

A0L033Q C Language Programming

(3 Credits, 48 Hours; Course Category: General Public Course/Engineering Fundamental Course; Specialty: Software Engineering; Prerequisite: None)

As the first basic course in the software engineering major, Programming Fundamentals provides the basic knowledge on computers and programming to lay a solid foundation for the freshmen to go on with the related study in the major.

The course is instructed by both English and Chinese. Through the study in the course, the students will pick up the basic knowledge of the structure of hardware and software, understand the computer dynamic mechanism and master the related concepts on programming. They will acquire the methods of structured programming, practice the skills of programming and observe the programming norms. They will grasp the data types, operators, expressions, control structures, functions, arrays, pointers, structures, files and linked lists of C language programs. They will master the frequently used algorithms like recursion, enumeration, find and sort, and be able to implement them with C programming language. Students' ability in reading and understanding English documentation on program design will be fostered. They will also be able to debug program independently, thus possess certain ability to solve practical problems.

A0L128Q Software System Analysis and Design

(2 Credits, 32 Hours; Course Category: Major Course; Specialty: Software Engineering; Prerequisite: Object-Oriented Programming and Design, Database System, Data Structure)

The key issues in software development are those on how to identify the problems, how to understand them in order to clearly define the requirements of business, the scope and objectives of the project or system, and develop the technical solution to solve the problems to better support business. Unfortunately, there is a huge gap in mutual understanding between the business staffs and technical experts especially at present with the concept that technology should serve business and under the situation that both business and technology are becoming more and more complicated and complex. In this case, the essence and effective software engineers are required to cultivate the qualities involving the vision of deep insight of business and technology as well as the skills of communication and collaboration. They are supposed to possess the abilities of finding problems and making solutions based on the framework of information system in the structure of component organization and the process of abstraction instantiation.

This course belongs to the National Excellent Courses. It plays an important role in the education of software engineering majors. As one of the main required courses, it aims at cultivating talents with innovation consciousness and the abilities of developing and designing the software systems. It lays a solid foundation for the future careers like information consultant, software system analyst and software designer. It

provides high quality education service to junior students focusing on the following aspects.

1. Education in expanding the vision. By exploring the contexts in which the information system locates, students will discuss the influences of software to society, economy, business, technology, as well as our daily life, thus build up the background on looking for problems and seeking for necessities.
2. Education in deepening the insight. By learning the concepts on software and its development framework, the students accumulate the basic knowledge to conceive and design the software system of high quality.
3. Education in software engineering experience practicing on methodology and process. By cooperation and teamwork, the students realize the methods to develop effective software with anticipated outcome through abstraction, deduction, division and governing. They pick up the abilities to design by themselves and to select from different models and get the final solution with a trade-off among multiple candidates.
4. Training in the abilities of implementation. By effective communication and collaboration, the students work together to organize and manage projects effectively in order to achieve their goals. In addition to this, they practice the ability to transfer the value of the software perfectly to the customers.

A0L158Q Software and Intellectual Property law

(1 Credit, 16 Hours; Course Category: Specialty Elective Course; Specialty: Software Engineering; Prerequisite: None)

Software and Intellectual Property Law is a specialty elective course for master of software engineering, beneficial to deepen the software engineering students' awareness on software and intellectual property law, and accumulate the legal knowledge of software. Students successfully completing this course will master the basic knowledge of intellectual property law, Chinese current legislation of patent, copyright, trademark, computer software, commercial secrets and other basic knowledge about unfair competition legal system. With the above knowledge, students can comprehensively improve their recognition level of software and intellectual property right, and meet the quality requirements of general software development practice in the future.

A0L168Q Software Project Management

(2 Credits, 32 Hours; Course Category: Compulsory Course; Specialty: Software Engineering; Prerequisite: Introduction to Software Engineering)

This course is a compulsory course for undergraduate students whose major is software engineering. It gives the students a general introduction to software

engineering management. The course covers relevant management fields and the relations among them. The key points include the introductions to the targets, activities, processes and key work products in the relevant fields. The students are supposed to have an overall idea and knowledge structure about software engineering management, which helps to build the foundation for their later study or work.

The major contents of the course include the analysis of the characteristics of software engineering management projects and the key factors leading to their success or failure. Through the introduction of the key processes and activities in software engineering management, the course focuses on the introduction of how to plan, evaluate and optimize the scheduling as well as some related technique such as the common software in the field: DEPHI, FP evaluation model, CPM, PERT and so on. It also covers the content of project tracing and control, together with other related software management like risk management, necessity management and configuration management.

A0S176Q Comprehensive Practice on Database Application System

(2 Credits, 32 Hours; Course Category: Specialty Basic Course; Specialty: Software Engineering; Prerequisite: Programming Language and Data Structure)

Comprehensive Practice on Database Application System is a core course of software engineering and other related disciplines, and it's a comprehensive practical course in fifth term. The course is designed for those students who have certain programming and development capabilities. Students should understand basic principles in operating system and database system, and they should also master the core algorithms in these two courses, grasp a database management system application(DBMS), design and implement a web database application (contract management system).

This course includes thematic design and practical project two sections. The first section consists of several basic course practicing on database system and operating system; and the second section is an enterprise project (contract management system) which requires students to design and implement a database application based on web, then hand in its English experiment report and finish the performance for the project.

A0L235Q Introduction to Software Engineering Major

(1 Credit, 16 Hours; Course Category: Specialty/Specialty Basic Course; Specialty: Software Engineering; Prerequisite: None)

“Introduction to Software Engineering Major” is a specialty course for software engineering students. As an introductory course for software engineering students, which covers the basic concepts, history of software and software engineering, software engineering knowledge system and ability and quality requirements, software engineering professional training model, software industry development, career planning, software companies experience and exchange etc. By the study of this course,

students can better understand the software engineering, including the basic concepts and major situations, and thus provide guidance for the follow-up of the professional learning and self-learning development plan.

The course is based on "class teaching + technical report + industry experience + project presentation" as the main teaching means, and assessed by assignments and presentation.

A0L237Q Object-Oriented Programming and Design

(2 Credits, 32 Hours; Course Category: Specialty Major Course; Specialty: Software Engineering; Prerequisite(s): C Programming Language)

This course is a specialty major compulsory course of software engineering. The course is practical and innovative. The course has a total of 32 lecture hours, and it is accompanied by a corresponding practical Course.

This course reaches the main technology and programming ideas of object-oriented programming, including the introduction of Java language, object-oriented programming concepts and programming techniques, exception handling, input / output streams, collections framework, multi-threaded, graphical user interface, network programming and other content.

By learning this course, students can understand the core ideas of object-oriented programming, master the main technology and programming ideas of object-oriented technology, lay a good foundation for the future development of large and complex enterprise projects, and provides powerful object-oriented programming support for the follow-up specialty courses.

A0L238Q Data Structure

(2 Credits, 32 Hours; Course Category: Basic Course; Specialty: Software Engineering; Prerequisite: Programming Fundamentals, Discrete Mathematics)

Data Structure is a compulsory basic course for students of software engineering professionals. This course provides foundation for other courses such as Database System, Operating System, Software Engineering, etc. To design an algorithm of high-efficiency, we must study the characteristics of data as well as the relationship between them and the corresponding storage representations. Data structure is a discipline that will study computer operation objects of non-numerical computation problems as well as their relationship and operation.

This course introduces a variety of commonly used data structures, such as the linear lists, stacks, queues, strings, arrays, trees, binary trees, and graphic and other basic types of data structures and their application scopes, as well as a variety of algorithm of sort and search, focusing on time and space analysis and comparison.

This course has enhanced the practical training, including the training on algorithm design and comprehensive problem analyzing, thus improved the utility of the experiments. Through the learning and training of this course, students will understand

the basic concepts of data structures, master the logical relationship, storage representations as well as operations and application of various data structures. They will enhance the ability of reading information in English, the ability of independent problem analyzing and solving. They will strengthen the capabilities of coding and cooperation work so as to lay a solid foundation to participate in innovative research and software developing work in the future.

After completing this course, students will be able to master

(1) the logical relationship, storage representation as well as operations and application scope of various data structures, include linear list, stack, queue, tree, graphic, set etc.

(2) the characteristic and performance of various research methods,

(3) the characteristic and performance of various sorting methods,

(4) the analysis of time and space complexities and comparison of various algorithm,

(5) the ability of analyzing and solving problems independently, and cooperation with teammates.

A0L240Q Operating System

(2 Credits, 32 Hours; Course Category: Specialty Basic; Specialty: Software Engineering; Prerequisite(s): Programming Fundamentals, Data Structure)

Operating System is a basic course for software engineering students, and also a specialty course for software engineering major. The course include much theory, which contains the basic structure and theory of the operating system, design ideas and methods, mainly the basic concepts of operating system, principles and implementation of technology. The teaching content include processor management, process management, memory management, file management, input and output systems etc. Through the study of this course students can better understand and master the basic principles of operating system, structure and design methods, understand the interaction between users and computer systems and the operating process of the operating system. The course will lay a solid foundation for the follow-up of the further specialty courses learning and research work

A0L241Q Database System

(2 Credits, 32 Hours; Course Category: Specialty Basic Course; Specialty: Software Engineering; Prerequisite(s): Programming Language, Data Structure)

Database System is a core course for sophomores majoring in Software Engineering and other related disciplines. It aims to cultivate students' basic concepts, modeling ideas and standardized design methods of database systems to understand the

latest developments in database technology. So as to have the ability to design and implement the database application system independently, enhance the ability to run and maintain the database system, enhance the ability to retrieve English technical literature and read English professional and technical information, improve the international communication level and the ability of autonomous learning advanced technology, students will be trained to become an internationally competitive elite software talent.

Through the study of this course, students understand and master the basic principles of database system: including the basic concepts of database, the characteristics of various data models, the basic concepts of relational database, SQL language, relational data theory, database design theory; Master the design and development methods of database application system, understand the main content of database technology and development trends; The aspect of practice: requires students to use the principles of database knowledge and practical tools to develop a database application system. The ultimate goal is to develop students' ability to use database knowledge to analyze problems, solve problems, improve students' ability to communicate and express and the ability to collaborate and learn independently.

A0L242Q Computer Network

(2 Credits, 32 Hours; Course Category: Specialty Major Course; Specialty: Software Engineering; Prerequisite: Discrete Mathematics, C Programming Language)

"Computer network" is a major course of software engineering. Through this course, the students will be aware of the basic concepts of computer network, and acquire the in-depth understanding of the basic working principles of network and protocol designs. They will master the computer network architecture and standards, the network practical techniques, and be familiar with a variety of network technology, as well as the basic principles of network management, network security, and the related methods and technologies. The students will have the knowledge and skills in the analysis and design of computer networks. These preliminary studies will provide a good foundation for their future research and courses.

The main content of this class includes all kinds of computer network concepts and techniques (the physical layer, data link layer, network layer, transport layer, application layer and specific technology); Use and design of network communication protocols; Data distribution in the network, etc.

A0L243Q User Interface Design and Evaluation

(2 Credits, 32 Hours; Course Category: Elective Course; Specialty: Software Engineering; Prerequisite: Programming Fundamentals)

As an elective course for students majoring in software engineering, the course

describes how to design user friendly interface and how to evaluate and enhance the usability of these interfaces. From the course, the students will grasp the related theories on the design of the interfaces and learn to create the interface with a rapid-prototyping programming language. They will learn to design the user friendly interface of high usability and master the major techniques to evaluate the usability of interfaces.

As an essential course to successful software design, this course combines the components of programming interactive user interfaces with the methods to evaluate and improve the usability of those interfaces.

This course consists of four parts—the basic theories for interface design, a rapid-prototyping programming language, interface design and interface evaluation.

Students successfully completing this course will be able to design and create usable, user-friendly interfaces, to evaluate critically the usability of user interfaces (including those that are not computer-based) and to report their findings in a systematic way.

A0L244Q Algorithm Design and Practice

(2 Credits, 32 Hours; Course Category: Specialty Basic Course ; Specialty: Specialty Elective ; Prerequisite(s): Higher Mathematics, Programming Fundamentals, Data Structures)

This course is a basic specialized which designed for undergraduates majoring in software engineering. The algorithm plays an important role in computing science and computing practices. The purpose of this course is to explain the solution of practical problems for computer application, and teach the basic principles, methods and techniques of algorithm design and analysis, and cultivate students' ability of analysis and understanding for algorithm complexity. Firstly, it introduces the definition of computational complexity and basic methods of algorithm analysis, according to the representative non-numerical problems in the fields of computer science and software engineering, then it introduces several important algorithm design methods, which are divide and conquer method, dynamic programming method, greedy method, retrospective search method, branch and bound method and algorithm design practice and so on. It can both enable students to grasp various algorithms and grasp the basic methods and techniques of algorithm analysis. Through the teaching and learning of the course, the students are expected to:

1. Grasp the basic theory of algorithm analysis and design;
2. Grasp the basic methods of algorithm analysis and design;
3. Grasp non-numerical algorithms in field of computer and learn how to use these algorithms to solve practical problems.

A0L245Q C++ Programming

(2 Credits, 32 Hours; Course Category: Specialty Basic Course; Specialty: Software Engineering; Prerequisite: Object Oriented Concept, Programming Fundamentals,)

It is one of the most widely used programming languages. You find C++ applications everywhere from the bottom of the oceans to the surface of Mars. It is good for coding back end and front end applications with multilayered architecture, in real time environment. It is an object oriented programming languages, and it directly supports the key concepts and techniques used in real-world software. This course combines a component that teaches programming using C++ language with one that teaches developing Windows Programming and Winsock Application.

This course is organized into four parts – C++ compatible programming in C, the syntax and usage of C++ programming language, OOAD in C++, windows and MFC programming.

Students successfully completing this course will be able to set workbench settings, and create console projects, earn enough knowledge for pure OOAD, familiar with C++ STL library, know how to use Exception, Namespace and standard thread, can develop full featured Window and MFC Applications, develop skills to undertake projects, other languages, and software subjects.

A0L248Q Software Architecture

(2 credits; 32 hours; Course Category: Required Course; Specialty: Software Engineering; Prerequisite: System Analysis and Design, Object-Oriented Design and Programming)

The course is a compulsory course for Software Engineering program. The motivation is to make students to further study the ideas and methods of modern software architecture design and be able to grow into excellent software system architects in five to eight years after their graduation.

The course mainly comprises of three parts, the ideas of software architecture design, the principles of object-oriented design, and the design patterns. In the course, the students need to learn the basic concepts of software architecture, software architectural style, principles of object-oriented design, design patterns, software product line and quality attributes. During the course, the students are required to grasp the knowledge and methods of software design, and apply the theory and methods to solve practical engineering problems based on to the actual requirements of the projects.

A0L249Q Software Testing Technology and Practice

(2 Credits, 32 Hours; Course Category: Elective Course; Specialty: Software

Engineering; Prerequisites: Object-Oriented Programming and Design, Software Process)

Software testing is an investigation conducted to provide the information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. In large software enterprises, the number and specialties of testers are larger than the developers, and the technical requirements of the testers are more diverse.

The purpose of this course is not only to cultivate qualified testing personnel for the software development team, but also to build up students' quality senses so as to let them adapt to different roles in future careers. The curriculum of the course is to let the students understand the concepts and definitions of software quality assurance and software testing first. Then, according to the typical test activities in the life cycle of software development, the course helps the students to master the testing skills, quality assurance workflows, and grasp all kinds of testing technologies based on the different testing types.

The contents of the course include different testing types and testing technologies (static testing, dynamic testing, unit testing, integration testing, system testing, regression testing, performance testing, security testing and localization testing, etc.), testing case design methodology (black box and white box), testing management (defect management, test case management), testing and project management, test driven development approach, and test automation development, etc.

A0L251Q Information Security – Principles and Practice

(2 Credits, 32 Hours; Course Level: Specialized Course; Specialty: Software Engineering; Prerequisite: Discrete Mathematics, Operating System, Computer Network)

This course is a professional course of software engineer. This course will include an appreciation of the general nature of: encryption techniques for providing confidentiality services (including stream ciphers, block ciphers and public key techniques), mechanisms for providing data integrity and origin authentication, including MACs and digital signatures, message exchanges to provide entity authentication and/or key establishment, and the use of Trusted Third Parties, such as Certification Authorities (CAs), to provide and support Public Key Infrastructures, computer network security and system security, information hiding and digital watermarking technology, software, security methods and technologies.

Students will learn the basic concepts and technologies in Information Security.

After taking this course, the student should:

(1) master the basic knowledge of information security, information security model, the main techniques of cryptography, digital signature, authentication technology, computer network security, system security, information hiding, digital

watermarking technology, software security, and relevant laws and regulations of information security technology, etc.;

(2) understand the trend of network and information security;

(3) establish the theory, the model and the methods of designing secure information systems; increase the interests of information security; and eventually have the basic consciousness and ability to analyze and solve secure problems;

(4) improve the ability of English reading and self-learning.

A0L252Q JavaEE Frameworks and Application

(2 Credits, 32 Hours; Course Category: Elective Course; Specialty: Software Engineering; Prerequisites: Object-oriented Programming and Design, Database System, JSP, JDBC, Servlet)

The course is an elective course targeted for juniors majoring in Software Engineering. With the knowledge from the previous courses, the students will go on learning the development techniques of JavaEE framework and the JavaEE Web application design and acquire the ability of developing the application systems based on the JavaEE framework.

The kernel of the course is Java EE Web application development. It focuses on Struts2 Framework, Hibernate Framework, and Spring Framework. After finishing the course, the students are expected to reach the following goals.

- To develop JavaEE Web applications.
- To understand software design ideas and principles, the architectures of JavaEE frameworks, and apply them to the actual project development.
- To master software architecture patterns and design patterns, and flexibly to apply them to the development of software projects.
- To deeply comprehend the multilayer architecture and loose coupling principle advocated by JavaEE platform and to apply it in the development of software projects.

A0L253Q Mobile Application Development

(2 Credits, 32 Hours; Course Category: Specialty Elective Course; Specialty: Software Engineering; Prerequisite: none)

This course focuses on real project practice; students will be familiar about Android system and master Android development skills through real demo lecture and exercises.

The target of this course makes students familiar with Android platform, master key knowledge; get fundamentals of Android application development. After this course, students will master the usage of Android IDE environment, controls and

properties.

This course is divided into two parts - the basic theory lecture for Android, and the real android project demo and exercises. The theory lecture part including: setup Android development environment, Android fundamental, basic components of Android, Android UI, data storage, process and thread and picture animation programming. During the project demo and exercises, students will emulate the real project development to create electric commercial application. Handle real debug and developing skills through project demo development.

A0L254Q Linux System and Network Programming

(2 Credits, 32 Hours; Course Category: Specialty elective Course; Specialty: Software Engineering; Prerequisite(s): Programming Fundamentals, Operating System)

"Linux operating system and network programming" is both theory and practice course. Experiment is an important part of the experimental teaching of the course. Through the experiment, students can find out the basic concepts of Linux operating system, and get more understanding of the basic functions. The task of this course is to enable students to master the Linux system, and master the Linux environment programming, related to the programming technology which is mainly Shell programming, kernel, driver and network C language programming. The construction of the operating system kernel, such as the construction of the operating system kernel, the driver architecture, the graphics subsystem, the file subsystem, the control subsystem, the network subsystem and so on, lay a good foundation for the future development work related to Linux.

A0L256Q Data Mining and Data Analysis

(2 Credits, 32 Hours; Course Category: Specialty Basic Course; Specialty: Software Engineering; Prerequisite(s): Programming Fundamentals, Statistics, DBMS)

Along with the widespread use of IT techniques, many companies have collected huge business related data from their everyday activities. It is more and more important to distill valuable information to help business decision. Therefore data analysis (Data Analytics) is becoming more and more popular and important. The requests for qualified data analysts are then becoming indispensable for business enterprises to ensure their competitive power and advantage, which should be definitely helpful to conduct the course design for students' later growth.

This course tries to expand the abilities of juniors majoring in software engineering. It is necessary for them to master concepts and techniques closely related to practical implication. From the instruction of the course they will master the basic

knowledge of data mining and analysis, frequently used methods and models to enhance the students' hand-on ability and innovative ability. Aiming to introduce the basic concepts and techniques of data analysis, the course covers many topics including the models, algorithms and techniques for users to understand the business, the data, and make appropriate decisions based on the data.

The first part of the course is focused on demonstrating the business background based on the MBA branches, from which many business questions are concluded and will be the hints to covers the methods scattered in later parts. The second part of this course distills related methods from statistics and data mining, which aim to discover valuable knowledge from the data. The third part is the introduction of popular software used now in real applications to data analysis, such as Data Warehouse, OLAP (On-Line Analysis Processing), etc.

To ensure the ability of data analysis after this course, many manual computation questions will be used during the teaching. And many practices will be proposed as projects for students to use given data and software (Python, R, SPSS, Excel, etc.) for data analysis. Students are required to finish a document for each project to demonstrate their understanding of the related algorithms, and data processing procedure. By those assignments, students can not only understand the related concepts, algorithms and techniques of data analysis, but also master the necessary skills to carry out data analysis, like using software, and writing documents.

A0L268Q Discrete Mathematics

(2 Credits, 32 Hours; Course Level: Required; Specialty: Software Engineering; Prerequisite: Basic Calculus(B)I, Geometry and Algebra(B), Basic Calculus(B)II, Basic Computer Calculus Programming)

Discrete mathematics is the foundation of computer science and software engineer. This course provides an overview of the branch of mathematics commonly known as discrete mathematics. It is intended for Computer Science and Software Engineer majors.

This course has two major goals: to learn certain material fundamental to computer science and software engineer, and to increase students' sophistication and ability in handling abstract problems.

The topics covered are sets, logic, relations, functions, proofs, induction, trees, and graphs.

Students will learn the essential mathematic concepts and ideas in discrete mathematics, which are required for rigorous studies in most areas in computer science and software engineer. Our objects in the source are as follows.

(1) Enable students to accept the modern view of discrete mathematics, and to master the basic concepts, theories and certain applying skills of propositional logic, predicate logic, sets, relationships, functions, graphs, trees, and other models.

(2) Train and improve students' skills and abilities of abstracting, logical reasoning, writing rigorous proofs, analyzing and solving problems using discrete mathematical models.

(3) Improve the students' ability of English reading and self-learning. Lay a solid foundation for students to learn the successive courses (such as, algorithm design and practice, operating system, database system, and information security - theory and practice) and to take or participate in innovative works.

(4) Enable students to understand the process and the basic mode of problem solving, which is not only the most interesting part, but also the most important part of this course.

A0L323Q Introduction to Software Engineering

(2 Credits, 32 Hours; Course Category: Required Course; Specialty: Software Engineering; Prerequisite(s): C Programming Language)

“Introduction to Software Engineering” is the core course for students majoring in Software Engineering. The course doesn't aim to teach in detail some techniques or tools for software development, but will focus on the systematic knowledge and methodology in software engineering by introducing the actual cases. Through this course, the students will be equipped with the macroscopic and systematic views on software engineering. They will learn the major knowledge and structure, and understand software engineering from system and engineering perspectives. The main contents cover in the course is as follows: (1) Concepts on software and software engineering; (2) Software process and modeling from the view of software system development; (3) Software quality management from the view of quality assurance; (4) Software project management from the view of projects. The students will learn and understand the knowledge structure on software engineering, use modern software modeling methods, develop software system by using advanced platform and tools, build the good system and engineering awareness, grasp the engineering principle (including technique, method, tools and environment) to develop and manage software system. The course lays a solid foundation for sequential specialty courses and the systematic engineering analysis and design in the real world project development.

A0L328Q Principles of Computer Organization

(2 Credits, 32 Hours; Course Level: Elective; Specialty: software engineering; Prerequisite: A0L236Q)

The course explains a single computer principles and internal operating mechanism, including arithmetic unit components, controller parts, memory subsystem, input / output subsystem (bus interface), input / output system device. Knowledge are introduced for each part, which includes the function, composition, design and implementation. Through the study of this course, students should to be able to understand the general computer Organization, the principle and internal operation

mechanism, and build up the foundation for learning the professional successor courses and hardware-related technical work.

A0L342Q Web Developing Technologies

(2 Credits, 32 Hours; Course Category: Elective Course; Specialty: Software Engineering; Prerequisite: Programming Fundamentals, Introduction to Software Engineering, Computer Network, Database Systems)

This is one of the elective courses for students majoring in software engineering. The course focuses on the basic knowledge and the mainstream technologies of web developing, aiming to help students with basic web programming skills. The objectives of the course includes: understanding the basic knowledge of web applications; mastering the mainstream technologies of web developing; and mastering the theoretical and technical methods for web developing.

The course covers the following fields: the basic knowledge of web applications, front-end developing technologies of web (such as HTML, DHTML, xHTML, XML, JavaScript, CSS, AJAX, HTML5), server-side developing technologies of web (i.e. PHP), and mainstream developing framework of web (i.e. PHP developing framework). In addition, the course also covers the knowledge of web developing tools and the environment for developing/testing/running web applications. Finally, some typical web applications will be analyzed in the course.

After the course, the students can lay a solid foundation for the development of complicated web applications.

A0L343Q Product Analysis, Design and Operation

(2 Credits, 32 Hours; Course Category: Specialty Basic Course; Specialty: Software Engineering; Prerequisite(s): Fundamental of Software Engineering)

This course focuses on the analysis, design and operation of products. It introduces the primary concepts, methodologies and tools for product analysis, design and operation. Instead of lecturing, this course will allocate more time for the students to practice through laboratory researches. Students are given actual projects to practice under the guidance from our professor. Laboratory researches are one of the most efficient way for the students to learn and practice. In the meantime, our professor will share many cases in reality from his/her personal experiences in the industry. This course will start from the product analysis, then the product design, and present the product operation as the most important content in the end. Our professor will also give several lectures, guide the student through the lab, and provide opportunities for students to summarize and present their results and achievements at the end of this course.

A0L344Q Non-relational Database

(2 Credits, 32 Hours; Course Category: Specialty Course; Specialty: Software Engineering; Prerequisite(s): Programming Fundamentals and Database

Fundamentals)

Within the rapid development of computer technology, especially with Internet and big data, traditional relational database can not satisfy the requirement of the large scale data processing. There's the big challenge of how to deal with huge amount of non-structured data.

In this class, we will introduce the database design and query for semi-structured data and unstructured data, by the real use case in enterprises.

Practices is designed to let the students to manage modern non-relational database systems. The scenarios, requirements and data are all from real use case of enterprises.

A0S004Q Comprehensive Practice Of Programming Design

(2 Credits, 32 Hours; Course Category: General Public Course/Engineering Fundamental Course; Specialty: Software Engineering; Prerequisite: C Language Programming)

This is one of the practical courses for students majoring in software engineering. It focuses on the practice in computer applications and programming, aiming to help the students with the basic knowledge in computer science to learn initial programming.

The objectives of the course include: comprehensive application of computer science and programming skills to solve practical problems; learning to use modular programming design, the top-down programming method, and incremental development model to solve practical problems; using C language and related IDE in programming and debugging skillfully and independently; writing software project documentations in accordance with the engineering thinking; experiencing the team development process.

This course is divided into two parts. The first part is the integrated curriculum design, and the second part is the real enterprise projects.

Through the course, the students can lay a good foundation for the study of the subsequent software engineering specialized courses.

A0S174Q Practice in Object-oriented and Interactive Application Development

(2 Credits, 32 Hours; Course Category: Specialty Major Course; Specialty: Software Engineering; Prerequisite(s): C Programming Language, Object - oriented programming and design, Introduction to Software Engineering)

This is one of the integrated practical courses for students majoring in software engineering. The course focuses on the practice in GUI and object-oriented design and programming, aiming to help the students with basic programming skills. The objectives of the course includes: mastering the object-oriented analysis, design and implementation technologies; and learning to use object-oriented technologies and interactive application design and development technologies to develop the interactive

applications with graphic user interface. The students are required to comply with the standardized software development process including analysis, design, coding, debug, test and submission, and to complete interactive GUI applications in accordance with mission requirements, prepare the documents of the software project, finally complete the project with presentation. This course is divided into three parts. The first part is course experiments, the second the integrated curriculum designs, and the third the real enterprise projects. After the course, the students can lay a solid foundation for the development of large-scale and complicated enterprise software applications.

A0S175Q Comprehensive Practice of Data Structure and Algorithm

(4 Credits, 64 Hours; Course Category: Specialty Course; Specialty: Software Engineering; Prerequisite: Programming Fundamentals, Practice in Programming and Computer Science Discrete Mathematics, Object-oriented Programming and Design, Practice in Object-oriented and Interactive Application Development, Data Structure, Algorithm Design and Practice)

This is one of the comprehensive practical courses for students majoring in software engineering. The course focuses on the practice in algorithm design and application for the students with basic programming skills.

The objectives of the course are to help the students understand the core idea of data structures and algorithms, to master the logical relationship and storage structure of data structure, the idea and applications of algorithm, and the basic theory and application skills of discrete mathematics, As a result, the students have the ability to analyze problems using discrete mathematical models, to store data using data structures, and to solve practical problems using algorithms.

By the end of this course, students will be able to:

- (1) Master the logical structures and physical storage structures of data, algorithm design and implementation technologies;
- (2) Master the discrete mathematics analysis models that are used to analyze problems and using algorithm to solve the problems;
- (3) Comply with the standardized software development process including analysis, design, coding, debug, test and submission;
- (4) Complete algorithm applications in accordance with requirements;
- (5) Write software project documents in accordance with the engineering thinking;
- (6) Experience the team development process;
- (7) Complete the project and show their works by presentation.

After the course, the students will lay a solid foundation for the development of large-scale and complicated enterprise software applications.

A0S178Q Comprehensive Training of System Conception and Design

(2 Credits, 32 Hours; Course Category: Required Course; Specialty: Software Engineering; Prerequisite: Software Analysis and Design Technology)

The value of software is determined by the value that the software brings to its stakeholders through the way of problem solving. So, the software designer, as a problem solver and key role player in the industry, it is of great importance of his (her) level to make the insight of the problems and the abilities to make software solution by which the value can be realized through business or product improvement or innovation.

The present course, as a complementary part of the course “Software System Analysis and Design”(A0L128Q), aims at enhancing the students’ abilities of software conceptions based on problem finding and necessity analysis as well as software design based on the specific definition of software requirements. It is training in the comprehensive application of software engineering methods and their processes, and also a practice of problem-solving in the real world.

A0S181Q Practice in Object-oriented and Interactive Application Development

(2Credits, 32 Hours; Course Category:Specialty Major Course; Specialty: Software Engineering; Prerequisite(s): C Programming Language, Object - oriented programming and design, Introduction to Software Engineering)

This is one of the integrated practical courses for students majoring in software engineering. The course focuses on the practice in GUI and object-oriented design and programming, aiming to help the students with basic programming skills. The objectives of the course includes: mastering the object-oriented analysis, design and implementation technologies; and learning to use object-oriented technologies and interactive application design and development technologies to develop the interactive applications with graphic user interface. The students are required to comply with the standardized software development process including analysis, design, coding, debug, test and submission, and to complete interactive GUI applications in accordance with mission requirements, prepare the documents of the software project, finally complete the project with presentation. This course is divided into three parts. The first part is course experiments, the second the integrated curriculum designs, and the third the real enterprise projects. After the course, the students can lay a solid foundation for the development of large-scale and complicated enterprise software applications.

A0S Comprehensive Case Studies Courses II

(2 Credits, 2 Weeks; Course Category: Specialty Practical Course; Specialty:

Software Engineering; Prerequisites: C Programming Language, Object-oriented Programming and Design, Introduction to Software Engineering, User Interface Design and Evaluation, Data Structure, Discrete Mathematics, Practice in Programming, Practice in Object-oriented and Interactive Application Development, Practice in Data Structure and Algorithm)

This is one of the integrated practical courses for students majoring in software engineering. The course adopts the enterprise training and practical mode. Through the introduction of real software projects, the real environment and the role of enterprises to develop students on the basic theory of software engineering, design and development skills, the comprehensive ability to analyze and solve the problem. Improve the engineering capacity on software system design and development, improve students' professionalism. This course is an integrated application of theoretical knowledge and practical skills involved in the prerequisite course. The enterprise project should cover the knowledge of the major and practical courses that have been completed before as much as possible, focusing on the requirements design and implementation technology of the software project. Try to improve the students' ability in solving practical problems, project execution, teamwork, communication, professionalism and other soft skill.

A0S Comprehensive Case Studies Courses III

(2 Credits,2 Weeks; Course Category: Specialty Practical Course; Specialty: Software Engineering; Prerequisites: C Programming Language, Object-oriented Programming and Design, Introduction to Software Engineering, User Interface Design and Evaluation, Data Structure, Discrete Mathematics, Computer Network, Operating System, Database System, Software Architecture, Software System Analysis and Design Technology, Software Project Management, Software Quality Assurance and Testing Technology, Practice in Programming, Practice in Object-oriented and Interactive Application Development, Practice in Data Structure and Algorithm, Practice in Database Application System, Practice in Software Architecture, Comprehensive Case Studies Courses II)

This is one of the integrated practical courses for students majoring in software engineering. The course is implemented in the 3rd summer semester lasting for 2 weeks, a total of 10 work days, Suggested 80 hours. Through the introduction of real software projects, the real environment and the role of enterprises to develop students on the basic theory of software engineering, design and development skills, the comprehensive ability to analyze and solve the problem. This course provides the necessary theoretical basis and practical experience for the employment and entrepreneurship. After this enterprise training, students have capabilities of software product design, research, development, and promotion. Get experience in the start-up company's product operations, cost accounting and other aspects.

Comprehensive Practice of Data Structure and Algorithm

(2 Credits, 32 Hours; Course Category: Specialty Course; Specialty: Software Engineering; Prerequisite: Programming Fundamentals, Practice in Programming and Computer Science Discrete Mathematics, Object-oriented Programming and Design, Practice in Object-oriented and Interactive Application Development, Data Structure, Algorithm Design and Practice)

This is one of the comprehensive practical courses for students majoring in software engineering. The course focuses on the practice in algorithm design and application for the students with basic programming skills.

The objectives of the course are to help the students understand the core idea of data structures and algorithms, to master the logical relationship and storage structure of data structure, the idea and applications of algorithm, and the basic theory and application skills of discrete mathematics. As a result, the students have the ability to analyze problems using discrete mathematical models, to store data using data structures, and to solve practical problems using algorithms.

By the end of this course, students will be able to:

- (1) Master the logical structures and physical storage structures of data, algorithm design and implementation technologies;
- (2) Master the discrete mathematics analysis models that are used to analyze problems and using algorithm to solve the problems;
- (3) Comply with the standardized software development process including analysis, design, coding, debug, test and submission;
- (4) Complete algorithm applications in accordance with requirements;
- (5) Write software project documents in accordance with the engineering thinking;
- (6) Experience the team development process;
- (7) Complete the project and show their works by presentation.

After the course, the students will lay a solid foundation for the development of large-scale and complicated enterprise software applications.

Comprehensive Practice on Database Application System

(2 Credits, 32 Hours; Course Category: Specialty Basic Course; Specialty: Software Engineering; Prerequisite: Programming Language and Data Structure)

Comprehensive Practice on Database Application System is a core course of software engineering and other related disciplines, and it's a comprehensive practical course in fifth term. The course is designed for those students who have certain programming and development capabilities. Students should understand basic principles in operating system and database system, and they should also master the

core algorithms in these two courses, grasp a database management system application(DBMS), design and implement a web database application (contract management system).

This course includes thematic design and practical project two sections. The first section consists of several basic course practicing on database system and operating system; and the second section is an enterprise project (contract management system) which requires students to design and implement a database application based on web, then hand in its English experiment report and finish the performance for the project.

Course Code Software Process and Improvement

(2 Credits, 32 Hours; Course Category: Specialty Elective Course; Specialty: Software Engineering; Prerequisite(s): Introduction to Software Engineering)

Software process and improvement is a basic course of software engineering and software project management. The software process connects all the practical activities of all disciplines in software engineering. Its quality reflects the ability of the organization and has a significant impact on the final delivery of the project. This course studies how to build, run, manage and maintain software processes that enable organizations and projects to succeed.

The main contents of this course include software process basic concept (model, framework, standards), each domain process method and practice in software life cycle, software process assessment and improvement.

Through the study of this course, students can fully understand the whole process of software life cycle; they can have a systematic view of software project operation. Students can deeply understand and apply software process for different type of organizations and projects by comparing various models of software process framework. The students can solve the problems encountered and based on the continuous improvement of the software process to strengthen organization and team capability maturity.