

# CDIO TRAINING MODE OF PROGRAMMING ABILITY FOR SOFTWARE ENGINEERING STUDENTS BASED ON ACM/ICPC COMPETITION STANDARD

Yunfeng Mu, Dianlong You, Yan Dou

College of Information Science and Engineering, Yanshan University

## ABSTRACT

Programming ability is the basic ability for the students in software engineering major, and the premise to enhance the ability of engineering practice. In view of the problems that there's a great demand gap for high-tech talents in the software industry and the graduates are lack of high level programming and engineering practice ability. Based on the CDIO teaching concept, we systematically design the training matrix of the programming ability. By putting the ACM/ICPC contest questions into the teaching, experimental teaching and assessment, and at the same time, using the teaching forms of 'learning by doing', 'project-based learning' and 'collaborative Learning', combining with the ACM/ICPC platform, we can inspire the students' computational thinking, enhance the ability to analyze the data and algorithm, as well as improve the capacity of programming.

## KEYWORDS

ACM/ICPC, programming ability, CDIO, training mode

## ISSUES RAISED

In the past few years, the software industry has developed rapidly, and there has been a great demand for high-tech talents. Although a lot of graduate students are trained for the software industry, they are not good enough to take the job immediately. Even some of them can't draw flowcharts and write the files. They compiled little codes and got no ideas about good data structure and they just implemented the program inefficiently. It is all because of their poor programming skills and inability of engineering practice. They just have a low level of programming skills, and they are in lack of practical capacity in engineering practice.

Program design ability consists of the abilities of computational thinking, data representation, algorithm design and analysis, program design and implementation, and systems analysis and development. It is the basic ability that software engineering students should have, and it is also one of the criteria to test their quality. Also it is the premise to enhance the ability of engineering practice[1,2]. But still there are some problems in teaching content, experimental teaching and the evaluation process.

**(1) *The courses lack the depth of convergence as well as the systematic and continuous process, which are designed to enhance the ability of programming.***

Although the software engineering starts with the core course such as "C / C ++ programming, data structures, algorithm design and analysis, calculation methods", they are independent and lack connections between each other. We rarely converge the courses orderly by using the case studies and project training, so we failed to instill the basic methods and ideas. Obviously, students often remain on the low level, they lack computational thinking and abstract thinking, so they can't solve practical problems on a high level[3,4].

**(2) Content knowledge is obsolete, conservative, lacking of depth and imagination. It is difficult to expand the student's thinking and desire to explore the mobilization.**

The cases used in the traditional courses are obsolete, virtual, and classic cliché. There are big differences between the class and the practical problems, so it is hard to keep up with the development of the industry. The cases are easy, unattractive and not challenging, thus students find no excitement from them, and they even think it is boring and useless, which makes them lose the motivation to continue the study. Based on this, the product made by the student could only be called 'work' rather than the cleverly conceived 'art' that reflects the imagination and creativity, which is what the companies really want.

**(3) The experiments are too patterned and lacking of high level practical resources.**

Programming training topics are mainly from text book for the particular knowledge. Students have seen the low-level, boring and tedious experimental topics many times such as "seeking the maximum minimum, the Fibonacci sequence, daffodils, multiplication tables". They are full of old, single, descriptive experiments, but less creative, design and comprehensive. It leads students to plagiarize, to cope with the task. So it has no challenge and fails to inspire the innovation. As for the guidance on experiments, teachers often make a ready-to-do preparation including the experimental environment, the detailed experimental guide books and the reporting framework. Students only need to complete the task step by step by following the guidance.

**(4) The evaluations emphasize scrutinization, but neglect the motivation. They emphasize the results, but neglect the process. They emphasize the authority, but neglect the creativity.**

In the past, the tests focused on basic theories and concepts, and the comprehensive questions were insufficient. Also, the evaluation was single and obsolete. Students still can't get rid of the bad habits of following whatever the teachers told them. They are more concerned for the grades rather than the knowledge itself. As we know, it is professional to use the good program syntax and it is artistic to use a good algorithm. Though the different codes can fulfill the same function, its efficiency, readability, reusability, robustness and portability may vary considerably. Chinese programmers often focus on the functional requirements, but often overlook the minor issues such as application performance and exception handling. Even some users have the idea that domestic software products are less useful and they dare not to use them.

To solve the above problems, we put forward CDIO engineering education philosophy that put the standard of ACM/ICPC(ACM International Collegiate Programming Con-test) into the program design training and use the engineering design as a guide to the develop personal skills, team spirit and system control capability, forming the training model that consists of "lecture, experiments, training platform and evaluation system"[5].

## ACM / ICPC AND THE CULTIVATION OF PROGRAMMING CAPABILITY

ACM / ICPC, sponsored by the American Association for Computing Machinery which is an International authoritative organization, is recognized worldwide as the largest and highest level of International Collegiate Programming Contest[6,7]. Because all previous competitions gathered the meta-world elite students and the future stars, it has caught the attention from international and domestic well-known universities. Even much of the world's leading IT companies such as Microsoft, GOOGLE, IBM, Oracle give sponsorship and support, offer award-winning students scholarships and hire them directly with high salaries. ACM award certificates are recognized by the well-known universities and IT companies. It has become the world's most influential international computer competition. ACM/ ICPC tests the players' abilities of innovation and working under pressure.

Students in ACM/ICPC use the computer to fully demonstrate the ability to analyze and solve the problems. These problems are encountered in practical engineering applications and cover a wide range of knowledge such as programming, data structures, algorithm design and analysis. As shown in Table 1.

Table 1. The Main Knowledge of ACM/ICPC

Courses	Knowledge
Programming language	C/C++,C#
Data Structure	Primary: list / stack / queue / binary tree/ heap Advanced: segment tree / Fenwick tree / SBT / suffix array / array
Algorithm Analysis	Dynamic programming, greedy algorithms, graph algorithms.....

## CDIO TRAINING MODE OF PROGRAMMING ABILITY

We need to put the ACM/ICPC contest topics and test platform into the development of the programming ability and build CDIO training matrix, in order to enhance students' computational thinking skills, ability to represent data and the capacity for algorithm analysis and implementation. It includes the following aspects. (1) Connect the ACM/ICPC topics with the traditional core courses, forming a targeted, appropriate lectures and experiments. (2) Put the "evaluation thinking of ACM/ICPC" into the assessment, and build evaluation system in line with "CDIO" student programming ability matrix.

### **Matrix of Training Mode**

Develop students' programming skills in the first two years, and select the key courses as the curriculum modules based on program design capabilities. As shown in Figure 1. In the freshman year, students focus on programming syntax, computational thinking, program design and implementation capabilities; and the sophomore year on major data representation and algorithm design and analysis capabilities. Meanwhile, in various stages of training, learning through curriculum theory, practice, and after-school programs so they can improve students' programming ability gradually[8].

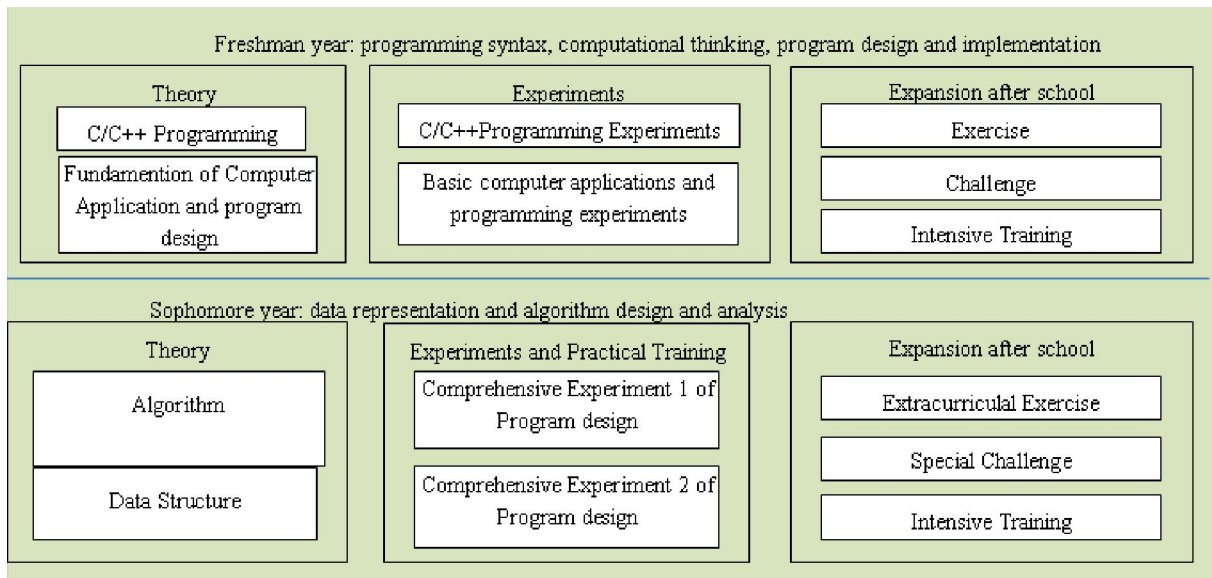


Figure 1. Course matrix in training of programming ability

### Implementation Matrix of CDIO Training Mode

Curriculum Reform need to put the ACM/ ICPC contest topics into three levels-teaching (including theoretical teaching and experimental teaching), the expansion after school and the assessment. So it could be sustainable, and there would be mastery of knowledge, and implementation of convergence matrix shown in Figure 2.

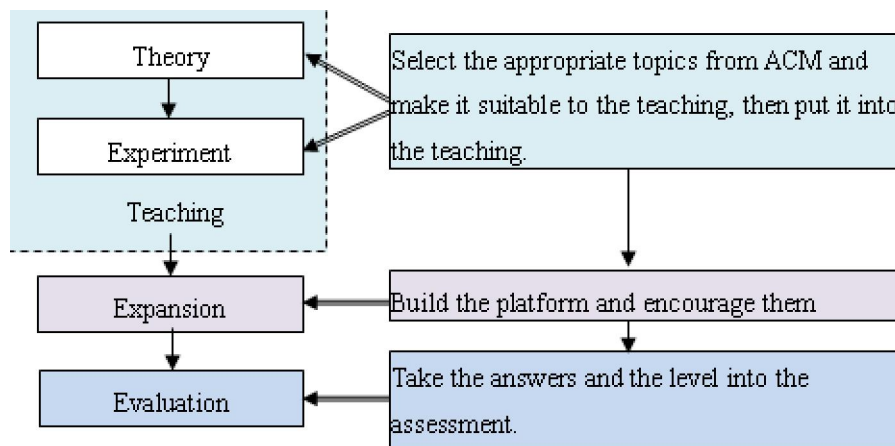


Figure 2. Implementation matrix

The implement of training mode should start from "teaching theory, teaching experiment, training platform, expansion after-school and assessment" and other aspects, specifically including the following:

- (1) **Refine and extract "ACM / ICPC contest topics" into the curriculum matrix of each course module, promote the convergence of the courses and enhance the teaching content systemically.**

The questions extracted from the test database will be divided by primary level and advanced level in accordance with the level of difficulty. And they are classified according to the type of knowledge to the C / C ++ programming, data structures, algorithm design and analysis courses. Take the key knowledge 'map' in Data structure as an example, the primary level corresponds to classroom teaching of theory, curricular experiments and comprehensive experiment. Advanced level corresponds to curriculum design, training and expansion after-school, as shown in Table 2. In addition, ACM / ICPC topics database are updated every year, which ensures that the novelty and cutting-edge knowledge of the content and mobilizes the students' desire to explore more.

Table 2. Knowledge Classification and Topic Extraction for Graph in ACM

Primary		Advance	
Knowledge	Examples	Knowledge	Examples
The basic depth-first search algorithm	poj2488,poj3083,poj3009,poj1321	Deep search optimization, bidirectional search, rotation search, IDA * algorithm	poj3131,poj2870,poj2286
BFS basic algorithm	poj3278,poj1426,poj3126,poj3414,poj2251	Bitwise compressed storage state, bi-wide search, A * algorithm.	poj1768,poj1184,poj1872,poj1324,poj2046,poj1482
Basic shortest path algorithm	poj1860,poj3259,poj1062,poj2253	Shortest Path Problem model of setting up and solving	poj3155, poj2112, poj1966,poj3281,poj1087
Augmenting path algorithm for maximum flow	poj1125,poj2240 , poj1459,poj3436	Model to establish the maximum flow problem and solution	poj2289,poj3216
The basic minimum spanning tree algorithm	poj1789,poj2485,poj1258,poj3026	Degree limit minimum spanning tree	poj2446 , poj1639

**(2) Put ACM/ICPC topics into teaching to enhance the depth of knowledge and facilitate knowledge migration.**

Put ACM/ICPC topics into the teaching of software engineering students, and construct a high-level study projects that can enhance the depth of learning, inspire students' computational thinking, and promote knowledge migration. Change the traditional knowledge-centered teaching methods to focus on ACM-related topics, let students begin the project with their own questions. Take data structure as an example. When you talk about the array and list, give the questions like 'which one is better when you storage the student achievement, and what about the store gas station waiting for refueling vehicles problem?', so you can guide students to transform abstract problem to practical one.

**(3) Transform "ACM /ICPC" topics into a multifaceted experimental project.**

The mode of ACM aims to solve the issue first, so the approach is not restricted. Then students have more chances for innovative thinking.

By selecting and processing the topics from ACM, and then introducing them to the experimental teaching, we can improve the level of the experiment as well as the students' ability to solve practical problems and knowledge migration ability. Specific measures include the following.

First, divide the experiment into two types----validating one and design one. Take data structure as an example, as shown in Table 3. Validating experiment focuses on the specific knowledge of the courses, and the design experiment focuses on the application of multiple levels of knowledge. Transform the traditional experiment which focuses on the knowledge to the new one which focuses on solving problems. Selecting the topics from ACM and transferring it into the experimental problems related to the core courses, so that students can choose the questions based on their own abilities. Then they could design rational data structure and algorithms, and solve the problem on the computer, which would exploit their potential and enhance their understanding of the knowledge. Then, according to the degree of difficulty, design experiment for each difficulty level. It will significantly inspire students' innovation by multiple solutions and interesting experiments. At the same time, give both experiments with correctly executed test data, and let the students finish the task in the given time according to the difficulty level. Their program will be submitted to the evaluation system to test, and the results and performance data would be returned immediately so that the teachers could offer help to them with the problems.

Table 3. Design of Data Structure

Type	Content	Purpose
Validating experiment	Linear lists, stacks and queues, strings, arrays, and generalized tables, trees and binary trees, diagrams, sequence binary search, internal sorting	Require students to master specific basic data structures and methods and have the strong ability to solve practical problems.
Design experiment	Parking management, Huffman encoder, phone number inquiries, student course management system.	Requires students to master specific algorithm design and analysis, skill in recursive ideas and methods, have the ability to design data structure and construct complex software systems.

**(4) Through "learning by doing", "project-based learning", "collaborative learning" and other forms to carry out project training, by using "ACM / ICPC online evaluation system".**

By "learning by doing", "project-based learning", collaborative learning", etc, combined with the ACM / ICPC platform, give students environment to create, design, develop, implement and operate, which is good for students to deepen understanding of abstract knowledge and improve students' ability to innovate and teamwork. Students can take advantage of ACM / ICPC online evaluation system to carry out independent practice and intensive training. Teachers can recommend the topics for the students. A knowledge point may contain different questions and the difficulty may vary from one to another. So students could choose the topics based on their own situations. Students can work on the topics at any time and submit the program online. Evaluation system will verify the correctness of the program and return to the users the running time and memory usage of the program. Students can also view other peoples' program and compare each other. For some students who are capable

enough and have the time, they could be arranged some more practical and larger topics, so that it would meet the requirement of different levels.

**(5) Put the "ACM / ICPC evaluation of thinking" into the assessment, build evaluation system in line with "CDIO ability matrix".**

Change the traditional assessment that the teachers verify the program and give the scores according to standard answers and their own understanding. The programs may vary from each other so it is impossible for the teachers to grasp all of them and they may do the wrong judgment. Learning from ACM black box method, we could pre-set multiple data to test the program submitted by the students, so that it could avoid the wrong judgment and prevent the defects of the student self-testing method. It may also reduce the intensity of the work of teachers. At the end of the semester, require every student to finish the topics on their own, using the knowledge to design the algorithm, program debug and run. It will be evaluated by the numbers of questions they submitted and the performance of the program. Putting evaluation into programming practice and the algorithm analysis, into the practical cases, they will test student, incentive students and guide students.

## **IMPLEMENTATION EFFECT OF CDIO TRAINING MODE**

At present, CDIO training mode of programming ability based on ACM/ICPC competition standard has been put into effect and become popular among software engineering students in Yanshan University, which is highly evaluated by the majority of teachers and students. Take the teaching reformation in software engineering (60 students in Yanshan University) as an example, the proposed CDIO training mode of programming ability based on ACM/ICPC competition standard is put into effect especially on classroom lectures, experiments, training platform and evaluation system. The teaching practice shows that the students trained by CDIO mode get greatly promoted in computational thinking, the ability to analyze data and algorithm, and the capacity of programming. Compared with the students in related major (for example, information safety major with the same basic courses except for CDIO training mode, 30 students), software engineering students obviously take on strong programming ability. Especially in the following courses such as "programming --- curriculum design", the trained students show strong ability of engineering practice, ability to communicate, personal comprehensive ability, and team cooperation ability. They have high sense of responsibility and noble occupation morality.

## **CONCLUSION**

Programming ability is the basic ability of software engineering graduates and the key to develop the engineering practice ability. We can put ACM / ICPC topics, training environment and evaluation system into the development of the software engineering students; build the development mode in line with the CDIO which comes from "lectures, experiments, training platform, the evaluation system" and use the engineering design as the guide to develop students' skills, teamwork and system control capability. And we can gradually put carry out the reform to form a new comprehensive system of training defined as "from theory to practice, from curricular to extracurricular". Finally it will improve the depth of learning, promote understanding of the problem, strengthen the students' ability and inspire interest in learning. Ultimately, students' programming ability will be significantly improved.

## ACKNOWLEDGMENT

The authors wish to thank Professors C.W. Wang and F.D. Zhao in Yanshan University for their valuable comments and suggestions. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation. This work is partially supported by the Teaching Reformation Project of Yanshan University (No.zzjg201412) and Scientific Technology Research and Development Project of Qinhuangdao City (No.2012021A046).

## REFERENCES

- Geng G.H. (2009). Search and practice on training mode of programming ability. *China University Teaching*, (3), 29-32.
- Yang S.T., & Li J. (2014). Training of computational thinking on ACM/ICPC programming competition. *Heilongjiang Researches on Higher Education*, (10), 175-176.
- Gao J.Y., & Zhu Q.X. (2008). Training of programming ability in series courses. *China University Teaching*, (11), 47-48.
- Han J.M., & Wang L.X. (2012). Layering training mode of programming ability for university students. *Computer Education*, (3), 15-22.
- Gu P.H., & Shen M.F. (2008). From CDIO to EIP2CDIO: A probe into the mode of talent cultivation in Shantou University. *Higher Engineering Education Research*, (1), 12-19.
- Yang S.T., & Wang B. (2010). Study of university students' ability training based on ACM/ICPC competition. *Computer Education*, (24), 16-18.
- Gou S.P., & Yang P. (2010). Research of course system construction and training mode for creative talents with ACM/ICPC competition as a carrier. *China University Teaching*, (7), 71-73.
- Ma X.Y., & Wang G.P. (2011). Research on programming-ability-oriented teaching system of program design courses. *Computer Education*, (4), 74-79.



## BIOGRAPHICAL INFORMATION

**Yunfeng Mu** is a teacher in the College of Information Science and Engineering at the Yanshan University. He collaborates regularly with universities in P.R. China on topics related to curriculum design and the improvement of teaching and programming. His current research focuses on the cultivation of programming ability in software engineering students and ACM/ICPC competition.

**Dianlong You** is an Associate Professor in the College of Information Science and Engineering at the Yanshan University. His current research focuses on curriculum development methodology and programming practice.

**Yan Dou** is an Associate Professor in the College of Information Science and Engineering at the Yanshan University. Her current research focuses on programming platform development and professional education.

### **Corresponding author**

Mr. Mu Y.F.  
College of Information Science and Engineering  
Yanshan University  
No. 438, West Hebei Avenue  
Qinhuangdao, Hebei Province, P.R. China, 066004  
+8613933568208  
[yfm@ysu.edu.cn](mailto:yfm@ysu.edu.cn)



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License](https://creativecommons.org/licenses/by-nc-nd/3.0/).