Instructive Cultivation Plan for the Program of Software Engineering

(Grade 2020)

Course code: 080902

1. Orientation

In order to train the software engineering and technical talents required for the development of the global network economy and society, this training program is based on actual engineering, takes engineering technology as the main line, and aims at cultivating application-oriented software engineering and technical talents of the software-related programs who have good software design and development capabilities, and software testing and operation and maintenance capabilities.

2. Cultivation Objectives

2.1. General cultivation objective

This program cultivates engineering and technical application-oriented talents who are comprehensive developed on the aspects of morality, intelligence, physical education, beauty, and labor, have a solid theoretical foundation knowledge, have the ability to analyze and solve complex engineering problems and the ability to communicate and collaborate, manage and innovate, and can be engaged in software development, technology application, quality assurance and project management in the field of software-related professional fields, especially in information technology outsourcing field, and can meet the needs of technological progress and social development.

2.2. Objective of value guidance

With the objective of cultivating application-oriented engineering and technical talents that adapt to social development, in the implementation process of education and teaching, the values of engineers and engineering professional ethics will be integrated in the teaching, and this program will train students to have good moral, humanistic, scientific and professional qualities, and obtain an international vision, a sense of social responsibility, steadfast integrity, teamwork ability, lifelong learning ability, innovative spirit, and hard working ability.

2.3. Objectives students must achieve five years after graduation:

2.3.1: Master the multi-disciplinary comprehensive knowledge required in software engineering related professional fields, and have the ability to analyze and solve complex engineering problems in software engineering related professional fields through completing phased comprehensive practical projects with increasing difficulty and degree of integration.

2.3.2: Have a sense of social responsibility and engineering professional ethics, understand and respect the laws and regulations related to the software industry, have good communication skills, teamwork skills, and be competent for a team leader.

2.3.3: Have an innovative spirit and an international vision. Master at least one foreign language, have a basic understanding of the international situation of software engineering and related fields, and be able to communicate and exchange under a cross-cultural context. At the same time, be able to choose to select the corresponding second foreign language according to the needs of software outsourcing.

2.3.4: Master the basic principles of software engineering disciplines and software project management, be able to use software engineering methods to model, analyze, design, and test software engineering problems, and analyze and optimize models through experimental results. Be

able to engage in product development, technology application, quality assurance and on-site management in software engineering related professional fields.

2.3.5: Have the consciousness and ability of independent learning and lifelong learning, and be able to adapt to the changes in scientific and technological progress and the needs of social and economic development. Can recognize and understand the characteristics of rapid knowledge update, new technologies and new methods in the field of software engineering, and be able to establish stable career objectives.

3. Requirement for Graduation

3.1. Engineering knowledge: Have a solid natural science foundation, master the basic theories and application methods of advanced mathematics, engineering mathematics, college physics and other basic courses; be able to use mathematics, natural sciences, engineering foundations and professional knowledge to solve complex software engineering problems.

3.2. Problem analysis: Be able to apply basic principles of mathematics, natural science and engineering science to identify, express, and analyze complex software engineering problems through literature research and obtain effective conclusions.

3.3. Design/development of solutions: be able to design solutions to complex software engineering problems, design systems, units (components) or process flows that meet specific needs, and be able to reflect the sense of innovation in the design process, while taking into account social, health, safety, legal, cultural and environmental factors.

3.4. Research: Graduates are able to understand the basic principles of software engineering, and can use scientific methods to research, analyze, model, design, and test complex software engineering problems, and be able to analyze and optimize models through experimental results.

3.5. Use modern tools: Graduates are able to use various common tools (development tools, modeling tools, analysis tools, management tools, testing tools, etc.), environments and platforms, and understand the differences between these tools, environments and platforms and their applicable areas, meanwhile, are able to select appropriate tools, environments and platforms for experiments and analysis of complex software engineering problems

3.6. Engineering and society: Be able to conduct reasonable analysis based on the background knowledge of software engineering, evaluate the impact of software professional engineering practices and complex software engineering problem solutions on society, health, safety, law and culture, and can understand the responsibilities that should be undertaken.

3.7. Environment and sustainable development: Be able to understand and evaluate the impact of professional engineering practices for complex software engineering issues on the environment and sustainable development of society.

3.8. Professional norms: Have a good foundation in humanities, arts and social sciences and have a sense of social responsibility, be able to understand and abide by engineering professional ethics and norms in software engineering practice, and always perform responsibilities.

3.9. Individuals and teams: be able to assume the roles of individuals, team members and leaders in a team with a multidisciplinary background.

3.10. Communication: Be able to effectively communicate and exchange with industry colleagues and the public on complex software engineering issues, including writing reports and design manuscripts, making statements, expressing clearly or responding to instructions, and have a certain international perspective, master a foreign language, have good listening, speaking, reading, and writing skills, be able to read the foreign books and materials of the program smoothly, and be able to communicate and exchange under a cross-cultural context.

3.11. Project management: understand and master the principles of software project management and economic decision-making methods, and be able to apply them in a multidisciplinary environment.

3.12. Lifelong learning: through innovative practical projects and corporate internships, can help students recognize and understand the characteristics of rapid knowledge update, new technologies and new methods in the field of software engineering, and have the awareness of independent learning and lifelong learning, and have continuous learning and adaptive development ability.

4. Schooling System

Four years

5. Length of Study

Flexible study period, generally four years, the minimum length of flexibility is not less than three years, the longest not more than six years.

6. Requirements for Graduation and Degree Conferring

Students of this program must complete the minimum credits required for each category of courses and complete all the content specified in extracurricular class according to the requirements of the instructional training plan, and the total credits must reach 152 credits for graduation; those who meet the requirements for bachelor's degree can be conferred bachelor degree in engineering.

7. Disciplines

Software engineering

8. Core Courses

8.1. Introduction to Software Engineering

The purpose of this course is to introduce the basic characteristics and concepts of software engineering and service outsourcing, solutions to general engineering problems, the economic characteristics of engineering projects, and the tasks and responsibilities faced by engineers, so as to enable students to understand the relevant engineering programs, stimulates students' interest in learning software engineering programs and clarify their learning motivations. Through the study of this course, students can put forward some ideas and thoughts to solve engineering problems when facing general engineering problems; can cultivate students' interpersonal communication and teamwork spirit; and lay a certain foundation for them to learn subsequent professional courses and proceed smoothly.

8.2. Foundation of programming design

This course mainly teaches the basic concepts and basic techniques of programming design. Taking C language as an example, students are required to be more proficient in the grammar and semantics, master the basic methods of structured programming and knowledge points including data types, control structures, functions, arrays, files, operating mechanisms and preliminary commissioning, master some common programming design skills, master programming techniques of top-down refinement and cultivate good programming habits and styles, thus enabling students to master the basic process of computer programming operations and the basic methods of eliminating grammatical and semantic errors.

8.3. Data structure and algorithm

This course mainly teaches data construction methods and algorithms for operating these data structures. The focus of this course is on various typical data structures and their storage structures, related algorithms and basic spatiotemporal analysis, including linear tables and their derived structures (stacks, queues, strings and multidimensional arrays), trees and graphs, and typical algorithms for search and internal sorting. The focus is to enable students to further master more standardized algorithm design skills and improve their thinking skills on the basis of the existing programming capabilities.

8.4. Principles of Computer Composition

This course focuses on the basic composition and working principle of the single CPU computer hardware system of the von Neumann architecture, and systematically describes the internal structure, functional characteristics, working principles, interaction methods and basic design methods of the computer hardware system and its functional components. At the same time, through the combination of classroom teaching, course experiment and course practice, students can systematically understand the organization structure and working principle of computer hardware system, and master the basic analysis methods of computer hardware system. The main contents of this course include: overview of computer composition, machine representation of values, calculation methods and calculation components, storage systems, instruction systems, central processing units, input and output systems, buses, etc.

8.5. Introduction to Database System

This course mainly teaches the basic concepts and basic theories of database systems. The main contents include: the progress of data management, the composition of database systems, three basic data models (focusing on relational models), and the standard design of relational models (including functional dependencies, paradigms, multi-value dependence, joint dependence, representation theory), relational database systems (focus on relational database theory, SQL and query optimization), database security and integrity constraints, database design, database technology development trends, etc.

8.6. Object-oriented analysis and design

This course mainly teaches the basic ideas and concepts of object-oriented, the whole process of object-oriented analysis, and the method of object-oriented system design for specific implementation conditions. The course contents mainly include: around the demand model (use case diagram), basic model (class diagram), auxiliary model (package diagram, sequence diagram, activity diagram and various other UML model diagrams) and model specifications in object-oriented modeling, introducing various object-oriented analysis methods and related UML content in detail; based on the analysis model, teaching object-oriented design methods, including problem domain design, human-computer interaction design, control-driven design, data interface design, etc., as well as system componentization and deploy.

8.7. Introduction to Software Engineering

This course mainly teaches the basic concepts, technologies and methods of software engineering and service outsourcing, including software development models, service outsourcing, feasibility analysis, demand analysis, software design, coding, testing, and maintenance; data structure-oriented analysis and design methods, object-oriented analysis and design methods; brief introduction to software reuse and rapid prototyping technology. The purpose of this course is to make students have a basic understanding of software engineering theories and methods, and to lay a foundation for subsequent special courses. Meanwhile, this course will cultivate students' ability to analyze and design actual software systems and the ability to use various popular software technologies to solve practical problems and control software quality.

8.8. Algorithm design and analysis

This course introduces common non-numerical algorithm design strategies in computer programming, with a certain depth and breadth. The course mainly teaches the concepts of time and space complexity of algorithms and their analysis methods, and students will learn typical algorithms such as greedy method, divide-and-conquer method, backtracking method, dynamic programming, branch and bound from the perspective of complexity analysis, so that they can master algorithms for solving some of the actual problems that often appear in computer application. Meanwhile, through the study of this course, students will master the basic principles and techniques of common algorithm analysis and design; have the ability to design and implement algorithms for given actual problems, and the ability to evaluate algorithms.

8.9. Software Project Management

This course introduces the basic theories, methods and techniques of software project management; including the main work content of the initial phase, planning phase, execution control phase and project end phase of a software project, basic principles, effective methods and related tools to be followed. Through the explanation of cases, students will master the methods and techniques of scope, schedule, cost, quality, risk, human resources, communication, configuration management and project execution control process in the software project management process; through course practice, students will be able to apply the theoretical knowledge they have learned to software project management practices.

8.10. Software Design and Development I, II, III

As the main line of engineering training projects, software design and development projects run through six semesters from the first grade second semester to the fourth grade first semester. Through the study of this course, students can have a comprehensive understanding from a series of theoretical basic knowledge of software engineering to professional knowledge of the realization of practical project applications. The main contents include: software interface design, prototype software design, requirements acquisition, database design, architecture design, code writing, testing, and mastering the basic methods and skills of software documentation. The difficulty of software development and design tasks will gradually increase, and students' ability to integrate theory with practice and comprehensive practice ability will be significantly improved, so that they will gradually have the ability to independently design and develop software.

Through this project, students can establish the concept of engineering project and cultivate their sense of innovation. Through the collaborative work of the competition team, students' communication skills and teamwork spirit will be cultivated. Through the summary and speech defense after the competition, students' speech ability and report writing ability can be exercised.

8.11. Website Architecture Project

The project organizes teaching and practice activities based on the course group, including: Web front-end development technology, Java programming, object-oriented analysis and design, Java Web development technology, principles and applications of website framework and other main courses. Through the study of this course, students will be able to use the knowledge and skills of these courses to solve specific problems in engineering practice related to the course group, so that theoretical teaching and engineering practice are closely integrated and mutually supported, and students' professional ability, communication ability, team spirit and leadership ability are trained.

The project is a Java Web development project. Students are required to conduct customer demand analysis, application development plan design, application program development and delivery with the background of participating in website design and development projects. The application requires a good user interface and the development requires using Java Web technology. Through the study of this course, students will be proficient in the use of HTML5, Javascript, JQuery&Ajax, Java Web website development and website framework technology, etc., so as to comprehensively improve and test their practical ability and professionalism in the practice of website design and development.

Assessment content: demand analysis (10%) + design (30%) + coding (40%) + topic defense (20%).

8.12. Database Development Project

The project organizes teaching practice activities in the unit of curriculum group, including: Web programming (.net), database design and application development, enterprise-level database management system practice and other main courses. Through the study of this course, students will be able to apply the knowledge and skills of these courses to solve specific problems in engineering practice related to the course group, so that theoretical teaching and engineering practice are closely integrated and mutually supported, and students' professional ability, communication ability, team spirit and leadership ability are trained.

Through this project, students will have a deep understanding of the concepts and database design methods in the database system, and can skillfully use the knowledge of six stages including database design requirements analysis, conceptual structure design, logical structure design, physical structure design, database implementation, and database operation and maintenance. The course focuses on design of the database structure, including the use of design tools, the design of stored procedures, triggers, training of T-SQL statements, and use of development tools to design simple database applications.

Assessment content: demand analysis (10%) + design (40%) + coding (30%) + thesis defense (20%).

8.13. Mobile terminal project

The project organizes teaching practice activities in the unit of curriculum group, including: Web front-end development technology, Java programming, human-computer interaction technology, mobile terminal software development and other main courses. Through the study of this course, students will be able to apply the knowledge and skills of these courses to solve specific problems in engineering practice related to the course group, so that theoretical teaching and engineering practice are closely integrated and mutually supported, and students' professional ability, communication ability, team spirit and leadership ability are trained.

The project is a mobile terminal application development project. Students are required to participate in the actual mobile terminal application development project as a background to conduct customer demand analysis, application development plan design, application development and delivery. The application requires a good user interface and the development requires the use of Android technology. Through the study of this course, the students will be proficient in the use of project development skills such as SDCard, Android SQLite, Content Provider, Android system services, etc., so as to comprehensively improve and test their practical ability and professionalism in the practice of mobile terminal software design and development.

Assessment content: demand analysis (10%) + design (30%) + coding (40%) + topic defense (20%).

9. Practical Training (Related Courses)

Website architecture project, mobile terminal project, database development project, software design and development I, II, III, data structure and algorithm internship, software quality assurance and testing internship, practice of database system, graduation internship, graduation design (thesis).

10. Course Structure and Course Hours (excluding extracurricular class)

Category	Total Credit	%	Total Course Hours	Theory Learning	Practical Training
Public Course	50.5	33	960	896	64
Basic Course	32	21	512	423	89
Professional Course	22	15	352	250	102
Practical Training	36.5	25	880	0	880
General Course	10	6	160	160	0
Total	151	100	2864	1729	1135
Theory : Practice(%)	60 :40				

11. Teaching schedule (1)

Category	Type	Provided by	Course Code	Course Name	Assessment	Credit	Course Hour	Theory Learning	Practical Training	Semester
General Education	Required	School of Marxism	b1080001	Basic principles of Marxism	test	3	48	42	6	spring 1
	Required	School of Marxism	b1080003	Ideological and moral cultivation and legal foundation	non-test	3	48	42	6	spring 1
	Required	School of Marxism	b1080006	Outline of Chinese Modern History	non-test	3	48	42	6	autumn 1
	Required	School of Marxism	b1080004	Introduction to Mao Zedong Thought and the Theoretical System of Socialism with Chinese Characteristics I	test	3	48	42	6	autumn 2
	Required	School of Marxism	b1080007	Introduction to Mao Zedong Thought and the Theoretical System of Socialism with Chinese Characteristics II	test	2	32	28	4	spring 2
	Required	School of Marxism	-----	Situation and Policy (Module 1～4)	non-test	2	32	28	4	autumn 1～spring 2
	Required	School of Marxism	b1080008	Labor Education A	non-test	0.5	16	16		autumn 2
	Required	College of Arts and Sciences	b1020080	Advanced Mathematics A1	test	4	64	64		autumn 1
	Required	College of Arts and Sciences	b1020081	Advanced Mathematics A2	test	4	64	64		spring 1
	Required	College of Arts and Sciences	b1020012	Linear algebra	test	2	32	32		autumn 2
	Required	College of Arts and Sciences	b1020013	Probability Theory and Mathematical Statistics	test	2	32	32		autumn 2
	Required	College of Arts and Sciences	b1020018	College Chinese	non-test	2	32	32		spring 1
	Required	College of Arts and Sciences	b1020063	College Physics A(Module 2)	test	3	48	48		spring 1
	Required	College of Arts and Sciences	b1020065	College Physics B	test	2	32	32		autumn 2
	Required	College of Arts and Sciences	b1020066	College Physics C	non-test	1	32		32	spring 1
	Required	Department of Physical Education	-----	Physical Education I～VI	non-test	3	160	160		autumn 1～autumn 4
	Required	Others	b1110003	Military skills	non-test	0.5	2W			autumn 1
	Required	College of Arts and Sciences	b1110002	Military theory	non-test	0.5	32	32		autumn 2
	★English	Module A	b1020003	General English III	test	3	48	48		autumn 1
			b1020004	General English IV	test	3	48	48		spring 1

Category	Type	Provided by	Course Code	Course Name	Assessment	Credit	Course Hour	Theory Learning	Practical Training	Semester
	(Selective 1 Module 10 credits)		b1020005	General Academic English A	test	2	32	32		autumn 2
			---	English development	non-test	2	32	32		spring 2
		Module B	b1020002	General English II	test	3	48	48		autumn 1
			b1020003	General English III	test	3	48	48		spring 1
			b1020006	General Academic English B	test	2	32	32		autumn 2
			---	English development	non-test	2	32	32		spring 2
		Module C	b1020001	General English I	test	4	64	64		autumn 1
			b1020002	General English II	test	3	48	48		spring 1
			b1020003	General English III	test	3	48	48		autumn 2
	★German	College of Arts and Sciences	b1020040	German I	test	3	48	48		autumn 1
		College of Arts and Sciences	b1020041	German II	test	3	48	48		spring 1
		College of Arts and Sciences	b1020042	German III	test	4	64	64		autumn 2
	★Japanese	College of Arts and Sciences	b1020077	Japanese I	test	3	48	48		autumn 1
		College of Arts and Sciences	b1020078	Japanese II	test	3	48	48		spring 1
		College of Arts and Sciences	b1020079	Japanese III	test	4	64	64		autumn 2
Total (General Education Basic Courses)						50.5	960	896	64	
General Course	Selective	Others	b0-----	Social Science and Humanities Literacy (4 credits) Natural Science and Technological Innovation (4 credits) Public Art (2 credits)	non-test	10	160	160		autumn, spring
Subtotal (general course)						10	160	160		

(★Note: The first foreign language has a total of 10 credits, including College English, German, and Japanese. Choose the appropriate language according to your needs; among them, if you choose College English, please choose the appropriate module in module ABC)

11. Teaching schedule (2)

Category	Type	Provided by	Course Code	Course Name	Assessment	Credit	Course Hour	Theory Learning	Practical Training	Semester
Basic professional courses	Required	College of Engineering	b2012179	Introduction to Software Engineering	non-test	1	16	16		autumn 1
	Required	College of Engineering	b2012162	Foundation of programming design	test	5	80	48	32	autumn 1
	Required	College of Engineering	b2012231	Data structure and algorithm	test	4	64	56	8	spring 1
	Required	Work training	b2090006	Fundamentals of circuit analysis	test	2	32	32		spring 1
	Required	College of Engineering	b2012258	Introduction to Database System	test	3	48	39	9	autumn 2
	Required	College of Engineering	b2012046	Principles of Computer Composition	test	3	48	42	6	spring 2
	Required	College of Engineering	b2012016	Operating system	test	2	32	28	4	spring 3
	Required	College of Engineering	b2012045	Computer network	test	3	48	42	6	autumn 3
	Required	College of Arts and Sciences	b1020022	Discrete mathematics	non-test	2	32	32		spring 2
	Required	College of Engineering	b2012105	Algorithm design and analysis	test	2	32	24	8	autumn 2
	Required	College of Engineering	b2012170	Object-oriented analysis and design	test	3	48	32	16	autumn 3
	Required	College of Engineering	b2012070	Introduction to Software Engineering	test	2	32	32		spring 2
	Subtotal (Basic professional courses)					32	512	423	89	
Professional courses	Required	College of Engineering	b2012259	Web front-end development technology	non-test	2	32	22	10	autumn 2
	Required	College of Engineering	b2012075	Software quality assurance and testing	non-test	2	32	20	12	autumn 3
	Required	College of Engineering	b2012014	Compilation technology	test	2	32	32		autumn 4
	Required	College of Engineering	b2012074	Software Project Management	non-test	2	32	24	8	spring 3
	Required	College of Engineering	b2012260	Software process management	non-test	2	32	32		autumn 4

	Required	College of Engineering	b2012006	Java programming (English teaching)	non-test	2	32	20	12	autumn 2
	Subtotal (required professional courses)					**12**	**192**	**150**	**42**	
	★Selective 10 credits (2 modules)	Module A	b2012261	Java programming (English teaching)	non-test	2	32	20	12	spring 2
			b2012262	Principle and Application of Website Framework	non-test	3	48	30	18	autumn 3
		Module B	b2012303	Intelligent interaction technology	non-test	2	32	20	12	spring 2
			b2012140	Mobile terminal software development	non-test	3	48	30	18	autumn 3
		Module C	b2012234	Web programming	non-test	2	32	20	12	spring 2
			b2012263	Database design and application development	test	3	48	30	18	autumn 3
	Subtotal (program module courses)					**10**	**160**	**100**	**60**	
Subtotal (professional courses)						**22**	**352**	**250**	**102**	

11. Teaching schedule (3)

Category	Type	Provided by	Course Code	Course Name	Assessment	Credit	Course Hour	Theory Learning	Practical Training	Semester
Practice	Required	Engineering Training Center	b4090002	Basic Engineering Training B	non-test	2	48		48	autumn 1
	Required	College of Engineering	b4012005	Programming design and practice	non-test	2	48		48	spring 1
	Required	College of Engineering	b4012038	Enterprise-level database management system internship	non-test	2	48		48	autumn 3
	Required	College of Engineering	b4000014	Innovation and Entrepreneurship in Software Engineering	non-test	2	48		48	spring 3
	Required	College of Engineering	b4012154	Software design and development I	non-test	1	24		24	summer 1
	Required	College of Engineering	b4012155	Software design and development II	non-test	2	48		48	summer 2
	Required	College of Engineering	b4012156	Software design and development III	non-test	2	48		48	summer 3
	Required	College of Engineering	b4012157	Data structure and algorithm internship	non-test	3	72		72	summer 1
	Required	College of Engineering	b4012054	Database system course practice	non-test	2	48		48	summer 2
	Required	College of Engineering	b4012048	Software quality assurance and testing internship	non-test	2	48		48	summer 3
	Required	College of Engineering	b4012186	Labor Education B	non-test	0.5	16		16	spring 3
	Required	College of Engineering	b4012045	Graduation Practice and Graduation Design (Thesis) of Software Engineering	non-test	12	288		288	spring 4
	Subtotal (required practice courses)					32.5	784		784	
	★ Professional module, selective 4 credits	Module A	b4012158	Website Architecture Project	non-test	2	48		48	spring 3
		Module B	b4012159	Mobile terminal project	non-test	2	48		48	spring 3
		Module C	b4012160	Database development project	non-test	2	48		48	spring 3
	Subtotal (practice module)					**4**	**96**		**96**	
Subtotal (professional practice)						**36.5**	**880**		**880**	
Extracurricular Class	Required	Others	b5110001	Extracurricular Class	non-test	1	-	-	-	autumn, spring, summer

Total		152	2864	1729	1135	

★1.Guidance for module professional courses and module practical courses:

Selective module description: Choose 2 modules. The professional practice module must be selected according to the corresponding professional course module.

1. Module A: Java Web Application Development

The direction of module A focuses on in-depth learning of Web front-end development technology, Java Web website development technology and website framework principles. Through the study, students will obtain the ability to develop web front-end, website development and maintenance, testing and software project management, and cultivate familiarity with website development technology. This module will cultivate comprehensive technical personnel who can carry out related program development and software technical support.

2. Module B: Mobile Application Development

The direction of module B focuses on in-depth learning of human-computer interaction technology and mobile terminal software development, and cultivates the ability to develop web front-end, Android-based mobile application development, testing, and software project management, so that students will be able to carry out related program development and software technical support.

3. Module C: Database Development

The C module direction focuses on in-depth study of Web programming (.net), database application design and development, and cultivates comprehensive technical talents with capabilities such as data model design, database-side programming and interface design, testing, and software project management, and the ability to develop related database systems and software technical support.

2. Professional Certificates can be gained after learning following courses:

Students who have obtained the computer technology and software professional technical qualification (level) examination (referred to as the soft test) qualification certificate of junior programmer, software designer/software tester, can apply for exemption from Software Project Management, Software process management, Compilation technology, comprehensive training of service outsourcing, Innovation and Entrepreneurship in Software Engineering (choose one) course and obtain corresponding credits.

Students who have obtained the CCF CSP exam qualification certificate can apply for exemption from data structure and algorithm course practice, data structure and algorithm, algorithm design and analysis, innovation and entrepreneurship in software engineering (choose one) course and obtain corresponding credits.

12. Prerequisite for Course Study

No.	Course name	Prerequisite Course
1	Data structure and algorithm	Foundation of programming design
2	Operating system	Data structure and algorithm
3	Introduction to Database System	Data structure and algorithm
4	Object-oriented analysis and design	Data structure and algorithm
5	Introduction to Software Engineering	Data structure and algorithm
6	Software Project Management	Introduction to Software Engineering
7	Software quality assurance and testing	Introduction to Software Engineering
8	Compilation technology	Discrete mathematics Data structure and algorithm
9	Algorithm design and analysis	Discrete mathematics Data structure and algorithm Probability Theory and Mathematical Statistics
10	Software process management	Object-oriented analysis and design Software Project Management Software quality assurance and testing
11	Database design and application development	Introduction to Database System Database system course practice Web programming
12	Web front-end development technology	Data structure and algorithm Computer network and communication
13	Java programming (English teaching)	Web front-end development technology Java programming Object-oriented analysis and design
14	Principle and Application of Website Framework	Java programming (English teaching)
15	Mobile terminal software development	Web front-end development technology Java programming Human-computer interaction technology

13. Extracurricular Class

Through taking extracurricular classes, students are encouraged to take part in academic lectures, social practice activities, campus cultural and sports activities, innovative and entrepreneurial activities, voluntary activities, etc. to improve their social adaptability and enhance the competitiveness in the job market. Details are specified in Students' Manual.