Teaching reform of the "data structures" course for computer majors in independent colleges

Xu Xiaogang[1, a*]; Eliza beth[1, b]; Liu Jun[2, c]

[1]Oxford University Press Great Clarendon Street Oxford Oxfordshire UK OX2 6DP.
[2]Northwestern Polytechnical University youyi West Road Beilin District Xi'an City Shaanxi Province China 710072.
a.li_yong_1@sohu.com;b.558569133@yahoo.com;c. chen_0223haoran@126.com
***Corresponding Author**

Abstract: Based on the training objectives for applied talents, this paper analyzes the current situation and existing problems of students majoring in computer science in independent colleges and the teaching of the "Data Structures" course. It also studies the teaching reform of the "Data Structures" course from aspects such as teaching content, methods, means, training of applied teachers, and strengthening the learning of prerequisite courses. Practice has proved that the teaching reform plan can improve students' practical application ability and the quality of talent training.

Keywords: Independent Colleges; Double-Qualified Teachers; Data Structures;

1. Characteristics and current situation of computer major students in independent colleges

1.1. Low professional quality

Due to the relatively low admission scores, most students in independent colleges have a significant gap in professional academic performance compared with those in second-tier undergraduate colleges. They often struggle with professional courses, especially highly specialized ones. After completing a course, students have an incomplete understanding, inadequate comprehension, unsteady mastery of the knowledge learned, and unskilled skills.

1.2. Uneven foundations

The most important prerequisite course for "Data Structures" is "Programming Language", such as C or C++. This is the first programming language learned by students. Especially when starting with the object-oriented C++ language, students with weak abstract thinking ability may encounter great difficulties in understanding relevant concepts. In addition, due to students' lack of adaptation to teaching methods and the limited class hours of the course, teachers can only explain basic grammatical knowledge and simple programming knowledge, without sufficient time to elaborate on complex data types such as pointers and structures taught later. As a result, students' insufficient and inadequate mastery of programming language knowledge affects their learning of subsequent courses such as "Data Structures" [3].

1.3. Weak application and development capabilities

The "Data Structures" course is generally offered in the first semester of the sophomore year. Students have just completed the study of C or C++ and have no experience in project development training, so their programming ability is relatively weak. Some students even cannot successfully complete very simple programming tasks, such as finding the maximum number among N numbers and simple sorting. Overall, students' application and development capabilities are very poor, which inevitably affects the teaching of course design in "Data Structures". Students are unable to complete or fail to complete the design projects assigned by teachers well, seriously affecting the teaching effect.

https://doi.org/10.62852/csa/2026/192

1.4. Insufficient recognition of the importance of the "data structures" course

A major difference between computer majors and other majors is that computer majors emphasize practical operation ability very much. For computer major students, programming ability is the most basic professional skill. Only by independently completing code writing and debugging can they truly master programming languages and improve their problem-solving ability, programming ability, debugging and testing ability. The "Data Structures" course is not only a theoretical study but also a combination of theory and application. This course precisely provides a good programming practice opportunity to deepen the understanding, application and mastery of the previously learned programming language knowledge, and it is also a foundational course for large-scale project development in the next step. If students do not recognize its importance and do not devote themselves to learning the "Data Structures" course and conducting programming practice training, it will be difficult for them to learn the course well [4].

1.5. Tendency to relax requirements on themselves

After the intensive study of the college entrance examination, many students have the idea of relaxing themselves thoroughly when entering university. Due to the convenience of using computers, computer major students are more likely to indulge in online games, staying up all night playing games and sleeping during the day instead of attending classes. If teachers do not pay attention to and attach importance to this problem and strengthen management at this time, such students will definitely fail to learn well and even face the risk of failing to complete their studies.

2. Characteristics of the "data structures" course

2.1. Abstract theories, wide coverage of knowledge, and strong practicality

From the four basic logical structures to the four basic storage structures, in addition to classic chapters such as linear tables, stacks, queues, trees, and graphs, it also includes external storage, search, sorting, etc. Different textbooks have different degrees of content expansion. Many algorithms in the "Data Structures" course need to be implemented through programming to be understood, so it has strong practicality [5].

2.2. Difficulties in understanding and designing some algorithms

In the "Data Structures" course, many students need to spend a lot of time to understand the algorithm design of many problems, such as recursive algorithms. When doing course design, it is necessary to use other application-related knowledge and skills, such as software development content. Students who are newly exposed to it will feel it is quite difficult.

2.3. Short teaching time and heavy tasks

To complete the teaching plan, teachers have to accelerate the teaching progress, so they cannot focus on explaining relevant knowledge, leading to students' inability to keep up with the teaching progress, which affects their mastery of knowledge and learning enthusiasm.

3. Research on the teaching reform of the "data structures" course

3.1. Solving students' ideological problems

Based on the author's years of teaching experience, most computer major students in independent colleges have the problem of insufficient self-confidence. Although their foundations are relatively weak, it is completely possible for them to learn professional courses well through their own efforts and teachers' guidance. Therefore, the first problem that teachers need to solve is to let students overcome their inferiority complex and devote themselves wholeheartedly to course learning. Of course, there are many methods to do such ideological work. For example, let students start with simple problems to gain self-confidence; then gradually increase the difficulty, understand students' actual difficulties, and provide guidance to continuously enhance students' sense of achievement and learning enthusiasm.

https://doi.org/10.62852/csa/2026/192

3.2. Reforming teaching content

The "Data Structures" course has a lot of content and complex data type descriptions. Some complex problems have a lot of code, which makes students feel frustrated at first sight. Therefore, according to the characteristics of students in independent colleges, teachers can appropriately adjust the teaching content and focus on explaining linear tables, stacks and queues, trees, graphs, search, sorting, recursion and other contents.

Strengthen the teaching of object-oriented "Data Structures" course. Many "Data Structures" textbooks describe algorithms using C language, while object-oriented methods are only mentioned in a small part at the end of the textbooks. If students take C++ programming language, teachers can consider choosing "Data Structures" textbooks described in C++. In this way, the implementation of algorithms is closer to project development practice, and students can further understand and master the relevant concepts of object-oriented programming. Of course, for computer major students in independent colleges, the difficulty can be appropriately reduced.

3.3. Improving teaching methods

For students in independent colleges, teaching methods need to be changed to a certain extent. The author believes that improvements should be made in the following aspects: (1) The examples explained initially should be as simple as possible so that students can keep up with the teacher's thinking.(2) For the programming of algorithm implementation, teachers should preferably demonstrate and run the program in class to obtain correct results. (3) First of all, ensure that students understand the algorithm; secondly, let students think more, practice more, write more, and use their brains more in class, increase students' participation in the teaching process, and even let students act as teachers to explain a certain knowledge point in class. (4) In practical computer classes, practical tasks must be assigned to guide students to implement algorithms through programming on the computer. (5) Pay attention to stimulating students' learning interest. Some students may have the idea of giving up after studying for a period of time because they feel the difficulty is increasing. At this time, teachers need to timely grasp the students' learning situation, adjust the teaching progress, do a good job in students' ideological work, and encourage students to persist in learning.

4. Improving teaching means

First, it is best for students to take theoretical courses and practical computer courses in computer labs, so that students can follow the teacher to practice, achieving better learning results. At the same time, teachers can let students implement algorithms through programming in class in a timely manner, and understand and tutor students' learning in a timely manner.

Second, for relatively abstract teaching content, some animation courseware can be introduced to enhance students' intuitive sense. Especially for content related to pointers, such as adding animation teaching, students can better understand the changes of pointers.

Third, online auxiliary teaching. The development of the current network has greatly facilitated the search for information. There are also many specialized learning websites on the network, as well as high-quality teaching courseware and teaching videos. These resources are of great help to students' learning, especially for some difficult-to-understand knowledge points, students can find satisfactory answers.

4.1. Strengthening the teaching of programming languages (C or C++) to improve students' programming ability

At present, most computer majors in colleges and universities offer C or C++ programming languages. Due to the limitation of class hours, a considerable part of the time is spent explaining basic grammar and statements to students, while there is little teaching practice for more important contents such as pointers, structures, and object-oriented programming. As a result, students do not master these difficult-to-learn contents well, and even do not understand some concepts.

"Programming Language" is a prerequisite course for learning the "Data Structures" course. Therefore, it is necessary to strengthen the teaching of language courses, especially the improvement of students'

https://doi.org/10.62852/csa/2026/192

programming ability. The author believes that for language courses, first of all, a one-year study period must be arranged: students learn the basic part in the first semester and object-oriented content in the second semester; secondly, strengthen the training of students' practical programming ability on the computer, and improve students' ability through teaching methods such as project-based, task-based, problem-based, heuristic, and comparative teaching; thirdly, strengthen the construction of daily assessment mechanisms, establish an assessment and evaluation mechanism for students' learning effects to better reflect students' learning situation; finally, supplement and improve according to students' deficiencies.

4.2. Emphasizing the course design link

Course design is an important practical link in the teaching of the "Data Structures" course. For students in independent colleges, course design projects can be arranged in an order from easy to difficult. Course design can be arranged twice: once in the first half of the semester and once in the second half. The first half mainly focuses on the application design of linked lists and stacks, and the second half can focus on the application design of trees and graphs. In addition, some students' practical ability is indeed not strong. To achieve the teaching effect, a hierarchical teaching model must be adopted. For students at different levels, course design projects of different difficulties can be assigned, and each project is marked with a difficulty coefficient. Projects with a higher difficulty coefficient will get higher scores after completion. At the same time, it is necessary to strengthen assessment management. For the course design projects carried out by students, defenses and tests should be conducted to ensure that they are completed by the students themselves rather than plagiarizing others' results.

4.3. Cultivating practical and applied teachers

To achieve good teaching effects in the "Data Structures" course, the quality of teachers is quite important, especially their practical teaching ability. University teachers generally have a good theoretical foundation but weak practical application ability. Many students always fail to understand the importance of this course in programming when learning the "Data Structures" course. Students only learn what teachers ask them to learn, and their independent learning ability is poor. If the relevant knowledge points of the "Data Structures" course can be interspersed in the teaching of actual project development, students' theoretical understanding will be greatly deepened. Of course, to achieve this effect, it is inseparable from teachers' strong practical skills. At present, computer technology is changing rapidly, and the teaching content in colleges and universities is often slower than the latest technologies. University teachers must constantly keep up with the pace of technological development to meet the changing social needs. For this reason, schools need to strengthen teachers' abilities and regularly send teachers to companies and enterprises for further study, so that teachers can become "double-qualified teachers".

4.4. Encouraging students to participate in more computer academic competitions

Encourage computer major students to actively participate in competitions to deepen their understanding of the knowledge of the "Data Structures" course. When students participate in computer academic competitions, they will inevitably use knowledge related to data structures, such as application software competitions, ACM Programming Contests, network application software APP competitions, robot competitions, etc. These competitions are the best touchstones to test students' learning effects. In the process of participating in competitions, students' understanding of the knowledge of the "Data Structures" course will also be greatly improved.

5. Conclusion

The importance of the "Data Structures" course for computer major students is self-evident. It can directly affect students' programming and development capabilities. Although the "Data Structures" course is relatively difficult to learn, it can be mastered through efforts. Only when teachers continuously explore, summarize and improve teaching concepts and methods in teaching practice, and form a set of teaching models suitable for students in independent colleges, can they improve students' logical thinking ability

https://doi.org/10.62852/csa/2026/192

and practical application ability, better complete the teaching of the "Data Structures" course, achieve the teaching objectives of the course, and obtain satisfactory teaching effects.

6. References

[1] Li Chunbao. Data Structure Tutorial[M]. Beijing: Tsinghua University Press, 2013.

[2] Yan Weifan, Wu Weimin. Data Structures (C Language Edition) [M]. Beijing: Tsinghua University Press, 2008.

[3] Wang Ning. Teaching Reform and Practice of Data Structures Course Based on Applied Talent Training in Independent Colleges[J]. Journal of Kaifeng Institute of Education, 2016, 36(5):147-148.

[4] Liu Zhi, Zhao Mingzhu, Liu Duanyang. Research on Teaching Methods of Data Structures Course Oriented to Programming Ability Training[J]. Computer Education, 2015(8):66-69.

[5] Qin Yuping. Research and Practice on Teaching Methods of Data Structures Course[J]. Journal of Bohai University (Natural Science Edition), 2012, 33(4):350-352.

https://doi.org/10.62852/csa/2026/192