachers to pay more attention to it in class, so as to avoid paying too much attention to the details of operation and ignoring the understanding of the overall solution of a task[7, 8]. Next, we will take the knowledge system of style application, layer application and program design as an example to introduce our approach.

Style Application. For the long document processing in document typesetting, it focuses on the use of styles. Its core logic is as follows:

(1) For formal documents, the format of titles, text, table questions and graph questions at all levels should be consistent.

(2) In order to maintain consistency and improve efficiency, the "style" technology provided by word processing software is adopted.

(3) Creating various styles according to specifications and actual needs

(4) Applying the corresponding style to the relevant text

(5) Generating the directory and document structure according to the style, and reference the style in the header.

Layer Application. For the animation production in digital image processing, it focuses on the use of layers and frames.

(1) Different objects are placed on different layers. Do not place too many objects on one layer. The ideal situation is a theater like performance, mainly including the background, multiple actors and props involved in the performance. Background as a layer, each actor a layer, each need to move props a layer.

(2) The overlay relationship of layers, pay attention to the upper and lower order relationship of layers.

(3) Different actors appear in different frames at different time points, while the background layer is generally unchanged, and can also be changed when necessary.

Knowledge System of Programming. In the content system of the basis of programming, we should not pay too much attention to grammar knowledge, but to the basic methods of programming technology. The main problems of traditional programming courses are as follows:

(1) Too many and too detailed sentences and grammar.

(2) Ignore the solving process of basic problems.

- (3) The algorithm consciousness in the course is weak.
- (4) Can't arouse students' interest in learning.
- (5) There are problems in teaching objectives. I can't use it. I forget it quickly.

The content is too much and too difficult, which makes the students only understand some sporadic knowledge points after finishing the program design course, and the overall concept of program design is seriously lacking.

Therefore, for the program design course, we should fully consider the characteristics of students and the needs of future learning and work, and adopt the strategy of "taking something, not taking something" for the program design language itself. For those common language components, they are introduced in detail and accurately in the process of programming. For those that are not closely related to the programming method and are not very important, it is a brief introduction when the program is exemplified. For those parts that are not related to the method of program design and are not commonly used, they will not be explained and students will learn independently after class. In addition, for many kinds of expression methods of the same knowledge point (such as the initialization of array, the initialization of structure variable, etc.), only one of the most commonly used methods is selected for detailed introduction, and students are required to be the preferred method for their own programming. For other methods, as long as they are understood, they do not need to be excessively entangled in such details.

Practice Teaching. It needs practical training to cultivate students' creative ability. The practical teaching content of the basic computer course designed by us can be divided into three levels in general. The first level is based on the theoretical teaching content, which mainly allows students to complete the relevant practical content according to the requirements and steps in the experimental instruction book, so-called "draw gourd according to the sample". This is very necessary for beginners. The goal is to "do it while learning". The second level is that on the basis of higher than theoretical teaching, students complete digital works independently, with the goal of "learning by doing". The third level is to integrate with the students' major and carry out engineering practice projects combining the characteristics of the major, with the goal of "thinking while doing". The perfect combination of these three levels can promote students to use computational thinking, solve problems in thinking, verify problems, innovate and test thinking, and obtain insights[9].

Improving the Classroom Teaching Mode and Build a Pleasant, Efficient and Personalized Classroom

Classroom is the lifeline of the whole teaching process. An effective classroom can make teaching half successful. We mainly explore and practice from the aspects of gradually showing the programming process in the multimedia environment and paying attention to students' participation in classroom teaching.

Programming Process Presentation Based on Multimedia Environment. One of the disadvantages of using multimedia classroom in class is that it is easy for teachers to "read the textbook" for PPT courseware. For this course, the process of algorithm design and programming cannot be displayed. The teacher just explains the existing algorithm and program, and then demonstrates the running results of the program in the programming environment. Because students have not experienced a process of derivation and compilation from scratch, this teaching method is very unfavorable to cultivate students' ability to analyze and solve problems.

Instead of showing the complete algorithm and program directly to students in class, we first use the process of "ask questions-->analyze problems-->propose preliminary algorithms-->refine algorithms" to give solutions to problems. Then teachers make full use of the advantages of multimedia teaching environment, write programs step by step in the real programming environment, provide students with a real "from scratch, from simple to complex" programming experience, and fully give students time and space to think.

Let's take the bubble sorting of 10 numbers (from small to large) as an example to describe the method adopted in our class.

The first step is to write out the basic framework of the program, including the input and output of numerical values, and leave the sorting part blank. The program can be run. Students can see that the current order of the 10 numbers of input and output is the same (See Fig. 1).

```
#include <stdio.h>
int main()
{
    int i,a[10];
    //Input 10 integers
    printf("Please Input 10 Integers:\n");
    for (i=0;i<=9;i++)
        scanf("%d",&a[i]);
    // Processing: Bubble Sort

    //Output 10 sorted integers
    printf("The Sorted 10 Integers:\n");
    for (i=0;i<=9;i++)
        printf(" %d ",a[i]);
    printf("\n");
    return 0;
```

```
}
```

```
Please Input 10 Integers:
1 3 5 7 9 8 6 4 2 0
The Sorted 10 Integers:
1 3 5 7 9 8 6 4 2 0
Press any key to continue
```

Figure 1. Results of the first step

Step 2: in the sorting part, write a circular statement to compare the 10 numbers in turn. When the number in front is greater than the number in the back, exchange them. After the comparison, output all the 10 numbers. Then students can see the results of the first bubble sorting (See Fig. 2). The code is as follows:

```
// Processing: Bubble Sort
//First round
for (i=0;i<=8;i++)
    if(a[i]>a[i+1]) {temp=a[i+1];a[i+1]=a[i];a[i]=temp;}
```

```
Please Input 10 Integers:
1 3 5 7 9 8 6 4 2 0
The Sorted 10 Integers:
1 3 5 7 8 6 4 2 0 9
Press any key to continue
```

Figure 2. Operation results of the first round of bubble sorting

Step 3: keep the first sorting code written in step 2 unchanged, write another cycle to compare the remaining 9 numbers in turn. When the number in front is greater than the number in the back, exchange them, and output all the 10 numbers after the comparison. At this time, students can see the result of two bubble sorting(See Fig. 3). In the same way, write a separate piece of code for the third round of bubble sorting, and output the results(See Fig. 4).

```
//Second round
for (i=0;i<=7;i++)
    if(a[i]>a[i+1]) {temp=a[i+1];a[i+1]=a[i];a[i]=temp;}
```

```
Please Input 10 Integers:
1 3 5 7 9 8 6 4 2 0
The Sorted 10 Integers:
1 3 5 7 6 4 2 0 8 9
Press any key to continue
```

Figure 3. Operation results of the second round of bubble sorting

```
//Third round
for (i=0;i<=6;i++)
    if(a[i]>a[i+1]) {temp=a[i+1];a[i+1]=a[i];a[i]=temp;}
```

```
Please Input 10 Integers:
1 3 5 7 9 8 6 4 2 0
The Sorted 10 Integers:
1 3 5 6 4 2 0 7 8 9
Press any key to continue
```

Figure 4. Operation results of the third round of bubble sorting

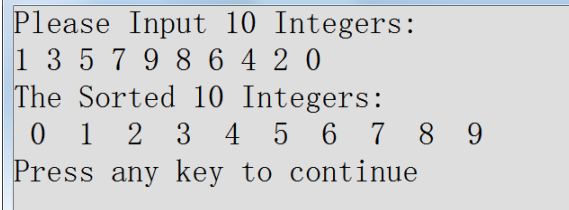
Step 4: after the first two steps, students basically understand the bubble sorting process, and know that 10 numbers can be sorted by using the same 9 cycles. Therefore, we propose to use a two-level loop to describe the repetitive and regular instructions, so as to complete the whole sorting process. The final sorting result is shown in Fig. 5.

```
//Two level circulation
for (j=0;j<=8;j++)
```

```

for (i=0;i<=9-j;i++)
    if(a[i]>a[i+1]) {temp=a[i+1];a[i+1]=a[i];a[i]=temp;}

```



```

Please Input 10 Integers:
1 3 5 7 9 8 6 4 2 0
The Sorted 10 Integers:
0 1 2 3 4 5 6 7 8 9
Press any key to continue

```

Figure 5. Final result of bubble sorting

Step 5: delete or comment the auxiliary statements in the program, and a complete program will be formed. The code is as follows.

```

#include <stdio.h>
int main()
{
    int i,j,temp,a[10];
//Input 10 integers
    printf("Please Input 10 Integers:\n");
    for (i=0;i<=9;i++)
        scanf("%d",&a[i]);
// Processing: Bubble Sort
    for (j=0;j<=8;j++)
        for (i=0;i<=9-j;i++)
            if(a[i]>a[i+1]) {temp=a[i+1];a[i+1]=a[i];a[i]=temp;}
//Output 10 sorted integers
    printf("The Sorted 10 Integers:\n");
    for (i=0;i<=9;i++)
        printf(" %d ",a[i]);
    printf("\n");
    return 0;
}

```

Through this method, the vast majority of students can follow the teacher's explanation ideas, understand the analysis and solution process of a specific problem, and help students to analyze problems and write programs independently after class.

Student Engagement and Collaboration. In the multimedia classroom, the amount of information in the classroom is generally large, continuous and high-intensity listening will lead to the decline of students' attention. Therefore, we often set aside a certain time to communicate with students in class, so that students can participate in the teaching process. The common ways are as follows:

(1) Program restore. After the teacher explained an example, he deleted the main algorithm part of the program and asked the students to rewrite it on the stage.

(2) Program error correction. The teacher gave a wrong program in the student's homework and asked the students to modify it.

(3) Program modification. The teacher first gives an algorithm, and asks the students to use another algorithm.

(4) Program imitation. The teacher gives the solution of a problem and asks the students to solve a similar problem.

In the process of students participating in teaching on the stage, other students are also required to complete the corresponding requirements together in the seats. When the students on the stage can't finish it by themselves, he can also invite the students in the dormitory to discuss with them on the stage, so as to solve the problem.

This mode of students' participation in classroom teaching has at least the following advantages:

(1) The initiative of students has increased, because they want to ask questions, so they must listen to the class carefully.

(2) Giving students time to think. With the development of multimedia teaching, students can hardly stop to think. So this participation is actually the time for students to think about problems.

(3) Contributing to the atmosphere of mutual cooperation and discussion. In the process of teaching, we always hope that students should discuss more about programming. Only by discussing more can we broaden our thinking. The mutual "help" between students in the same dormitory helps to form this kind of discussion atmosphere.

(4) It helps to activate the classroom atmosphere. It's hard to hear laughter in the classroom of University Science and engineering courses. This part of us ignited the enthusiasm of students and made the classroom full of vitality.

Conclusions

In September 2015, we began to conduct experiments in mechanical engineering, electrical automation, civil engineering, environmental engineering and other engineering disciplines and some science disciplines of the University, and achieved good results. The students reflected well in the final questionnaire. The typical evaluation of students is: "In class, the teacher is humorous and funny, with vivid examples. Language C is more understood by his own logic. Before each class, I will preview the content of the class. When I listen carefully in class, I feel that it's very cool to listen to the class, and I get different results every time. When I encounter problems, I can solve them in time. I find that the learning effect is very good." Students have scored 96.74 and 95.96 points (full score is 100) in the assessment of classroom teaching quality, which has been highly recognized by students.

Acknowledgements

This work is completed under the support of the Higher Education Teaching Reform Project of Guangdong Provincial Undergraduate Colleges in 2017, "Computer Basic Course Reform under the Background of High Level University of Science and Engineering and Emerging Engineering Education".

References

- [1] D. Zhong, Connotations and Actions for Establishing the Emerging Engineering Education. Research on Higher Engineering Education, Vol.23(2017)No.3,1-7.
- [2] Q. He, H. Wang. University Computer Basic Course System and Course Construction for the Emerging Engineering Education. University Teaching in China, (2019), No.1, 39-43.
- [3] L. Yao, Y. Song, Z. Shi. The thinking and Exploration of University Computer Basic Course System Based on the Emerging Engineering Education. Computer Education, (2019), No.3, 112-116.
- [4] J. Ma, Z. Luo. A talk on University "Golden Courses" from the Perspective of Students. Jiangsu Higher Education, (2019), No.5, 60-66.
- [5] Y. Wu. Building China's "Golden Courses". University Teaching in China, (2018), No.12, 4-9.
- [6] Y. Wang, H. Yin, F. Li. The Teaching Reform of the Basic Computer Education in the University Computational Thinking Oriented. Computer Education, (2017) No.8, 74-77.
- [7] Z. Jin. Research on the Evaluation of Classroom Teaching of Computer Basic Courses in University. China Computer & Communication, (2019), No.16, 252-254.
- [8] X. Shang. Research and Practice on Teaching Reform of Programming Language Course. The Theory and Practice of Innovation and Entrepreneurship, (2019), No.4(b), 85-86.
- [9] X. Zhou. Exploration on Teaching Practice and Reform of the Course of "Computer Foundation". Education Teaching Forum, (2019) No.36, 88-89.