

# *Exploration and Practice of the software series course Practical Training System in Application-oriented Undergraduate*

An Xinjun

Information Engineering Department  
Shandong University of Science and Technology  
Tai'an, 271021, China  
sdkdaxj@126.com

Liu Jimin, Su Na

Information Engineering Department  
Shandong University of Science and Technology  
Tai'an, 271021, China

**Abstract**—Based on the requirements for application-oriented undergraduate students, we analyzed the disadvantages in traditional training course. Such as single target, content dispersion, etc. We integrated the curriculum system, reconstructed the teaching content, realized the seamless cohesion between the curriculum and training, and established the three-phase training system which contains 19 ability structures and gives the clear targets in each phase. It ramps students' abilities to solve problems, to process complex information of data with object thinking. Further it strengthens their engineering practice abilities and develops their innovation and team-work spirit.

**Keywords**—*application-oriented undergraduate; problem solving; object thinking; engineering practice; three-phase practical training system*

## I. INTRODUCTION

In recent years, most of Chinese colleges or universities were reposition their target on the application-oriented undergraduate talents target. Application-oriented talents are cultivated to be talents for the actual use and for the production line according to the needs of society[1][6]. The objective of the training is to improve the students' comprehensive ability and quality by taking the market as the guidance and the general education as the foundation, so that it can lay the foundation for the students' professional learning and sustainable development. The training treats ability-developing as a main line, cultivating students' ability to use knowledge and technology, to solve practical problems, to equip themselves with a thick foundation, wide aperture, focus on practice and the application. The training plan defines the theoretical teaching and practical teaching systems respectively, appropriately increases the proportion of practice teaching, and integrates these two systems, giving priority to the development of their application ability and engineering practice ability.

At present, strengthening the training of teaching innovation and teaching method has become the focus of teaching reform in colleges and universities. Whether "The excellent engineer education and training program" in famous

universities [2] , or "The order form training mode" in high vocational colleges with IT enterprises [3], or "The practice of entrepreneurship education" and "The project-driven management mode" in Colleges and universities of Europe and the United States, Integrated the curriculum and training session[4][5] were common practice, even in the IT enterprise is their own business or professional practice, it is combined by the curriculum and practice with practice and employment. But for the application-oriented undergraduate especially in the underdeveloped areas in the information industry, large-scale implementation of the training mode, educational system, involving funds management, the burden of students, teachers and other aspects factors, are difficult to implement. According to its own characteristics, to explore and implement a school as the main body of the training system is a feasible way.

## II. PROBLEMS

In the traditional cultivation system, the computer major software curriculum practice includes the design/classroom experiment and teaching practice (Hereinafter referred to as the training):

### A. Problems in classroom experiment

Classroom experiments aim at quantities of curriculum knowledge and experiments. The experiments either verify the correctness and validity of some knowledge point or targets at the difficulty of the design. Because they aim at the curriculum knowledge points, they are constrained by the course schedule, time and other reasons. The lack of comprehensive application of knowledge points leads to the whole experiment process relatively discrete, closed, so after the completion of the course of theoretical study, it needs solving in a practical training form.

### B. Problems in training courses

Training courses are designed and integrated. Under normal circumstances, the training content is the main content of this course on the subject through the design, complete with comprehensive knowledge. Pointing to the curriculum itself, not so associated with the curriculum connection, the training task has certain limitations. In addition, the complete software

1. The construction of college special majors of Shandong Province in 2011.

2. Shandong University of Science and Technology teaching research project (QX2013139).

system encompasses all things; even the small one involves all aspects of comprehensive elements of software development, so the training mode is still lack of comprehensiveness, the practice is still relatively a discrete, closed system.

Take the traditional “database technology” practice as an example. The training content relates to the basic database operations, database table design, view and stored procedure design. Due to the restriction of time and schedule and other factors, the training courses usually focus on the database design while lack of demand for database applications and interface design. Therefore, there is no way to know whether there is database design and performance evaluation of defects. In contrast, “object-oriented programming” and “WEB technology” training courses focus on the development of platform application, problem analysis and solution design, code realization. Besides, database design and application are the important components of software system. But because of many factors like curriculum, teachers and the term design selection, the students find it hard to apply the “database technology” to the practical curriculum training, resulting in the disjointed, lacking of comprehensive training and system etc.

### III. THREE-PHASE PRACTICAL TRAINING SYSTEM

#### A. Our guiding principles

Aiming at the problems existing in training system, we integrated the curriculum group teaching and training system, the guiding principle of our curriculum and training system reform has been established. The procedure-oriented programming language as a tool, the “data structure” plus “algorithm” equal “program” as the core content learn how to solve the formal computer problems, in order to develop the students' ability to solve the problem. Based on the abstraction of “data structure”, “OOP” and “Database application” as the core content, make the students to learn how to solve the problem of multi-type complicated data processing with object thinking, to preliminary master the basic process of object oriented software engineering, to develop their object-thinking ability; Based on the development technology of WEB , the modern software development platform application as the core content, strengthen the skills of students' programming, improve students' engineering practice ability and enhance the innovation consciousness and team-work skills.

#### B. Our guiding principles

The three-phase practical training system is the basis of courses that are related to software. In the course of classroom teaching, the three-phase practical training system, that is, the “problem solving”  $\Rightarrow$  “object thinking”  $\Rightarrow$  “Engineering practice” progressive process is put forward on the basis of the experiment, according to the objectives and requirements of curriculum, focusing on “ability” culture as the main line. With the three-phase practical training system back to promote research, the nature of the issue transits from the calculation problem to the practical engineering problems; the project size and complexity and the difficulty also gradually increase. The individual work gradually transits to the team work and training duration is gradually extended, so the direction of

training is clearer, namely “view tools application, problem analyzing and programming as the basic ability, take software engineering issues as the content of training to gradually train students’ engineering practice ability, to cultivate students’ thinking ability, team cooperation and innovation consciousness”. As shown in figure 1.

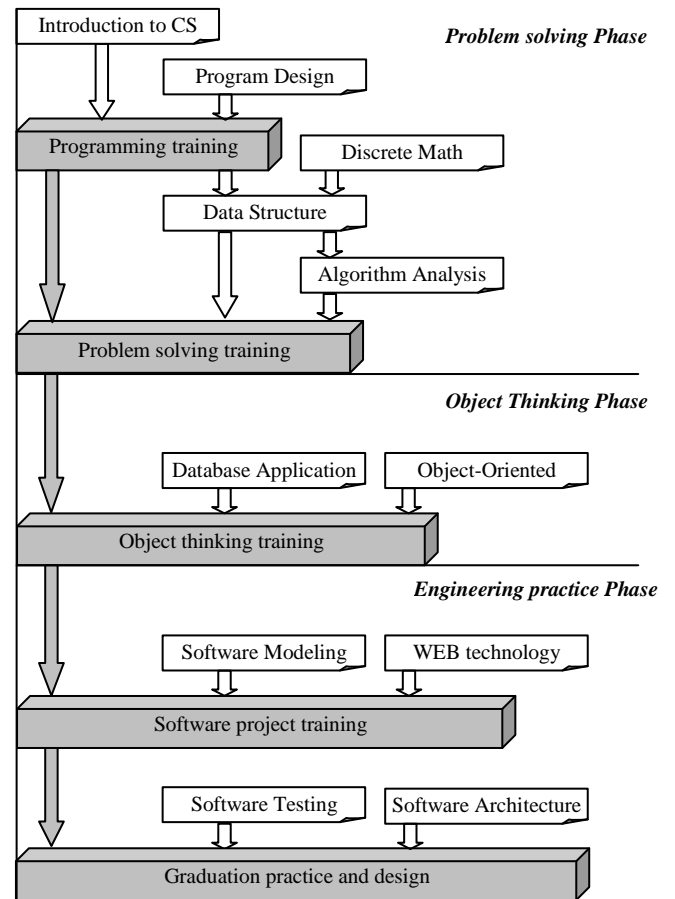


Fig. 1. Schematic diagram of the three-phase practical system software courses training progressive structure.

As a complete training system, with the deepening of the training progress, the fundamental theory and knowledge of computer science are also integrated into the training courses. As shown in figure 2.

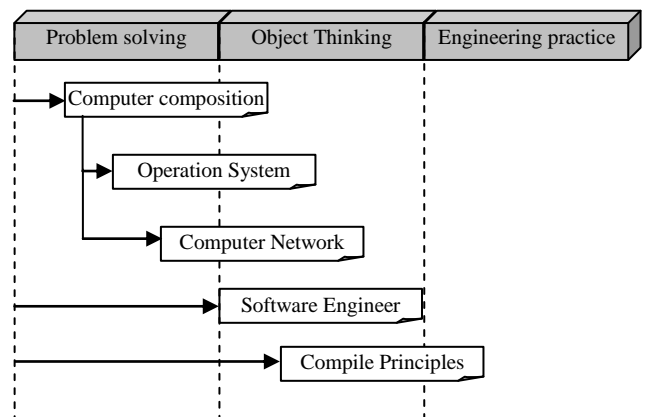


Fig. 2. The fusion diagram of the theory course and practical system

The three-phase practical training system follows the way from simplicity to difficulty. With practical training link layer extended downward, the scale of the problem complexity increases, and the requirement of the students' knowledge, skills and ability should be enhanced and expanded.

#### IV. IMPLEMENTATION

"Three-phase" training system contains 19 ability structures of clear objectives for each phase, highlights the using of tools, strengthen the abilities of analyze problem and programming. With the extension of training, the "practical" ability is come up with as the new ability structure of the target, based on CC2001 and CC2005[7][8], as shown in table 1.

TABLE I. PHASES/ABILITY STRUCTURE UNIT

Ability Structures	Training phase				
	Problem solving		Object thinking	Engineering Practice	
	Programmi ng	Problem solving	Object thinking	Software project	Graduation practice and design
Tool application	●	●	●	●	●
Language foundation	●	●	●	●	●
Problem analysis	●	●	●	●	●
Problem abstract	●	●	●	●	●
Formal expression	●	●	●	●	●
Basic algorithm	●	●	●	●	●
Coding realized	●	●	●	●	●
Coding style	●	●	●	●	●
Innovation	●	●	●	●	●
Data organization		●	●	●	●
Algorithm optimization		●	●	●	●
Document specification			●	●	●
Objective thinking			●	●	●
Process engineering			●	●	●
GUI design			●	●	●
Team-work				●	●
Software system				●	●
Test and deployment					●
Literature retrieval					●

##### A. Problem solving Phase

"Problem solving" phase is the training session after students complete the foundation of the programming design

(C language), discrete mathematics and the data structure of the curriculum. For the purpose of cultivating students' computational thinking, and strengthening the students in the process of language (C and C++ language) application, coding ability, the rigorous scientific attitude and good programming habits, according to Nanjing University "problem solving" series of course, we have reformed professional introduction, basic programming, discrete mathematics, data structure and algorithm analysis of series main courses. And we carry out the integration of content based on maintaining the original curriculum guide subsequent order. Due to the aforementioned problem solving courses last semester long and large span, when the specific plan is put into effect, the phase is divided into two different units:

- Training program design. Solving the practical problem with the computer includes two factors. First of all, abstract the specific problems to the computational ones. The second is to find a feasible calculation process. Specialized training program design is carried out based on these two elements. It concerns basic knowledge about C language, compiler environment, coding style programming language and other contents in the field of program design language training. The training points include the basic data structure, algorithm logic and expression, and the relationship between the set of operations. It enables students to understand the program running in memory of the basic principle and process, know how to control the simple calculation process, and cultivate their basic formal and abstract expression ability.
- Problem solving training. The core problem of this phase is the implementation of specific terms; it is realized through a series of steps to solve problems, which transforms inputs into outputs. For the calculation of all problems, all cannot do without the calculation object or the processing of information, how to efficiently organize them up has a great impact on the lifting algorithm feasibility and efficiency.

Problem solving training is the extension of last unit, is a practical training after the systematic learning of "data structure" and "algorithm" courses to enhance the ability of solving problems. It involves the knowledge points about nonlinear and linear data structure and logical structure, physical structure design, typical algorithms, modeling and solving complex problems etc. The points are to empower the students to solve the numerical and non numerical problems, and to build a computer problem solving model.

This phase is crucial to the training system for application-oriented talents; a good grasp of degree is the priority. We cannot weaken or even ignore this process. In fact, because the courses and training of the problem solving strengthen the students on the basis of the learning phase of knowledge, they have an important effect on the cultivation of students' professional ability, good study habits and learning. The contents and requirements of this phase of training have not yet truly combined with engineering application; they are still in the consolidation phase of basic theory, so this phase of training is accomplished in schools as the main body.

### B. Object thinking Phase

The object is object-oriented abstraction of reality in the level of data structure, including the data abstraction and act abstraction. Compared with the process-oriented thinking, object-oriented thinking is more in line with the law of human cognition of the world. A more realistic question is the modern software engineering development and application all adopt the integrated development platform of object oriented, so this phase not only strengthens students' programming design methods and the training of skills, but also pays more attention to the students from the design thinking process oriented to object oriented design thinking, "Thinking" phase cultivates students' ability to solve the practical problem with object oriented thinking.

"Thinking" phase of training is carried out after the students complete object oriented programming, database technology and application of series courses. It covers the design and realization of application, object oriented integrated development environment (including the type of database modeling and instantiation), GUI design, C/S architecture software design patterns. The phase of the training project is closer to the reality of engineering application. The object to be processed is more complicated.

"Object thinking" training plays a vital role in the whole training system and functions as a transition. On the one hand, it summarizes and synthesizes the learning outcomes, so that the process of solving problems become more practical and more complex based on object oriented technology platform. It enables the students to understand the modern software development process and pattern.

On the other hand, the series of courses and training for engineering practice lay a solid foundation for a good knowledge, thinking, ability and consciousness of curriculum. The training content and requirements of the phase begin to relate to the actual engineering application problems, therefore, in practice we use the "going out, please come in" approach to invite technical staff in IT enterprises combined with the practical problems in students' training.

### C. Engineering Practice Phase

The core competence of application-oriented talents is the "Engineering" [6]. Training students' engineering practice ability mainly includes the understanding of user needs. To achieve the three main aspects of design and code implementation, we need to start from three aspects background, schemes and techniques. In the specific implementation, the schedule is divided into "software engineering practice project" and "graduation practice and design".

- Software project. It refers to the period of 4-5 weeks (the specific implementation according to the school calendar) training after the completion of the development of WEB technology, the software modeling technology and other major system development course, aiming to train students to follow the engineering ability of analysis, design, development, testing and deployment of software,

follow the PSP personal software process, step by step cultivate students' software development of individual ability, and make the students grasp the basic quality of staff in IT enterprises. The "task-driven" mode is applied to strengthen the training of design, code realization, engineering, focus on the content analysis, design and coding implementation, software requirements specification documents.

- Graduation practice and design. Graduation practice and design for undergraduate teaching is the last segment, also at the end of the ring of the three-phase training system. Compared with the previous practice, graduation practice and design of the subject is often large scale, some projects even are the teachers' scientific research project, therefore it is difficult to complete single-handed. The graduation design follows the procedure---organize students to set up a team, complete the design with team work, and ask teachers to regularly listen to the report in the process and conclude the design through the graduation reply form knot.

### V. SUMMARY

Therefore, graduation practice and design process not only cultivate students' engineering practice ability, but also exercise their communicative competence, team cooperation consciousness and innovation consciousness. This phase of training serves as the core of the three-phase training, and is the synthesis of knowledge, thinking and engineering practice.

### REFERENCES

- [1] Liao Jin yuan, "Discussion and Practice on Engineering Talents and Applied Talent", RESEARCH AND EXPLORATION IN LABORATORY, Vol.27, pp.132-134, December.2008
- [2] Shen Qi, Zhang Yan, and Luo Yang, "Construction and reform of application-oriented undergraduate practice teaching system" Experimental Technology and Management, Vol.27, pp.36-40, October.2010
- [3] Peng Yan fei, Shen Xue li, and Zhang Quan gui, "Study and practice on experimental teaching method of computer software courses", Experimental Technology and Management, Vol.29, pp.174-176, April.2012
- [4] Sun ying, "On Method of Curriculum System Construction For Application oriented Undergraduate Education-Case in Major of Computer Software", Journal of Shenyang University(Social Science), Vol.15, pp.522-524, August.2013
- [5] Wang Yong, Wang Zhong qun, "Research on Application-oriented Software Curriculum Group", JOURNAL OF CHANGCHUN UNIVERSITY, Vol. 21, pp.98-102, December.2011
- [6] Wang Yong, Wang Zhong qun, and Xie Xiao dong, "Research on the Teaching of Object-oriented Programming Course Oriented to Software Curriculum Group", JOURNAL OF CHANGCHUN UNIVERSITY, Vol.22, pp.498-501, April.2012
- [7] The Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery. Computing Curricula 2001 Computer Science.[EB/OL]. [http://www.acm.org/education/education/education/curric\\_vols/cc2001.pdf](http://www.acm.org/education/education/education/curric_vols/cc2001.pdf)
- [8] The Joint Task Force for Computing Curricula 2005.Computing Curricula 2005 The Overview Report [EB/OL]. [http://www.computer.org/portal/cms\\_docs\\_ieeeecs/ieeeecs/education/cc2001/CC2005-March06Final.pdf](http://www.computer.org/portal/cms_docs_ieeeecs/ieeeecs/education/cc2001/CC2005-March06Final.pdf)