***I.J. Education and Management Engineering*, 2012, 11, 18-23**
Published Online November 2012 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2012.11.04

Available online at http://www.mecs-press.net/ijeme

Research of Overall Optimization Based on a Series of Data Structure Courses Teaching Content

Dongmei Li, Wenfei Zhu, Zhibo Chen, Chunling Wang

School of Information and Technology Beijing Forestry University Beijing, China

Abstract

“Data Structure” is the core of computer science courses, which is closely related to “Programming Language”, “Discrete Mathematics”, “Operating System” and “Compiler Principle”. The five courses have a strong relationship between academic edge in the teaching content, teaching model and teaching method. This paper proposes a method of overall optimization based on a series of data structure courses teaching content. The method explains inherent relationships among the courses so as to avoid to be disconnected each other. As a result, general efficiency of the five courses teaching is increased and the students’ ability of the data abstraction and algorithm design is improved.

Index Terms: Data structure；a series of courses；teaching reform；overall optimization

1. Introduction

“Data Structure” is the core of computer science courses. After study this course, students can analyze the characteristic of computer processing object, then choose the right storage representation and realize the corresponding algorithm. Their capacity of the data abstraction and designing algorithms is improved.

According to the “CC2005” [1, 2] published by the Joint ACM/IEEE-CS Task Force, the curriculum system of computer consists of 15 courses. Course relationship is illustrated in Figure 1 (omitting the course “Social and Professional Ethics”). In accordance with the courses’ learning order, “Data Structure” ranks the forth. Four courses are closely related with “Data Structure”, including the prerequisites “Programming Language” and “Discrete Mathematics”, the successors “Compiler Principle” and “Operating System”. These five courses constitute a strongly correlated chain. “Programming Language” and “Discrete Mathematics” are the basis of algorithm realization in “Data Structure”, while many algorithms in “Data Structure” appear in “Operating System” and “Compiler Principle”.

Previously, the courses on the chain have not been associated systematically and closely. Students, who have learned “Programming Language”, are unable to program to realize data structures and algorithms in the Data Structure [3]. Besides, students have no ability to apply the knowledge and skills in “Data Structure” to

Foundation project: Education reform project of Beijing Forestry University in 2008

* Corresponding author.
E-mail address:

"Operating System" and "Compiler Principle". This learning effect has a negative impact on the cultivation of students' capacity of data abstraction and algorithm design.

This paper presents a method of overall optimization based on a series of Data Structure courses teaching content. The internal relationship among the five courses is explained so as to assure them truly correlated and construct a scientific, systematic and advanced curriculum structure [4, 5].

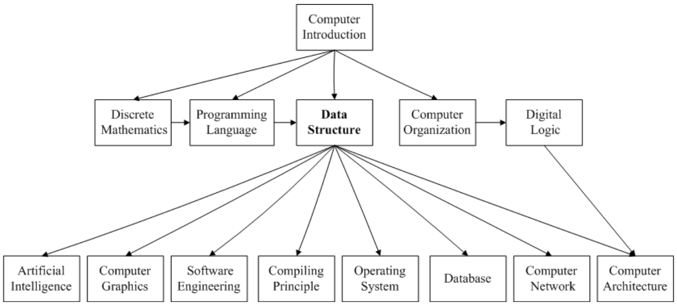

Figure 1. Computer science courses structure

2. Construct Series Courses and Emphasize Knowledge Integration

A. The Coherence Between "Programming Language" and "Data Structure"

According to the requirement of the core course "Data Structure", the existed teaching schemes of "Programming Language" should be reformed and the knowledge related "Data Structure" should be emphasized.

1) Theoretical Teaching Scheme Reform: In the theoretical teaching of "Programming Language", teachers should stress on explaining the knowledge used in "Data Structure" by giving more examples, mainly including:

a) Pointer and array: The difference between static allocation of array and dynamic allocation of pointer array.

b) Structure: Definition and useage of structure, pointer of pointing to a structure.

c) Function: Meaning and useage of function parameter by value and addresse. The difference between pointer and reference as function parameter.

d) Template: Template class and template function.

e) File: Operation of reading and writing in a text file.

2) Practical teaching scheme

a) Learning to trace and debug program: Students

analyze common types of compile errors and learn and debug program, which lay a foundation to debug more complicated program in the subsequent courses.

b) Adding comprehensive experiment: A comprehensive experiment can be added at least, which consists of important contents in "Data Structure"[6].

B. Applications of "Discrete Mathematics" in "Data Structure"

The major research contents of "Data Structure" include data logical structures, storage structures and basic operations. The data logical structures and basic operations come from discrete structures and algorithms in "Discrete Mathematics" [7].The logical structure of data is divided into four categories: set, linear structure, tree structure and graph structure, which are the main research content in “Discrete Mathematics”. Strengthens the following contents in “Discrete Mathematics” teaching:

1) Discrete Structure

a) Set composed of elements: Elements can be interpreted as the data elements, records, nodes and vertexes in “Data Structure”.

b) Some relationships existed between data elements: For instance, the one-to-one relationship between students in student form, the one-to-many relationship between ancestors and descendants in the family tree and the one-to-many relationship between vertices in campus graph.

2) Algorithm

a) Tree-related algorithms: Traversal algorithms, Huffman tree and Huffman encoding.

b) Graph-related algorithms: Königsberg Seven Bridges problem, minimum spanning tree and the shortest path problem.

C. Case-Based Teaching in “Data Structure ”

According to the corresponding cases and preliminary knowledge of the logical structures, teachers explain cases and guide students to understand them. Students study the knowledge of “Data Structures” with cases and resolve cases. Practice teaching is also closely related to cases. Students propose solutions by learning and discussing basic knowledge about these cases in class**,** and then solve them through practice [8, 9].

1) Case Selection: To choose effective cases, teachers should pay attention to the two aspects. One is to choose cases based on the teaching content, which means the cases should involve knowledge of what students have learned or will learn. The other is to choose cases with a suitable level for students. Good cases not only can strengthen students in grasping basic concepts, knowledge and skills, but help students to get familiar with the relevant knowledge and improve their ability in analyzing and solving problems.

2) Case Introduction: Before explaining a new data structure or the corresponding algorithm, the teacher should focus on clarifying principles, showing the analysis process and drawing the conclusion. For example, take washing dishes as a teaching case to introduce the stack structure. A pile of dirty bowls is putted on the left and another clean pile is putted on the right. A dishwasher keeps taking the dirty bowl from the stack in his left hand. After cleaning it, he puts it back to the stack on his right hand. When he takes a bowl from the stack, he always removes the top one on the pile (Pop Operation) and put the clean one back into stack on the top of the pile (Push Operation). He will take clean ones on the top for later use. This operation precisely confirms that the stack structure is one of the linear lists with special property of "first in last out".

3) Case Design: Cases for class discussion can involve relatively simple ones, which mainly reflect the integration of knowledge before and after. As students have learned basic concepts of linear list, storage mode and algorithms realization, etc., teachers can raise some questions as follows.

- Stack is a kind of linear list and the linear list can be stored in chain storage or sequence storage .Can the stack structure be stored in the same way?
- How do we choose chain storage or sequence storage?
- What do operations of the stack have? Do these operations relate to the storage modes?

Teachers can comment on what students are discussing and give one or more referenced answers. Sometimes teachers don’t even give answers, but let students consult some references after class and discuss each other again next class. Thereby, students’ interest in inquiry learning is stimulated.

4) Case Implementation: Generally, practice teaching is an integration of some contents. According to the specific practice task discussed in class, teachers require students to complete it by themselves. For instance, students can use the stack to implement the evaluation of the expression by simulating compiler.

D. Use Computation Modules to Deepen Students' Understanding of Correlative Algorithms in "Programming Language" and "Data Structure"

Both of "Operating System" and "Compiler Principle" use algorithms of linear list, stack, queue, search and sort in "Data Structure". Case introduction in "Data Structure" has laid the basis for the further study. In the two following courses, with cases realization through teaching and practicing, the corresponding computation modules help to deepen students' understanding of correlative algorithms in "Programming Language" and "Data Structure".

1) Applications of "Data Structure" in "Operating System"

a) Vector and matrix: The bankers' algorithm, which has always been used for avoiding deadlock, needs to define several vectors and matrices to store the corresponding resource, resource request and resource allocation, etc.

b) List: There are four management functions in "Operating System", including memory management, equipment management, processor management and document management. All of them use the knowledge of "Data Structure" extensively. For instance, in dynamic memory partition, free partition table and leisure partition chain are two kinds of data structures, both of which are repeatedly used in partition allocation algorithms, including the first fit algorithm, the next fit algorithm and the best fit algorithm.

c) Stack: One of the most popular page replacement algorithms in virtual memory is the most long unused replacement algorithm (LRU), which uses a special stack to save the current used page number of each page .

d) Queue: Buffers in a buffer pool can be shared among multiple processes. And the buffer pool brings the same type of buffer chains into a queue.

2) Applications of "Data Structure" in "Compiler Principle"

a) Stack: All of the common syntax analysis algorithms in "Compiler Principle"(such as operator precedence analysis, LL, LR analysis) and runtime storage organization and distribution solutions have used the stack structure.

b) Linear list, hash table: To manage the symbol table, comliler can use a general linear list, or adopt a more efficient hash table [11][12].

c) Tree: Many algorithms of semantic analysis and codes generation are based on the tree structure.

3. Well-Designed Practice Teaching Content for Practical Problems

Faced with practical and typical problems, teachers should make further adjustment of the content and methods in current practice sessions and carefully designed practice content. According to the course content, design the appropriate experimental guidance material and give the corresponding experimental examples, so that enable students to develop practical ability gradually and strengthen their understanding of theoretical knowledge.

- In the comprehensive experiments of "Programming Language", teachers can design tasks which are based on students' daily life, e.g., a simple student management system or a directory management system with functions of data input, output, sort and multiple ways to search, etc. Because students can combine with their practical situations and use actual data for system testing, they have a high enthusiasm in completing such a system.
- Based on comprehensive experiments of "Programming Language", all of the related experiments in "Data Structure", including linear list, hash table, searching and sorting can be improved to design more efficient data structures and algorithms. Combining the experiment or practice content of "Data Structure" with the school own situation, an information inquiry system for the campus can be designed, which comprises the inquiry function for each building and shortest path among scenic spots. Another

interesting practice scheme, using a stack to simulate compiler to complete expressions evaluation, can be designed. As these problems are focused on practical application, students have a great interest in solving these problems with the learned knowledge. They enjoy a great sense of achievement after finishing the corresponding experiment or the practice task.

- In the “Operating System” experimental task, queue structure is used in the process of scheduling and chain list is used in the storage management, etc. Students experience these experimental tasks in the process of using computers. Through simulating these algorithms, the mystery of internal management in operating system is uncovered, which makes students suddenly feel enlightened about the low-level management in computer [10].
- In the experimental task of “Compiler Principle”, the experiment results from "Data Structure” can be used. For example, the operator priority analysis algorithm in compiling program uses a grammar to judge whether expressions are legal and do further evaluation for the legal ones .Thus students get a deeper understanding of expression evaluation algorithm [11, 12].

By accomplishing a series of experiments on practical application problems gradually, students are not only able to apply theories to work, but also raise their interest and enthusiasm in learning and also improve their capabilities of data abstraction, algorithm design and analysis.

4. Establish a System of Multimedia Courseware

A common characteristic of "Data Structure", "Operating System" and "Compiler" is theoretical and abstract in curriculum content. There is some difficulty for students to understand and accept in the traditional teaching mode. Therefore, in order to get a better teaching effect, teaching methods can use visualization to make the abstract theory concrete, facilitate students understanding and memorizing and help them to eliminate the fare of difficulty. A system of multimedia courseware is developed to show the abstract ideas in "Data Structure", "Compiler Principle" and "Operating System" through interesting animations. Consequently, teaching is more intuitive and vivid so as that students obtain a more profound understanding of algorithms.

5. Conclusion

Through the overall optimization based on a series of data structure courses teaching content, preliminary results have been achieved. Table I illustrates the difference between Grade 2007 before reform and Grade 2008 after reform. The data analyzed comprises the final exam scores of the two corresponding courses and the student number of participating programming contests.

Besides, in respond to this curriculum reform, a questionnaire is designed to investigate the views of students. As a result 98 percents of them have approved of this reform method.

Synthesizing data of table I and survey results, some conclusions can be drawn as follows:

- It doesn’t need repeating examples to show that students have mastered the relevant knowledge of C/C++ language in “Data Structure” thoroughly. Thus, because of strengthening practice sessions on programming language, program design ability and debug skills of students are significantly higher than before.
- By introducing classical teaching cases, it helps to guide students to a specific problem with its background for study and research, raising them ability of translating realistic problems into the one solved by computer. The qualities of the relevant experimental works are obviously higher than before. More and more junior students take part in kinds of informatics related competitions and achieve better scores.
- By using the dynamic demonstrations to explain difficult algorithms, the course content turn to be systematic and visualized, so as to reducing the difficulty level of knowledge greatly, which helps students understand and master knowledge easily, stimulate students’ interest in learning and improve the learning efficiency.

TABLE I. FINAL EXAM SCORES AND PARTICIPATING CONTESTS

GRADE	Data Structure		Operating System		Participator	
	Average	***Pass Rate***	***Average***	***Pass Rate***	***Creativity Cup***	***ACM***
07	73.4	90.5%	74.5	91.7%	23	14
08	79.7	94.1%	78.5	93.6%	35	54

Cultivating the ability of data abstraction and algorithm design is one of the weak links in computer teaching. After the overall optimization based on a series of data structure courses , the students have a consciousness of choosing appropriate data structures and adopting corresponding algorithms, their competence in applying comprehensive knowledge to solve practical problems is improved obviously.

6. References

[1] The Joint ACM/EEE—CS Task Force off Computing Curricula. Computing curricula 2005-the Overview Report, ACM and IEEE Computer Society, 2005.

[2] WANG Zhi-wen ,CAI Qi-xian and LIU Mei-zhen, “Interpretation of and Consideration about Computing Curricula 2005(CC2005)”, Sci-Tech Information Development & Economy, vol.18, No.24, pp.164-165 , 2008, (in Chinese) .

[3] CHENG Hong-sheng, “Research on course integration of C Programming Language and Data Structures,” Computer Education, vol.7, pp.283-284, 2007 , (in Chinese) .

[4] WU Qing-jiang, HOU Jin and YE Shuang, Method of constructing computer curriculum with the core course, Data Structure, Journal of Huzhou Teachers College, vol. 1, no. 131, Feb. 2009 , (in Chinese).

[5] DING Xiao, Research into Teaching Method Reform of Data Structure Curriculum, 2008 International Conference on Computer Science and Software Engineering, Wuhan, China.

[6] WANG Chun-ling, Object-oriented Programming Language Experiment Guide Book, China Railway Publishing House, Beijing, 2010,8, in press ,(in Chinese).

[7] LI Qi-jian, On teaching of discrete mathematics, International Symposium on Computer Science&Technology 2009.

[8] YAN Wei-min and LI Dong-mei, Data Structure, Post & Telecom Press, Beijing, in press , (in Chinese).

[9] LI Ke-qing, “Researching in case teaching of Data Sructure ,” Journal of Yangtze University(Natural Science Edition), vol.1, no.4, pp. 135-137 ,2004 , (in Chinese).

[10] ZHAO Yu-hong, “Operating system reform in teaching and practice,” Computer Education, vol.24, pp. 31-32, 2008 , (in Chinese).

[11] LI Dong-mei,SHI Hai-hu, Compiler Principle, Post & Telecom Press, Beijing, 2006 , (in Chinese).

[12] LI Dong-mei, SHI Hai-hu, “Teaching research and exploration on Compiler Principle,” Computer Education, vol. 7, pp.103-104, 2008, (in Chinese).