Zhejiang Sci-Tech University

Teaching Syllabi of Computer Science and Technology Major

Chief Editor： Qing Yao

Editors：(names are arranged in no particular order)

Wei Shen Jia Bao Ting Shu

Yanyun Dai Haixiang Zhang

proofreader： Qiuhong Tian

Foreword

The teaching syllabus is the authoritative guidance of teaching plan and management, textbook compilation or selection and course evaluation. The Division of Academic Affairs supervises the general revision (or to some new courses, compiling) of the teaching syllabi to adapt to new trends, new normal and new concepts in tertiary education. The revised version emphasizes the student-centered approach in teaching, manifests frontier knowledge and represents development in education reform. The purpose of the revision is to enforce the requirements of Accreditation of Academic Programs and disciplinary assessment, to meet the demands of social and economic development and to guarantee the smooth implementation of the Program Outline (the 2016 edition) of each major for quality assurance of education.

This revision is major-based compilation of the course syllabi in ZSTU. Syllabi of the university general education courses are compiled into a separate volume. The volume of each major contains the syllabi of the courses of general education, basic major courses, major courses and practices.

The revision and compilation is achieved by the joint effort of the executives of teaching affairs and deans of each school and department. Participation of many teachers and support from the leaders of ZSTU guarantee its successful completion. Their efforts and support are hereby acknowledged.

The teaching syllabus only defines the general requirements of the course. Teachers are expected to explore new teaching materials and approaches and to renew course with the latest professional achievements to create high-quality curriculum with its own feature.

The completion of compilation of the teaching syllabi is the fundamental work. Further education reform is needed to promote the innovation of the curriculum system of each major, to improve the reform of teaching contents and approaches, in order to better achieve the course objectives. Comments and suggestions concerning this compilation will be greatly valued, since there is perpetual necessity to promote teaching practices and to achieve higher level of education quality.

Division of Academic Affairs

June, 2021

TABLE OF CONTENTS

Syllabus of Information Technology Fundamentals

Course Name/Title: Information Technology Fundamentals **Course Code: 62950**

Course Type： General Course, Compulsory Course

Total Teaching Hours: 16 (Classroom Hours: 16)

Course Credit: 2

I Course Introduction

This is the freshmen fundamental course of computer major in our university. Through the study of this course, students can master the basic knowledge of information technology, have the consciousness and ability of computational thinking, using computer to analyze and solve problems, and improve students' information ability and information literacy, strengthen the sense of responsibility of "strengthening the country through science and technology", cherish the feelings of home and country, and establish a correct political outlook. For students to lay a solid foundation to further learn computer related knowledge.

II Course Objective

Course Objective 1: Understand China's world status in the field of supercomputers and operating systems, promote patriotism, and understand the gap between these fields and the world's advanced level, so as to stimulate the sense of mission of striving and catching up. List the information technology used in fighting against COVID-19, witness the importance of the country's strong, embrace national feelings and establish a correct political outlook.

Course Objective 2: Master the representation of various types of data in the computer system; Master the architecture of computer system, understand the development history and current situation of software and hardware, and be able to analyze the problems and development direction of software and hardware; Understand computer network, information security technology and other computer related basic knowledge, understand the current situation and development trend of related technology, be able to retrieve relevant information and effectively use it according to needs.

Course Objective 3: Master the concepts, characteristics and classification of algorithms; Master the three kinds of structures and expressions of the algorithm and several commonly used algorithms; Use the flow chart to represent a simple algorithm to solve practical engineering problems. Understand the latest progress and new trends of information technology development; Cultivate the ability to actively acquire new information technology knowledge and new technology; To cultivate the comprehensive ability to retrieve, evaluate and effectively use information according to the needs, improve the ability to analyze and solve practical problems, and lay a solid theoretical foundation for students to independently analyze and solve practical problems in the information field in the future.

III Correlations between Course Objectives and Graduation Requirements

Table 1 Correlation between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 2: Problem Analysis	2-1 Can identify the complex engineering problems of computer, and express the requirements and key processes of computer complex engineering problems through investigation.	Course Objective 1,3
Graduation Requirement 7: Environment and sustainable development	7-1 Understand the connotation and significance of environmental protection and social sustainable development, and their interaction with computer engineering practice.	Course Objective 1,2

IV Correlations between Course Content and Course Objectives

Table 2 Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Chapter 1 Computer System (1) Development of computer; (2) Construction of computer hardware and software; (3) Von Neumann architecture	(1)The architecture of computer system and the relationship between software system and hardware system are described; (2)Expounds the composition of computer software and hardware system, and explains the working principle of computer; (3)Expounds the current situation of the development of computer hardware in China, and expounds that the research and development of Taihu Lake light supercomputer embodies the scientific spirit of independence, pioneering and innovation, and serving the country with science and technology of Chinese scientific researchers, enhances the sense of national pride, and enhances the sense of responsibility of "science and technology power".	2	Classroom teaching and discussion	Course Objectives 1,2
2	Chapter 2 Computer Software (1) Definition and classification of	(1) Expounds the definition, composition and classification of computer software; (2) The position of operating system in computer system is analyzed;	2	Classroom teaching and discussion	Course Objectives 1,2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	computer software; (2) Concept and classification of operating system	(3) Expounds the importance of operating system, and expounds that a large number of domestic operating systems represented by Deepin are rising, which ensures the national security of system software in our country.			
3	Chapter 3 Data Representation (1) Binary and conversion among different number systems; (2) Representation of different types of data	(1) Master the basis of binary, be able to skillfully convert between different number systems; (2) Expounds the representation and storage of different types of numerical values in computer.	4	Classroom teaching discussion	Course Objectives 2
4	Chapter 4 Fundamentals of algorithms (1) Concept and characteristics of algorithms; (2) Three types of algorithms.	(1) Explain the basic concepts, features and elements of the algorithm; (2) The algorithm can be represented by natural language, pseudo code and flow chart	2	Classroom teaching and discussion	Course Objectives 3
5	Chapter 5 Computer Network (1) Concept and architecture of computer network; (2) New development in computer network	(1)Expounds the basic concept, classification and architecture of computer network (2) Expounds the concept, characteristics and basic technology of LAN (3) At the beginning of COVID-19 in 2020, our country organized quickly to fight the epidemic in a short time. Through classroom discussion, we enumerated what information technology was used in fighting COVID-19. We should witness the importance of a strong country, cherish the feelings of family and country, and establish a correct political outlook.	4	Classroom teaching	Course Objectives 1,2,3
6	Chapter 6 Information Security (1) Definition and principle of virus and firewall; (2) Blockchain	(1) Explain system security and data security (2) Expounds the functions and characteristics of virus and firewall (3) Analysis of common information security cases behind the technical principles.	2	Classroom teaching	Course Objectives 2,3

V Period Distribution and Teaching Modes

(1) Period distribution of theory courses

Table 3 Period distribution of courses

Teaching Period / Course content	Theory teaching	Exercise Course	Discussion	Note	All
Chapter 1 Computer System	2				2
Chapter 2 Computer Software	2				2
Chapter 3 Data Representation	4				4
Chapter 4 Algorithms	2				2
Chapter 5 Computer Network	2		2		4
Chapter 6 Information Security	2				2
All	14		2		16

(2) Teaching Modes

The course is a basic theoretical course for computer major. At the same time, through classroom discussion, exercise explanation and small exercises, we can deepen the understanding and consolidation of theoretical knowledge and deepen the understanding of key points and difficulties; By the arrangement of a certain amount of homework, it help students further consolidate the understanding of classroom teaching content.

In order to make students give full play to their learning initiative, online teaching videos are provided for students to preview. Some discussion questions are arranged on the teaching network platform for students to think and submit answers online.

VI Assessment and Evaluation

Table 4 Assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Classroom performance	Attendance Other performances (answering questions, discussion, homework, etc.)	50%	Course Objectives 1,2,3
Final Exam	Closed-Book	50%	Course Objectives 1,2,3

VII Textbooks and References

(1) Textbook

Nell Dale, John Lewis. Computer Science Illuminated (The seventh edition). China Machine Press, 2020.

(2) References

[1] J.Glenn Brookshear. Computer Science: An Overview (The 13th edition). Posts & telecom

press, 2020.

[2] Yale N.Patt, Sanjay J.Patel. Introduction to Computing Systems (The third edition). China Machine Press, 2020.

[3] June Jamrich Parsons, DanOja. New Perspectives on Computer Concepts 2018: Comprehensive (The 20th edition). China Machine Press, 2019.

[4] Timothy J. O'Leary, Linda I. O'Leary, Daniel A, O'Leary. Computing Essentials 2021. China Machine Press, 2020.

Written by: Rong Jin **Reviewed by: Nang Zhang**

Date: 2021.05.20

Syllabus of C Programming*

Course Name/Title： C Programming **Course code：62900**

Course Type： (Basic Course + Compulsory Course)

Total Teaching Hours： (Classroom Hours: 48 Laboratory Hours:32)

Course Credit： 5.0

I Course Introduction

The course is an introduction to structured programming in C for solving problems for computer majors. The objective of this course is to enable the students to grasp the methods which break complex problems down into simple steps and then each step can be easily expressed in C, and ultimately to help the students to cultivate the thinking way of structured programming. Topics include variables declaration, data types, operators, control structures, functions, arrays and dynamic allocation principles, pointers, strings, structures and enumerations, file processing, algorithm development using flowcharts and extensive coverage of major C99 features, top-down and bottom-up software development, plotting for visualization ,writing, editing, compiling, linking and debugging computer programs.

II Course Objective

There are many popular programming languages. C is a fast, small, general-purpose, platform independent programming language. C is the most commonly used programming language in industry. C is the language of choice for programming embedded and mechatronic systems with hardware interfaces. C is one of the most commonly used programming languages in colleges and universities. C is the base for almost all popular programming languages. C excels as a model of programming languages. Once students have learned C, they can pick up any other languages by themselves. C is a standardized programming language with international standards.

Course Objective1: Grasping the C language foundations, utilizing the programming materials to design, write, debug and run C programs.

Course Objective2: The Course is also a demonstration of CT(Computational Thinking) and

The Top-Down Design method which breaks complex problems down into simple steps and then each step can be easily expressed. The Course emphasizes Algorithms + Data Structures = Programs

Course Objective3: The Course emphasizes TQ(Technology Quality）, the ability to design what we need by using engineering technologies.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Indicator point of Graduation Requirements	Course Objectives	Description of Correlations
Graduation Requirement 1	1.2 Students have acquired a reasonable professional knowledge structure. They can apply the knowledge and methods of discrete mathematics, algorithm, data structure and programming to Computational Thinking, to the analysis and implementation of basic algorithm problems and the analysis and understanding of the working principle or mechanism of complex software systems.	Course Objective 1,2	The Course is to grasp the C language foundations, to utilize the programming materials to design, write, debug and run C programs. The Course is also a demonstration of CT(Computational Thinking,). The analysis and implementation of some basic algorithm problems will be covered and emphasized, e.g. the bubble sort, the Queens problem.
Graduation Requirement 3	3.3 Be able to realize the computer system that meets the requirements according to the design scheme, fully considering the cost performance, and following the software engineering specification.	Course Objective 1,2,3	From the point of view of software engineering life cycle, the Course is a very important stage of the life cycle. The flowchart training and homework practices at the beginning of the course, the Top-Down Design method throughout the Course are powerful in helping students to break complex problems down into simple steps and then each step can be easily expressed.
Graduation Requirement 5	5.2 Be able to develop, debug and test computer systems, and understand their limitations by using integrated development tools, open source and third-party resources.	Course Objective 1,2	Our students are free and welcome to try any third party environments and resources to implement our application tasks. The Course has project assignment and paper home works and OJ lab works and Grab images lab works. A lot of chances are available for students showing Genuine New Idea Implementation by programming in C!
Graduation Requirement 7	7.2 Be able to assess the efficiency of resource utilization, the scheme of waste disposal and the safety precautions and identify the potential	Course Objective 2,3	The Course emphasizes programing styles, emphasizes sustainable programing without GOTO, emphasizes reusability, emphasizes

	hazards to human beings and the environment in the lifecycle of computer-related products for complex computer application problems.		easy maintaining codes, emphasizes comments and codes sharing and team working.

IV Correlations between Course Content and Course Objectives

Correlations between course theory contents and course objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Chapter 1 Getting Started The First C Program (1) Editing and Executing C Programs in ChIDE (2) Using Portable Commands to Handle Files in a Command Shell (3) Executing C Programs, Statements, and Expressions in Ch (4) Compiling, Linking, and Executing C Programs in a Command Shell (5) Editing, Compiling, Linking, and Executing C Programs in ChIDE	Showing and understanding the ChIDE, Command Shell, Editing, Compiling, Linking, and Executing C Programs, grasping Executing C Programs, Statements, and Expressions in Ch. Intellectual property thought and the cultivation of copyright consciousness	3	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 1, 2, 3
2	Chapter 2 Number Systems, Scalar Types, and Input/Output (1) Integer Number Systems (2) Character Set (3) Comments (4) Declaration (5) 32-Bit and 64-Bit Programming Models (6) Integer Types (7) Boolean bool Type in C99 (8) Character char Type (9) Real Floating-Point Types (10)*Complex Types in C99	Grasping all kinds of Number Systems, Scalar Types, Initialization ,and Declaration and Input/Output，understanding 32-Bit and 64-Bit Programming Models, Typedefs.	3	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 1, 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(11)The Pointer Type (12)Typedefs (13)Determining the Programming Data Model of a Computer (14)Initialization (15)Introduction to Formatted Input and Output				
3	Chapter 3 Operators and Expressions (1) Assignment Operator (2) Arithmetic Operators (3) Implicit Type Conversions (4) Precedence and Associativity of Operators (5) Relational Operators (6) Logical Operators (7) Bitwise Operators (8) Compound Assignment Operators (9) Increment and Decrement Operators (10) Cast Operators for Type Conversions (11) Sizeof Operator (12) Conditional Operator (13)Comma Operator	Grasping all kinds of Operators and Expressions, Precedence and Associativity of Operators. Cultivate team spirit and the sense of service.	3	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 1, 2
4	Chapter 4 Statements and Control Flow (1) Flowcharts for Algorithm Development (2) Simple and Compound Statements (3) Null and Expression Statements (4) Selection Statements (5) Repetition Statements (6) Jump Statements	Grasping Flowcharts for Algorithm Development, Simple and Compound Statements, Null and Expression Statements, Selection Statements, Repetition Statements, Jump Statements, Redirection of Input/Output , Compound applications and summary.	9		course objective 1, 2,3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(7) Redirection of Input/Output (8) Compound applications				
5	**Chapter 5** Functions (1) Function Definitions (2) Function Prototypes (3) Calling Functions: Call-by-Value versus Call-by-Reference (4) Standard C Header Files and Libraries (5) Mathematical Functions and Type Generic Functions in C99 (6) Functions for Mathematical Formulas (7) Plotting Functions for Graphical Display (8) Recursive Functions	Understanding the Function Prototypes, Plotting Functions for Graphical Display, Grasping Function Definitions, Calling Functions, Standard C Header Files and Libraries, Recursive Functions.	5	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 1, 2,3
6	Chapter 6 Arrays Declaration of Arrays (1) How Arrays Are Stored in Memory (2) Initialization of Arrays (3) Processing Data in Arrays (4) Passing Arrays to Functions (5) Plotting Data in Arrays for Graphical Display	Understanding How Arrays Are Stored in Memory, Plotting Data in Arrays for Graphical Display, Grasping Declaration of Arrays, Initialization of Arrays, Processing Data in Arrays, Passing Arrays to Functions.	6	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 1, 2,3
7	Chapter 7 Processing Directives (1) Macro Replacement (2) Source File Inclusion (3) Conditional Inclusion (4) Pragma Directive	Understanding the Source File Inclusion, Conditional Inclusion, Pragma Directive, Grasping Macro Replacement.	2	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 1, 2,3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
8	Chapter 8 Storage Classes and Program Structure (1) Global and Local Variables (2) Storage Classes (3) External Variables and Functions	Understanding the External Variables and Functions, Grasping Global and Local Variables, Storage Classes.	3		course objective 1, 2,3
9	Chapter 9 Pointers (1) Pointer Variables (2) Pointer Arithmetic (3) Calling Functions by Reference Using Pointers (4) Relation Between Pointers and Arrays (5) Using Pointers to Pass One-Dimensional Arrays to Functions (6) Pointers to Pointers (7) Arrays of Pointers	Understanding the Pointers to Pointers, Arrays of Pointers, Grasping Pointer Variables, Pointer Arithmetic, Calling Functions by Reference Using Pointers, Relation Between Pointers and Arrays, Using Pointers to Pass One-Dimensional Arrays to Functions	5	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 1, 2,3
10	Chapter 10 Characters and Strings (1) Character Code (2) Character Input and Output (3) Character-Handling Functions (4) Strings (5) String Input and Output (6) The Continuation Character (7) Converting Strings to Numerical Values (8) String Manipulation	Understanding the Character Code, Character-Handling Functions, The Continuation Character, Converting Strings to Numerical Values, Grasping Character Input and Output, Strings, String Input and Output, String Manipulation.	3	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 1, 2,3
11	Chapter 11 Structures, Enumerations, Unions, and Bit Fields (1) Structure Definition (2) Declaration of Structure Type (3) Pointer to Structures (4) Accessing Structure Members (5) Structure Initialization	Understanding the Enumerations, Unions and Bit Fields, Grasping Structure Definition, Declaration, Pointer to Structures, Accessing Structure Members, Structure Initialization, Size of Structures, Assigning and Comparing	3	Classroom lectures, online courses, classroom discussions,	course objective 1, 2,3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(6) Size of Structures (7) Assigning and Comparing Structures (8) Arrays of Structures (9) Passing Structures as Function Arguments (10) Functions Returning Structures (11) Functions Returning a Pointer to Structures (12) Handling Members of Pointer Type (13) Nested Structures (14) Enumerations (15) Unions (16) Bit Fields	Structures, Arrays of Structures, Passing Structures as Function Arguments, Function returning Structures, Functions Returning a Pointer to Structures, Nested Structures.		experiments, etc	
12	Chapter 12 File Processing (1) Opening and Closing Files (2) Reading and Writing Sequential Files (3) Design of a GPA Library Using Data Files and Its Applications	Understanding File Processing, Grasping Opening and Closing Files, Reading and Writing Sequential Files.	3	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 1, 2,3

Correlations between course experimentation contents and course objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Experimentation 1. Simple programming, ACM OJ(Online Judgement),Ch environment, COMMAND SHELL platform OJ problems contest 1 and paper	By finishing and debugging for the OJ problems programs and paper homework problems, Understanding the ChIDE, Command Shell, Editing, Compiling, Linking, and Executing C Programs, grasping	2	Submitting OJ problems programs online	course objective 1, 2, 3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	homework 1.	Executing C Programs, Statements, and Expressions in Ch, OJ(Online Judgement).			
2	Experimentation 2. Scalar types、operators and expressions(1) OJ problems contest 2 and paper homework 2.	By finishing and debugging for the OJ problems programs and paper homework problems, Grasping all kinds of Number Systems, Scalar Types, Initialization, Declaration and Input/Output.	2	Submitting OJ problems programs online	course objective 1
3	Experimentation 3. Scalar types、operators and expressions(2) OJ problems contest 3 and paper homework 3	By finishing and debugging for the OJ problems programs and paper homework problems, Grasping all kinds of Number Systems, Scalar Types, Initialization, and Declaration and Input/Output.	2	Submitting OJ problems programs online	course objective 1
4	Experimentation 4. Simple C OJ problems contest 4 and paper homework 4.	By finishing and debugging for the OJ problems programs and paper homework problems, Grasping Simple C programming, sequential programming.	2	Submitting OJ problems programs online	course objective 1
5	Experimentation 5. Logical structured programming OJ problems contest 5 and paper homework 5.	By finishing and debugging for the OJ problems programs and paper homework problems, Grasping Logical structured programming, sequential programming.	2	Submitting OJ problems programs online	course objective 1, 2
6	Experimentation 6. Iteration statements OJ problems contest 6 and paper homework 6.	By finishing and debugging for the OJ problems programs and paper homework problems, Grasping Logical structured programming, sequential programming, Iteration statements applications.	2	Submitting OJ problems programs online	course objective 1, 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
7	Experimentation 7. Mid-term examination Mid-term examination problems on OJ.	By Mid-term examination, it is a check on what we have studied on the fundamentals of programming and it is a half summary of the course.	2	Submitting OJ problems programs online	course objective 1, 2
8	Experimentation 8. arrays and sorting Lab works 1 and grab images and upload images to FTP homework space for arrays and sorting	By finishing and debugging for the Lab works problems programs, Grasping arrays applications and bubble sorting.	2	Upload grab images for Lab works to FTP homework space.	course objective 1, 2, 3
9	Experimentation 9. two dimensional arrays Lab works 2 and grab images and upload images to FTP homework space for two dimensional arrays	By finishing and debugging for the Lab works problems programs, Grasping two dimensional arrays applications.	2	Upload grab images for Lab works to FTP homework space.	course objective 1, 2, 3
10	Experimentation 10. Functions 1 Lab works 3 and grab images and upload images to FTP homework space for Functions 1	By finishing and debugging for the Lab works problems programs, Grasping functions definition and applications.	2	Upload grab images for Lab works to FTP homework space.	course objective 1, 2, 3
11	Experimentation 11. Functions 2 Preprocessing Directives Lab works 4 and grab images and upload images to FTP homework space for Functions 2	By finishing and debugging for the Lab works problems programs, Grasping functions applications and commonly uses Preprocessing Directives.	2	Upload grab images for Lab works to FTP homework space.	course objective 1, 2, 3
12	Experimentation 12. Pointers（1）	By finishing and debugging for the	2	Upload	course

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	Lab works 5 and grab images and upload images to FTP homework space for Preprocessing Directives	Lab works problems programs, Grasping Pointer Variables, Pointer Arithmetic, Calling Functions by Reference Using Pointers, Relation Between Pointers and Arrays.		grab images for Lab works to FTP homework space.	objective 1, 2, 3
13	Experimentation 13**.** Pointers（2） Lab works 6 and grab images and upload images to FTP homework space for Preprocessing Directives	By finishing and debugging for the Lab works problems programs, Understanding the Pointers to Pointers, Arrays of Pointers, Grasping Using Pointers to Pass One-Dimensional Arrays to Functions.	2	Upload grab images for Lab works to FTP homework space.	course objective 1, 2, 3
14	Experimentation 14**.** Strings Lab works 7 and grab images and upload images to FTP homework space for structure and union	By finishing and debugging for the Lab works problems programs, Grasping Strings applications and commonly used Strings functions.	2	Upload grab images for Lab works to FTP homework space.	course objective 1, 2, 3
15	Experimentation 15. Structure and union Lab works 8 and grab images and upload images to FTP homework space for structure and union	By finishing and debugging for the Lab works problems programs, Understanding the Enumerations, Grasping Structure Definition, Declaration, Pointer to Structures, Accessing Structure Members, Structure Initialization, Size of Structures, Assigning and Comparing Structures, Arrays of Structures, Passing Structures as Function Arguments, Function returning Structures, Functions Returning a Pointer to Structures, Nested Structures.	2	Upload grab images for Lab works to FTP homework space.	course objective 1, 2, 3
16	**C**hapter 12 File Processing	By finishing and debugging for the	2	Upload	course

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	Lab works 9 and grab images and upload images to FTP homework space for file operating	Lab works problems programs, Understanding File Processing, Grasping Opening and Closing Files, Reading and Writing Sequential Files.		grab images for Lab works to FTP homework space.	objective 1, 2, 3

V Period Distribution and Teaching Modes

1. Period Distribution

	Lecture	Experiment	Discussion	Others	Total
Getting Started	3				3
Number Systems, Scalar Types, and Input/Output	3				3
Operators and Expressions	3				3
Statements and Control Flow	9				9
Functions	5				5
Arrays	6				6
Processing Directives	2				2
Storage Classes and Program Structure	3				3
Pointers	5				5
Characters and Strings	3				3
Structures, Enumerations, Unions, and Bit Fields	3				3
File Processing	3				3
Simple programming, ACM OJ		2			2
Scalar types、operators and expressions(1)		2			2
Scalar types、operators and expressions(2)		2			2
Simple C programs		2			2

Logical structured programming		2			2
Iteration statements		2			2
Mid-term examination		2			2
arrays and sorting		2			2
two dimensional arrays and strings		2			2
Functions 1		2			2
Functions 2 Preprocessing Directives		2			2
Pointers 1		2			2
Pointers 2		2			2
Strings		2			2
Structure and union		2			2
File operating		2			2
	48	32			80

2. Teaching Modes

1) Teaching modes include Classroom lectures, online courses, classroom discussions, experiments, etc.
2) During the teaching processes of the course, we are going to take multiple platforms, the ACM OJ(Online Judgement)，the CHIDE environment, the LINUX environment, the COMMAND SHELL platform.
3) The course handouts are distributed at lecture time. Some of them are available on the Web of the home page which will be informed when it is ready. The homework is given out weekly in class through handout or through email.
4) Special stress should be put on the introduction of modern educational technology with an optimal integration of various teaching media.
5) Make reasonable use of presentation instruments, inquiring instruments, interaction instruments and design instruments to effectively improve the teaching quality.

VI Assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
conventional assignments	15 times	40%	Course Objective 1,2,3
mid-term exam	Online Judgement	10%	Course Objective 1,2,3

final exam	Closed Exam	50%	Course Objective 1,2,3
project	Group(one to three students)	Extral scores*	Course Objective 1,2,3

Extral scores*:one to ten points according to accomplishment quality. Extral scores can only make up the scores of conventional assignments and mid-term exam to no more than full scores.

VII Textbooks and References

1. Textbook

Harry H. Cheng. C for Engineers and Scientists: An Interpretive Approach. McGraw-Hill, 2009,ISBN: 978-0-07-337605-9.

2. References

[1] E Balagurusamy. Programming in ANSI C (Fourth Edition). Tsinghua University Press, 2007.

[2] Kernighan, B. W. and Ritchie, D. M., The C Programming Language, Prentice-Hall, Inc., Englewood Cliffs, NJ, second edition, 1988.

[3] Cheng, H. H., The Ch Language Environment Homepage, http://www.softintegration.com/docs/ch/

[4] IEEE, ANSI/IEEE Standard 754-1985, IEEE Standard for Binary Floating-Point Arithmetic, Institute of Electrical and Electronic Engineers, Inc., Piscataway, NJ, 1985.

[5] ISO, ISO/IEC Standard 9899:1990, Programming Language C, International Standards Organization, Geneva, 1990.

3. Resources

[1] http://mooc1-1.chaoxing.com/course/214786132.html

[2] http://i.mooc.chaoxing.com/space/index.shtml

[3] http://oj.acm.zstu.edu.cn/JudgeOnline/（10.11.246.150）

[4] ftp://10.16.23.2/

[5] http://10.16.23.20/

[6] http://iel.ucdavis.edu/

Written by：Desheng Huang **Reviewed by: Hairong Ye**

Date：2021.05.20

Syllabus of Introduction to Electronic Information as a Discipline

Course Name/Title： Introduction to Electronic Information as a Discipline*
Course code： 60903

Course Type：Specialized Course, Compulsory Course

Total Teaching Hours： 16 (Classroom Hours: 16)

Course Credit： 1

I Course Introduction

From the historical background of the emergence of computer science and technology methodology, the introduction course of electronic information discipline introduces some important basic concepts in the first level discipline of computer science and technology, focusing on the curriculum system, basic problems, core concepts, mainstream development direction, discipline methodology, knowledge organization structure and classification system of computer science and technology. Based on the concept of priority of connotation development, this course expounds how to make students correctly understand and learn computer science and technology in the process of cultivating first-class innovative talents and high-quality professional and technical development talents of computer science and technology. At the same time, it also describes the basic situation of computing model and digital computer system structure and working principle, software engineering, computer graphics and image processing, logic and artificial intelligence and evolutionary computing.

II Course Objective

Course Objective 1: Through the explanation, from the application of ancient abacus to China's leading advantage in modern service technology, we can promote patriotism; Through the application of information technology in many fields, we can cultivate students' spirit of hard study and craftsmanship.

Course Objective 2: focuses on the curriculum system, basic problems, core concepts, mainstream development direction, subject methodology, knowledge organization structure and classification system, subject personnel training objectives. Teaching focus and scientific literacy of computer science and technology. Understand the origin and use of abacus in ancient China, and promote patriotism.

Course Objective 3: Scientific problems in computer science. Guide students' interest and make them enter a good course learning link, which is conducive to the graduation project in the future.

Course Objective 4: Computers are used in social services. Cultivate students' practical ability and innovative consciousness.

Course Objective 5: Social and career related issues (including computer system based risks and responsibilities, intellectual property, Privacy and civil liberties, computer crime). Cultivate students' spirit of hard study and craftsmanship.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1	A systematic, theory-based understanding of the natural sciences applicable to the discipline	Course Objective 1,2,3
Graduation Requirement 7	Be able to understand and evaluate the impact of professional engineering practice on environmental and social sustainable development.	Course Objective 1,4
Graduation Requirement 8	With humanities and social science literacy, sense of social responsibility, student can understand and understand in engineering practice abide by the engineering professional ethics and norms, and fulfill the responsibilities	Course Objective 1, 5

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	The scientific problems in Computer Science	Scientific problems in Computer Science (including basic problems in main fields of computer science and some philosophical problems in Artificial Intelligence). Understand the origin and use of abacus in ancient China, and promote patriotism.	2	Classroom Lectures	Course Objective 1,2
2	Curriculum system and course selection of Computer Science	Curriculum system and course selection of Computer Science	2	Classroom Lectures	Course Objective 1,2,3
3	Core concepts in Computer Science	Core concepts in Computer Science (including algorithm, data structure, program, software, hardware system, etc.)	2	Classroom Lectures	Course Objective 1,2
4	Mathematical methods in Computer Science	Mathematical methods in Computer Science (including commonly used mathematical concepts and terms)	2	Classroom Lectures	Course Objective 1,2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
5	Overview of system science methods in Computer Science	Overview of system science methods in Computer Science (including structured method and object oriented method)	2	Classroom Lectures	Course Objective 2
6	Computer application in social service	The application of computer in social service. Cultivate students' practical ability and innovative consciousness	2	Classroom Lectures	Course Objective 4,5
7	Overview of operating system, compiling principle and software engineering	Operating system, compiling principle and software engineering.	2	Classroom Lectures	Course Objective 2,3
8	Social and professional issues	Social and professional issues (including risks and responsibilities based on computer systems, intellectual property rights, privacy and civil liberties, computer crime). Cultivate students' spirit of hard study and craftsmanship.	2	Classroom Lectures	Course Objective 1,4,5

V Period Distribution and Teaching Modes

1. Period Distribution

Teaching mode / hours / contents	lecture	exercise	Discussion	remark	S total
The scientific problems in Computer	2				2
Curriculum system and course selection	2				2
Core concepts in Computer Science	2				2
Mathematical methods in Computer	2				2
Overview of system science methods	2				2
Computer application in social service	2				2
Overview of operating system, etc.	2				2
Social and professional issues	2				2
Total	**16**				**16**

2. Teaching Ways

The course is a basic theory course of information technology, the content is complex and tedious, so we should pay attention to the combination of theory and practice in teaching methods.

Teacher should pay attention to arrange certain homework, and guide students to consult relevant information about the frontier and development trend of information technology.

In order to help students understand the content of classroom teaching better, we need to discuss in class.

In order to make students give full play to their learning initiative, online teaching videos are provided for students to preview. Some discussion questions are arranged on the teaching network platform for students to think and submit answers online. Through the distribution of questionnaires, timely understanding of students' learning, and make adjustments in teaching methods, timely answer questions.

VI Assessment

1. It is suggested that teachers should use the relevant propositions closely related to practical application, and let students submit reports and debate their experimental practice;

2. academic achievement (100) = usual achievement (30) + comprehensive practice achievement (70);

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
the rates of attendance	Number of attendance	5	1,2,3
homework	Submitted homework	10	1,2,3
Online test	scores	10	1,2,3
Self learning	hours	5	1,2,3
Comprehensive practice results	Report and debate	70	1,2,3,4

VII Textbooks and References

1.Textbook

① J.Glenn Brookshear, Computer Science: An Overview, 11th version

2.References

② 董荣胜，等编著：《计算机科学与技术方法论》，人民邮电出版社，2005 版出版

③ David Reed[美]著：张玉芳/孙思译：《新编计算机科学导论》，清华大学出版社，2005 年出版。

1. **Network Resource**

④ http://www.freetechbooks.com/introduction-to-computer-science-f10.html

⑤ https://alison.com/

Written by：Lican Huang **Reviewed by: Qiuhong Tian**

Date：2021.05.20

Syllabus of Discrete Mathematics

Course Name/Title： Discrete Mathematics B **Course code：** 62607

Course Type：Professional Basic Course、compulsory courses

Total Teaching Hours： 48 (Classroom Hours: 48)

Course Credit： 3

I Course Introduction

Discrete mathematics is a core curriculum of basic theory in computer science, with a study of discrete structure and relationships as the main target. The course is designed to cultivate students’ abstract thinking and strict logic reasoning ability and construct a good foundation of mathematics for computer application, information management and scientific research. Simultaneously, the course aims to help students to understand the inner relationship between abstract thinking and practice in computer science, and then gain the ability to apply these ideas to solve practical problems.

II Course Objective

Course Goal 1: Through explaining the basic concepts, basic principles, basic methods and application fields of discrete mathematics, from the application of ancient mathematics in our country to our country’s leading advantages in modern transportation, express delivery and logistics industries, we promote patriotism; The application in each field cultivates students' hardworking spirit and innovative spirit.

Course Goal 2: Master basic propositional logic, relation and set and graph theory knowledge, be able to use the basic concepts and principles of propositional logic, relation and set, and graph theory to solve discrete mathematics problems, be able to establish mathematical models for discrete systems, and use mathematical description tools for complex engineering Analysis and understanding of the problem.

Course Goal 3: Master the common methods of proof such as direct proof, contradiction, mathematical induction, construction, etc., and develop the ability to use symbolic and automated logical reasoning. Ability to use the concepts, theories and methods of discrete mathematics core propositional logic, relations, and graph theory, and to research and analyze complex engineering problems through literature research, scheme reasoning and other methods, and determine suitable solutions. Through the study of this course, students can master the ability of logical thinking and logical reasoning.

Course Goal 4: Cultivate students' independent thinking and innovative consciousness. The accumulation and creation of human knowledge is achieved step by step through continuous exploration and innovation. This course can guide and motivate students to analyze, explore and extend problems from basic problems, thereby achieving the cultivation of their independent thinking and innovative consciousness through exercises.

III Correlations between Course Objectives and Graduation Requirements(48 teaching hours)

Graduation Requirements	Graduation Requirements Index Point	Course Objectives	Support analysis description
Graduation Requirement 1：Engineering knowledge	1-2 Establish appropriate mathematical models for complex engineering problems, and use basic theoretical knowledge to solve them.	Course Objective 1、2	Establish mathematical models for discrete systems, use mathematical logic, relationship and graph theory analysis methods to solve the models, and give corresponding explanations, and understand the relationships between discrete quantities.
Graduation Requirement 2：Engineering knowledge	2-3 apply basic principles of mathematics, natural sciences and computer science, analyze the main factors affecting computer complex systems, demonstrate the rationality of solutions and obtain effective conclusions.	Course Objective 1、3	For the logical relationship in the real scene and the various phenomena of the natural system, it is necessary to apply reasoning, induction in discrete mathematics, and theoretical knowledge in graph theory, and use symbolic forms to automate logical reasoning to identify and judge key links and parameters of complex engineering problems.
Graduation Requirement 12：life-long learning	12-1 Recognize the necessity of continuous exploration and learning, have the awareness of active learning and lifelong learning, master independent learning methods, and understand the ability and ways to expand knowledge.	Course Objective 1、4	By learning the relevant knowledge of discrete mathematics, modeling complex problems to mathematical models, to stimulate the consciousness of independent thinking and innovative thinking, and then cultivate the continuous learning ability of continuous learning and continuous innovation.

(Note: Basic courses and specialized courses must correlate with the graduation requirements as specified in the Program outline. The correlated graduation requirement index point must be put before the descriptive phrases or sentences. General courses are exempted from this rule.)

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter1 Propositional Logic** (1) The definition, symbolization and connectives of propositions, formulas and classification of propositions; (2)Equivalent calculus of propositional formula, main disjunctive paradigm and main conjunctive paradigm of propositional formula, full set of connectives, reasoning theory.	(1) Explain the definition of the proposition formula, master the symbolic method of the proposition formula; list the five commonly used conjunctions and their truth tables; and establish a systematic mathematical logic model； (2) Perform equivalence calculations of propositional formulas and judge the types of propositional formulas; be able to calculate the main conjunctive normal form and main disjunctive normal form of the proposition formula, and then analyze its assignment and related properties; (3) Explain the full function set of connective words and explain their mutual substitution rules; enable analysis and reasoning of propositional logic. (4) Enumerate the application of ancient Chinese mathematics knowledge, such as Qing Zhu's entry and exit, nine chapters arithmetic, understand the wisdom of ancient Chinese people, and promote patriotism. (5) Nowadays, my country's logistics system is developed, and the express delivery industry has a wide coverage and fast speed. Through classroom discussions, list what mathematical knowledge is used. Witness our country's amazing development speed and efficient innovation spirit, embrace family and country feelings, and establish a correct political outlook.	9	Lecture, online courses, discussions	Course Objective 1、2、3、4
2	**Chapter2 Predicate Logic** (1) Basic concepts of predicate logic, formulas and explanations of predicate logic；	(1) Explain the definitions and basic concepts in predicate logic, and build predicate logic models of natural systems； (2) Describe the steps and methods of	9	Lecture, online courses, discussi	Course Objective 1、2、3、4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(2) Quantifiers in predicate logic and their toe paradigm; predicate logic equivalents and predicate logic proof.	predicate logic symbolization in detail, and analyze its nature and corresponding interpretation; (3) Explain the two quantifiers commonly used in first-order logic (universal quantifiers and existential quantifiers) and their usage methods; be able to perform equivalent transformation of first-order logic formulas, solve and analyze the toe-beam paradigm; be able to perform first-order logic Logical reasoning and proof.		ons	
3	**Chapter3 Basic concepts and operations of sets** (1) the concepts of set, representation of set and the relationship between sets； (2) relevant operation of set、Count of elements in the set。	(1) Explain the basic concepts of sets and the representation methods of sets in detail, including enumeration methods, element properties description methods, etc.; define the relationship between sets (inclusive, equal)； (2) perform set related operations, including union, complement, intersection, symmetric difference, relative complement, power set, Cartesian product, etc.; able to use the Venn diagram to calculate the number of elements in the set.	5	Lecture, online courses, discussions	Course Objective 1、2、4
4	**Chapter4 Binary relations and functions** (1) Cartesian product of sets、the definition of relations and representation； (2) the basic operations of relations、Properties of relations, closure、Equivalence relations and Partially ordered sets； (3) the definition of functions, discriminant methods of function Properties and the basic operations.	(1) Explain the definition of the relationship in detail, the enumeration method of the binary relationship, the relationship matrix, and the relationship diagram； (2) Able to perform common relational operations, including relational inverse operations, relational composition, and exponentiation operations; explain the definition of relational closures and the solving methods of relational reflexive closures, symmetrical closures, and transitive closures; (3) Explain two special binary relations: equivalence relations and partial order relations in detail. Be able to obtain the equivalence class	12	Lecture, online courses, discussions	Course Objective 1、2、4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		and quotient set of any equivalence relation; be able to obtain the corresponding Hass diagram according to the partial order relation, and analyze the special elements on the Hass diagram, including the largest element and the smallest element, the maximum element and Minimal element, upper bound and lower bound； (4) Ability to determine the properties of functions and perform compound operations of functions.			
5	**Chapter5 Basic concepts of graphs** (1) undirected graphs and directed graphs, paths, circuits and connectivity； (2) Three matrix representation methods of graphs, shortest path, critical path, and coloring.	(1) Explain the basic concepts of graphs in detail, including undirected graphs, directed graphs, degrees, simple graphs, complete graphs, etc； (2) Explain in detail the three representation methods of graphs, including incidence matrix, adjacency matrix, reachability matrix, etc., as well as the connectivity and classification of graphs。 (3) Be able to use the knowledge of graphs to model and analyze actual problems, including shortest path problems, critical paths, and coloring problems; (6) List the staff deployment problems and shortest path problems in real work and life, introduce their connection with graph theory knowledge, and stimulate students' inspiration from practical problems to mathematical modeling; through classroom discussion, list other existing mathematical modeling problems , Feel the beauty of mathematics, feel the growing strength of the motherland, and promote patriotism.	8	Lecture, online courses, discussions	Course Objective 1、2、4
6	**Chapter6 Special Graph**	(1) Explain the relevant definitions of the four	5	Lecture,	Course

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	Bipartite graph, Euler graph, Hamilton graph, floor plan.	types of graphs, including bipartite graphs, Euler graphs, Hamilton graphs, and plane graphs in detail； (2) Able to judge several special graphs； (3) Able to apply several special graphs to practical application problems.		online courses, discussions	Objective 1、3、4

V Period Distribution and Teaching Modes

VI Assessment

This course adopts a centralized written examination assessment method. The test questions can be multiple-choice questions, fill-in-the-blank questions, short answer questions, calculation questions or essay questions.

(1) Final assessment

Score evaluation: total score (percentage system) = usual homework × 10% + course test × 20% + final test score × 60%

Assessment Methods or Approaches		Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
usual homework	Classroom performance	attendance rate, classroom performance, and small class exercises.	20%	Course Objectives 1、2、3、4
	Usual homework	the number of times the homework, the accuracy rate, and the writing quality.		
course test		According to the evaluation of the correctness of the test paper, the correctness of the answering steps and the correctness of the problem-solving ideas will be given points as appropriate.	20%	Course Objectives 2、3、4
final test score		According to the evaluation of the correctness of the test paper, the correctness of the answering steps and the correctness of the problem-solving ideas will be given points as appropriate.	60%	Course Objectives 1、2、3、4

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

（2）Evaluation standard

The assessment method of this course includes three parts: usual scores and final exam scores. The evaluation criteria are shown in Tables 7-8.

Table 7 Evaluation Criteria for Normal Performance

Usual grades	Evaluation standard	Calculation formula
Classroom performance A	① According to the number of check-ins, 10 points for full attendance. 0.5 points will be deducted for 1 time late, and 1 point will be deducted for 1 time absenteeism; the score will be recorded as A1. ② Class performance, full score is 10. 1 point will be deducted if the question is not answered; the score is recorded as A2. ③ Classroom exercises, with a maximum score of 10. One point is deducted for each practice without answering; the score is recorded as A3.	A=(A1+A2+A3)/30*100
Usual homework B	① According to the number of submissions, 50 points for all submissions. 10 points will be deducted for one missed payment. The score is recorded as B1. ② According to the accuracy of homework and writing quality, excellent 40 points, good 40 points, medium 35 points, passing 30 points, failing 30 points or less; the score is recorded as B2。	B=B1+B2
Classroom test C	Test 2 times, with a total score of 100 points each time. According to the assessment of the correctness of the test paper, the correctness of the answering steps and the correctness of the problem-solving ideas are given points as appropriate; the scores are recorded as C1 and C2.	C=(C1+C2)/2
Total score	**Usual grades =A*30%+B*30%+C*40%**	

Table 8 Evaluation Criteria for Final Exam Results

Question type	Focus of investigation	Evaluation standard	Proportion	Course Objectives

Fill-in-the-blank/multiple choice/simple calculation (Optional)	Determine the basic concepts, basic theories, basic analysis methods, calculation skills, commonly used paradigms and binary relations in propositional logic, first-order logic, sets, relations and graph theory.	According to the evaluation of the correctness of the test paper, the correctness of the answering steps and the correctness of the problem-solving ideas will be given points as appropriate.	60%	Course Objectives 2、3
Comprehensive calculation and analysis questions	The establishment of simple proposition formulas, predicate logic formulas, relations and special graphs solve mathematical models in representation methods, practical meanings and standard forms, and provide physical explanations; basic concepts and judgments of plane graphs, Euler graphs and Hamilton graphs Some important conclusions in engineering applications such as theorems and practical applications。	According to the evaluation of the correctness of the test paper, the correctness of the answering steps and the correctness of the problem-solving ideas will be given points as appropriate.	40%	Course Objectives 1、2、3、4

VII References

Textbook：

1. Kenneth H. Rosen (Eds.) Discrete Mathematics and Its Applications. China Machine Press, 2012.

Reference Books：

1. Qu wan-ling etc. Discrete Mathematics. Tsinghua University Press, 2005.
2. Seymour Lipschutz, Marc Lipson. Discrete Mathematics. American Muze Inc, 2002.

Written by：Qiuhong Tian **Reviewed by: Lichan Huang**

Date：2021.5.20

Syllabus of Object-Oriented Programming A

Course Name/Title: Object-Oriented Programming A **Course code:** 62906

Course Type：(Basic Course, Specialized Course)(Compulsory Course)

Total Teaching Hours: 48 (Classroom Hours:39 Laboratory Hours or Tutorial Hours:9)

Course Credit: 3

I Course Introduction

It is a compulsory basic course for computer majors. And it is an important course that combines the characteristics of C++ language to talk about the idea of object-oriented programming. Through the study this course, we can understand the idea and concept of object-oriented programming, master the method of object-oriented programming, and use C++ language as a practical tool to design and implement object-oriented programming. It enables students to write application programs in C++, establish a correct sense of professional honor and professional mission, have a serious and rigorous attitude towards study and work and lay a solid foundation for further learning the ability of software system analysis and design in complex environment.

II Course Objective

Course Objective 1: Understand the development history and latest trends of computer software industry; Understand the status and role of software industry in the field of information technology, as well as the development status of China's software industry; Establish a correct sense of professional honor and professional mission, have a serious and rigorous attitude towards study and work, and lay a correct ideological foundation for their future professional work.

Course Objective 2: Understand the basic syntax and structure of C++, be able to apply object-oriented programming method to system modeling, and understand the advantages of object-oriented programming method in solving the complexity of large-scale system and software reuse. Enable students to have a higher level of problem solving, programming, exploration and innovation ability.

Course Objective 3: Can apply the object-oriented programming method, design the implementation scheme, write the C++ program in line with the idea of object-oriented programming. At the same time, can continuously improve the design scheme according to the running results, so that the running results of the program meet the needs of software development.

Course Objective 4: Can choose and use appropriate development tools and STL resources to develop, debug and test C++ programs to meet the requirements of software development and improve the development and execution efficiency of software.

Course Objective 5: By consulting the technical documents, students can understand the new features of the new C++ standard, and explore the solutions to various problems from various angles, so as to lay a solid theoretical foundation for analyzing and solving the practical problems in the field of computer engineering independently in the future. Cultivate the spirit of inquiry and innovation, and meet the future challenges through autonomous learning and lifelong learning.

III Correlations between Course Objectives and Graduation Requirements

Table 1 Correlation between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 2: Problem Analysis	2-2 Through literature analysis, find a variety of solutions and improvement methods for computer complex engineering problems, and correctly describe the solutions used	Course Objective 2
Graduation Requirement 4: Research	4-1 Be able to research and verify the key algorithms and modules related to computer science	Course Objective 1, 3
Graduation Requirement 5: Using modern tools	5-2 Be able to use integrated development tools, open source and third-party resources to develop, debug and test computer systems, and understand their limitations	Course Objective 4
Graduation Requirement 12: Lifelong learning	12-2 Be able to meet the needs of personal career development, pay attention to the frontier and trend of computer field, learn new technologies independently, and adapt to the development of the times and environmental changes	Course Objective 1, 5

IV Correlations between Course Content and Course Objectives

(1) Relationship between theoretical teaching contents and curriculum objectives（39hrs）

Table 2 Relationship between theoretical teaching contents and curriculum objectives

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 Introduction of OOP** (1) Programming paradigms (2) Introduction of OOP (3) Development of C++ (4) Fundamentals of C++	(1) Combined with the daily use of software, from the size, complexity and other aspects of different programming paradigm (2) Introduces the concept of object-oriented from the familiar concepts of animals, social organizations and so on. (3) Let students participate in the summary of the advantages and disadvantages of C, lead to the development and improvement path of C++ (4) Based on the idea of object-oriented programming, it can guide students to respect science, attach importance to technology, and fully understand that science and technology is the primary productive force, and the problem demand generated by the development of the times is the objective soil for the development of science and technology.	3	Classroom teaching, discussion	Course Objective 1,2,5
2	**Chapter 2 Fundamentals of C++** (1) Const (2) Reference & Pointer (3) new & delete (4) Function Prototype and Argument passing (5) Default arguments of function (6) Function overloading & function template (7) inline function (8) Namespace	(1) Introduce const from the deficiency of macro (2) Introduces the reference through the inconvenience of pointer, and compares the differences and relations between reference and pointer (3) Compared with C, new and delete are introduced (4) Starting from the modification of function name in compiling process, explains function prototype, parameter passing and default parameter, function overloading and inline function (5) Through the program example, introduce namespace (6) The dialectical content of improvement and innovation is integrated to make students	3	Classroom teaching, discussion	Course Objective 2

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		realize the importance of reform to the national development and social progress.			
3	**Chapter 3 Classes and Objects** (1) Definition of Class and Object (2) Constructor & Destructor (3) Object Array & Object Pointer (4) this pointer (5) Static Members (6) Friend function & Friend class	(1) Introduces the concept of class, the definition of class and object, member variables and member functions from struct. By comparing with struct, it is easy for students to understand and deepen their impression. (2) Constructor and destructor are introduced by solving problems (3) Through examples to explain the object array, object pointer, object reference and the corresponding array. (4) Explain this pointer through inquiry learning. (5) Based on practical problems, static member variables and static member functions are introduced (6) Introduce friendship mechanism through discussion on national treatment of foreign citizens.	9	Classroom teaching, discussion, lab	Course Objective 2,3
4	**Chapter 4 Inheritance** (1) Concept of Inheritance and Derivation (2) Inheritance mode of class (3) Constructor and Destructor of derived class (4) Multiple Inheritance (5) Initialization of derived class object by virtual inheritance	(1) Review the definition of class, the types of reuse, and introduce inheritance and derivation. (2) Types and grammatical mechanism of inheritance (3) Using base class members in derived classes (4) The advantages and disadvantages of repeated inheritance (5) Introduces virtual destructor through destructor errors, and explains the construction and destructor order of derived class objects (6) Discuss the advantages and disadvantages of inheritance and different inheritance methods.	6	Classroom teaching, discussion, lab	Course Objective 2,3
5	**Chapter 5 Polymorphism**	(1) Introduce polymorphism through examples	6	Classr	Course

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(1) Concept of Polymorphism (2) Dynamic binding and static binding (3) Virtual function (4) Pure virtual function and abstract class (5) Virtual destructor	(2) The concept of joint editing is introduced in combination with the course of operating system and compilation (3) Focus on the definition and use of virtual function, as well as the difference with function overloading (4) The virtual base class, pure virtual function and abstract class are further derived from the virtual function (5) Guide students to discuss the advantages and uses of virtual functions, and encourage students to explore the realization mechanism of virtual functions		oom teaching, discussion, lab	Objective 2,3
6	**Chapter 6 Operator Overloading** (1) Concept of Overloading (2) Common operator overloading (3) Overloading of function operator and input/output operator (4) Overloading of friend function and member function	(1) Understand concept of overloading (2) Master overloading of common operator (3) Master overloading of special operator (4) Overloading of input/output operator (5) Discusses and verifies the priority and ambiguity of operator overloading, as well as the use and precautions in type conversion	3	Classroom teaching, discussion	Course Objective 2,3
7	**Chapter 7 Template & STL** (1) Concept of Template (2) Function Template (3) Class Template (4) STL (5) Container, iterator and algorithm of STL	(1) Internal relations and similarities and differences of ideas among variables, data types, classes and templates (2) Master the definition of function template (3) Master definition and instantiation of class template (4) Learn to use container, iterator and algorithm of STL (5) Combine with exception	3	Classroom teaching, discussion	Course Objective 4,5
8	**Chapter 8 Exception** (1) Concept of Exception (2) Exception handling mechanism	(1) Familiar with exception (2) Master the exception handling mechanism of C++ (3) Understand the exception class of C++	3	Classroom teaching,	Course Objective 2,3

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(3) Exception class in C++	standard library (4) Combined with the exception, introduces the engineering thinking of redundancy safety and bottom line guarantee, and guides students to explore the realization cost of the exception, to understand that redundancy, safety and guarantee have costs, and to consider comprehensively and dialectically when analyzing and solving problems		discussion	
9	**Chapter 9 File & Stream** (1) Input/output (2) Operation of stream (3) Format output (4) String stream (5) File and stream	(1) The input and output of computer are abstracted into stream (2) The polymorphism of stream and the matching of diversified output (3) The function of string stream in device, memory and file (4) Stream plays an important role in program and storage, database and storage, network and cloud storage (5) Through the hierarchical structure of stream, compares the important role of information construction in the development of national science and technology, and improves students' sense of mission and honor in professional learning	3	Classroom teaching, discussion	Course Objective 2,3

(2) Relationship between experimental teaching contents and curriculum objectives (9hrs)

Table 3 Relationship between experimental teaching contents and curriculum objectives

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Lab 1 Definition and application of class** (1) Declaration of class (2) Definition and initialization of object (3) Data member and member function	(1) Familiar with IDE of C++ (2) Learn to use header file of C++, namespace and output (3) Declaration of class and definition of object	3	Classroom teaching and lab	Course Objective 2,3
2	**Lab 2 Inheritance** (1) Declaration of Inheritance (2) Access mode of base class member (3) Type compatibility of derived and base class objects	(1) Application of Inheritance (2) Difference among different inheritance mode (3) Initialization of derived object (4) Access base class member by derived class object	3	Classroom teaching and lab	Course Objective 2,3
3	**Lab 3 Polymorphism** (1) Virtual function (2) Destructor of virtual base class (3) Application of STL	(1) Application of Polymorphism (2) Virtual base class and abstract base class (3) Virtual destructor (4) STL and polymorphism (5) Declaration and definition of virtual function	3	Classroom teaching and lab	Course Objective 2,3,4

V Period Distribution and Teaching Modes

(1) Period distribution of theoretical courses

Table 4 Period distribution of theoretical part

Teaching Mode / Teaching period / Teaching Content	Theory teaching	Exercise	Discussion	Note	Total
Chapter 1 Introduction	3				3
Chapter 2 Fundamentals of	3				3
Chapter 3 Class and Object	7	1	1		9
Chapter 4 Inheritance	5		1		6

Chapter 5 Polymorphism	5		1		6
Chapter 6 Operator	2		1		3
Chapter 7 Template and	4	1			3
Chapter 8 Exception	3				3
Chapter 9 File and Stream	3				3
Total	**33**	**2**	**4**		**39**

(2) Period distribution of experimental courses

Table 5 Period distribution of experimental part

Name	Content	Environment	Period	Group size	Attributes (Basic/ Comprehensive / Design / Creative Study)	Requirements (Compulsory /Optional)
Lab 1: Class and Object	Based on the object-oriented programming method, abstracts and encapsulates a class of objects with C++, and writes a test program to test the rationality of the object instantiated by the class.	Visual Studio C++ or Codeblocks or Dev C++ or other C++ IDE	3	1	Basic	Compulsory
Lab 2: Inheritance	Using C++ based on object-oriented programming method to realize the function derived class of the class in the first experiment, or rewrite a class	Visual Studio C++ or Codeblocks or Dev C++ or other C++ IDE	3	1	Comprehensive	Compulsory

	family with inheritance relationship, and write a test program to test the rationality of the object instantiated by the derived class.					
Lab 3: Polymorphism	Based on the object-oriented programming method, a small system is realized, which contains several classes with inheritance relationship, can be realized by polymorphism, and is closely related to practical problems.	Visual Studio C++ or Codeblocks or Dev C++ or other C++ IDE	3	1	Comprehensive	Compulsory

(3) Teaching Modes

This course is a very practical course, the content has its systematic structure, but also complex and scattered. Therefore, in the teaching method, we should change the traditional teaching method of simply instilling knowledge in the classroom, and implement heuristic and discussion teaching.

When studying the teaching methods of the course, we should adopt different teaching methods according to the content and the students' conditions. We should combine the methods of self-study, explaining the key and difficult points, organizing classroom discussion or interspersed discussion in the teaching. We should pay attention to mobilize the enthusiasm of students, and change the original process of "full class irrigation" into the process of teachers and students' joint exploration under the guidance of teachers.

VI Assessment and Evaluation

(1) Assessment

Table 6 Course Assessment

Assessment Methods or Approaches		Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Performance	Classroom performance	Attendance Other performances (answering questions, discussion)	30%	Course Objectives 1,2,3,4,5
	Homework	Submit homework on time and quality		
Experiment		According to the classroom performance, experimental scheme design, experimental operation and quality of experimental report evaluation.	20%	Course Objectives 1,3,4,5
Final Exam		Closed-book exam	50%	Course Objectives 2,3,5

(2) Assessment criterion

The assessment of the performance is shown in Table 7-9.

Table 7 The Evaluation of the usual performance

Usual performance	Evaluation Criterion	Equation
Classroom performance A	④ According to the number of sign in, full attendance is 10 points. 0.5 point will be deducted for one time of being late and 1 point will be deducted for one time of absenteeism; The score is A1. ⑤ Classroom performance, full score 10. 1 point will be deducted if the question is not answered; The score is A2.。 ⑥ Small exercises in class, with a full score of 10. 1 point will be deducted if no answer is given in each exercise; The score is A3.	A=(A1+A2+A3)/30*100
Homework B	According to the number of completions, full completion is 10 points. 0.5 point will be deducted for one time of being late and 1 point will be deducted for one time of no submission; The score is B.	B
Total Score	**Usual Performance=A*40%+B*60%**	

Table 8 Evaluation of experiment

Aspects	The quality of Code	The quality of report	The quality of solution
Weight	50%	20%	30%
Excellent (A)	The code is very clear, with readable comments; results are correct; innovations are involved in the experiments.	Text format is standard, the report is very clear and easily to be understood.	The problem and the existing solutions are reasonably analyzed; the solution embodies the innovative consciousness, and the result analysis is reasonable
Good (B)	The code is clear, with readable comments; results are correct.	Text format is fine; the report is clear and readable.	The problem and the existing solutions are reasonably analyzed; the solution is based on some existing solutions.
Medium (C)	Code format is standard; can achieve experimental objectives; individual conditions are not considered	Text format is not right; some contents are unclear and are difficult to be understood.	The solution is fine, but the analysis is not sufficient.
Pass (D)	Basically achieve the experimental goal	Have required contents.	The solution is workable.
Fail (E)	Do not achieve the experimental goal	Do not fully contain the required contents.	The solution is not workable.
Total	**The quality of Code*50% + The quality of report*20% + The quality of solution*30%**		

Table 9 Evaluation criteria of final exam

Type	Emphasis on	Criteria	Percentage	Course Objective
Fill in the blanks/Multiple Choice Questions/ Answer the questions (Optional)	Mainly focus on the basic concepts, basic theories and basic usages of C++ foundation, object-oriented, inheritance, derivation, polymorphism, exception handling, paradigm programming, file and stream.	If the answer is correct, scores will be given.	10%	Course Objective 2
Program Reading /Find the errors(Optional)	Class declaration and object definition, construction and deconstruction process, function overloading, template, class inheritance and derivation, virtual function	According to the reference answers of the test paper, those who are partially	40%	Course Objective 3

	and polymorphism, initializer list, operator overloading, static member, friend and so on.	correct or show clear ideas can be given points as appropriate		
Fill in the blanks for program(Optional)	Object oriented mechanism, exception handling, template related syntax, semantics, execution order, ambiguity, etc	According to the reference answers of the test paper, those who are partially correct or show clear ideas can be given points as appropriate	20%	Course Objective 2, 3
Program Design	The problem can be solved by analyzing, designing and programming with OOP knowledge	According to the reference answers of the test paper, those who are partially correct or show clear ideas can be given points as appropriate	30%	Course Objective 2,3

VII Textbooks and References

(1) Textbook

Bjarne Stroustrup. The C++ Programming Language (The fourth edition). China Machine Press, 2016.

(2) References

[1] Bjarne Stroustrup. Programming: Principles and Practice Using C++ (The first edition). China Machine Press, 2009.

[2] Stanley B. Lippman. C++ Primer (The fifth edition). Addison-Wesley Professional, 2012.

[3] Scott Meyers. Effective C++: 55 specific ways to improve your programs and designs (The third edition), Publishing House of Electronics Industry, 2011.

Written by: Rong Jin **Reviewed by: Qiaoli Zhuang**

Date: 2021.05.20

Syllabus of Curriculum Design for Object-Oriented Programming

Course Name/Title: Curriculum Design for Object-Oriented Programming

Course code: 62989

Course Type: Basic Course, Specialized Course, Compulsory Course

Total Teaching Hours: One week (Laboratory Hours 20)

Course Credit: 1

I Course Introduction

It is a compulsory basic course for computer majors. It is an important course combining the characteristics of C++ and the idea of object-oriented programming. On the basis that students have basically understood and mastered the basic concepts and methods of object-oriented programming, and can initially use C++ programming language to realize, through the study of this course, they can use object-oriented programming method to analyze, design and put forward program solutions for engineering problems with a certain degree of complexity, and use C++ programming language to realize. And establish a correct sense of professional honor and professional mission, have a serious and rigorous attitude towards study and work.

II Course Objective

Course Objective 1: Establish a correct sense of professional honor and professional mission, have a serious and rigorous attitude towards study and work, and lay a correct ideological foundation for their future professional work.

Course Objective 2: According to the assigned proposition, after completing a relatively complete application requirement analysis and functional module design independently, can give a reasonable design scheme of class that conforms to the idea of object-oriented programming, and strengthen students' understanding, application and internalization of the idea of object-oriented programming.

Course Objective 3: Be able to skillfully use the integrated development environment of C++ to build the software development environment based on multiple files, and implement, debug, analyze and improve the design scheme, so as to further improve the students' programming engineering ability.

Course Objective 4: After the integration, improvement and perfection of the system implementation, the whole process of curriculum design report from analysis, design, implementation to summary can be presented according to the requirements. The C++ program meets the task requirements and runs without errors.

Course Objective 5: Be able to present and report the project results in the form of oral expression, and answer the relevant questions raised by teachers or students accurately, so as to lay a certain foundation for future project communication and team cooperation.

III Correlations between Course Objectives and Graduation Requirements

Table 1 Correlation between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 3: Design/ development solutions	3-4 Adopts the idea of hardware and software cooperation, integrates, improves and perfects the system implementation, and can present the design results in the form of drawings, documents and objects	Course Objective 1, 4, 5
Graduation Requirement 4: Research	4-2 Based on scientific principles and scientific methods, can make experimental plans, build experimental systems and carry out experiments for the overall realization of complex computer systems	Course Objective 2
Graduation Requirement 5: Using modern tools	5-1 Be able to use software and hardware simulation tools to verify the computer related theory, simulate and analyze the system design scheme, and understand its limitations	Course Objective 1, 3
Graduation Requirement 10: Communicate	10-1 Be able to make effective written and oral representations on complex engineering problems, and communicate effectively with others, including writing reports and design documents, making statements, clearly expressing or responding to instructions	Course Objective 4, 5

IV Correlations between Course Content and Course Objectives

Table 2 Correlations between Experimental Course Content and Course Objectives

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Detailed interpretation of the purpose, content and requirements of curriculum	(1) The content of curriculum design must have certain engineering complexity, clear task objectives and detailed functional description;	2	Classroom teaching	Course Objective 1, 2

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	design, understand assessment methods	(2) Establishing a grading mechanism from low to high in task complexity; (3) Through the demonstration of the excellent works, the students will be inspired to challenge their professional spirit			
2	Analyze the system requirements, divide the functional modules and complete the class design	Help the students to clarify the task objectives, improve the design of functional modules, rationalize the design of classes, and make them conform to the idea of object-oriented programming	2	Individual counseling	Course Objective 2
3	Write and debug C++ program according to the design scheme	Guide students to build C++ programming environment and use C++ to realize class design	4	Lab	Course Objective 3
4	Interim checking	Check the rationality of the system design scheme and the progress of the program development	2	Presentation	Course Objective 2
5	Modify and perfect the design scheme and procedure	According to the suggestions given by the teachers in the mid-term examination, the design scheme and procedure should be further modified and improved to cultivate the students' scientific spirit of preciseness and excellence	4	Lab	Course Objective 1, 2, 3
6	Project presentation	Students demonstrate and explain the results of the project. Teachers should ask at least three questions for students to answer	4	Presentation	Course Objective 4
7	Write a report on course design	Curriculum design report should at least include analysis, design, implementation and summary of four parts, illustrated, logical clear, standard format	2	Individual counseling	Course Objective 5

V Period Distribution and Teaching Modes

(1) Period distribution of courses

Table 3 Period Distribution of Curriculum Design

Name	Contents	Environments	Period	Group size	Attributes(Basic /Comprehensive/ Design/	Requirements(Compulsory /Optiona

					Creative Study)	l)
Guiding topic selection and task interpretation	Detailed interpretation of the purpose, content and requirements of curriculum design, clear assessment methods	C++ IDE	2	1	Comprehensive	Compulsory
System solution design	Analyze the system requirements, divide the functional modules and complete the class design	C++ IDE	2	1	Comprehensive	Compulsory
Program implementation	Write and debug C++ program according to the design scheme	C++ IDE	4	1	Comprehensive	Compulsory
Interim checking	Check design and program completion	C++ IDE	2	1	Comprehensive	Compulsory
Program improvement	Modify and perfect the design scheme and procedure	C++ IDE	4	1	Comprehensive	Compulsory
Project Presentation	Project Presentation	C++IDE	4	1	Comprehensive	Compulsory
Report writing	Write curriculum design report, including task content, purpose, requirements, indicators, business process design, function design, class design, code and operation combination and analysis	C++IDE	2	1	Comprehensive	Compulsory
Total			20			

(2) Teaching Modes

1. This course is mainly based on computer experiments. Teachers need to explain to students the nature, tasks, requirements, course arrangement and progress, contents to be assessed,

experimental rules and laboratory safety system of the course.

2. Each time the teacher gives the topic for the students to choose, the students can also draw up their own topic, and the rationality of the content is checked by the teacher. Students are required to complete the tasks specified in the selected design project. The complexity of the tasks increases from low level to high level. Students are encouraged to challenge themselves.
3. Students with ability are encouraged to use the third-party development library to design UI and database based system development.

VI Assessment and Evaluation

(1) Assessment

Table 4 Assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Curriculum design report	It mainly investigates the rationality, integrity and format standardization of the project design	30%	Course Objectives 2,4
Function completion	According to the hierarchical scoring mechanism in the project assignment	20%	Course Objectives 2,3,4
Code quality	Score according to the rationality of class design and the standardization of code writing	20%	Course Objectives 2,3
Presentation	Score according to the answer to the questions raised by the teacher	20%	Course Objectives 5
Attendance	Score according to the required class attendance	10%	Course Objectives 1

(2) Assessment criterion

The assessment method of this course includes five parts: course design report, function completion, code quality, defense and attendance. The evaluation criteria are shown in Table 5.

Table 5 Reference standard of course design scoring

Project	Percentage	Excellent (>90)	Good (80~89)	Medium (70~79)	Pass (60~69)	Fail(<60)

Curriculum design report	30%	The content is complete, the design is reasonable, the drawing is standard, the text is clear and fluent, no typos, the format fully meets the requirements, there is a detailed analysis and summary	To some extent the content is complete, the design is reasonable, the drawing is standard, the text is clear and fluent, no typos, the format meets the requirements, there is analysis and summary	The content is relatively complete, the design is basically reasonable, the drawing is basically standard, the writing is basically fluent, there are few typos, the format basically meets the requirements, there is analysis and summary	The content is basically complete, the design is somewhat reasonable, the drawing is basically standard, the writing is basically fluent, there are few typos, the format basically meets the requirements, and there is no analysis and summary	The content is incomplete, the design is unreasonable, the drawing is not standardized, the writing is not fluent, there are many typos, the format does not meet the requirements, there is no analysis and summary
Function completion	20%	The running program has friendly interface and friendly design, and has completed all functions between 60 and 90 of the task	The running program has beautiful interface and friendly design, and has completed all functions between 60 and 80 of the task	The interface of the running program is neat, and all the functions between 60 and 70 of the task are completed	The interface of the running program lacks friendliness, and only completes the function of <= 60 of the task.	The interface of the program running is not friendly, and it fails to complete any group of tasks of the task, or the completed program is defined as plagiarism
Code	20%	The design	To some	Class design is	Class design	Class design

quality		of the class is reasonable, the code writing is standard, and the readability is good.	extent Class design is reasonable, code writing is standard, readability is good.	basically reasonable, code writing is basically standard, and readability is acceptable	is not reasonable, code writing is standard, readability is acceptable	is not reasonable, code writing is basically standard, readability is acceptable, class design is unreasonable, code writing is not standard, readability is poor, or the completed program is defined as plagiarism
Presentation	20%	Be able to make clear and complete description of the design scheme, and answer the questions raised by the teacher timely and accurately	Be able to explain the design scheme clearly and completely, and answer some of the questions timely and accurately	Can carry on the basic elaboration to the design plan, can answer the teacher's question accurately basically	The explanation of the design scheme is not clear enough to answer the teacher's questions, but it is not accurate enough	Unable to explain the design clearly, unable to give answers to the teacher's questions
Attendance	10%	No absenteeism	No absence, late for one	No absence, late for one time,	Absence for one time, or	Absence 2 times or more,

		, lateness or early leave	time, no early leave	early leave for one time	late for one time, early leave for one time	or late, early leave 3 times or more

VII Textbooks and References

(1) Textbook

Bjarne Stroustrup. The C++ Programming Language (The fourth edition). China Machine Press, 2016.

(2) References

[4] Bjarne Stroustrup. Programming: Principles and Practice Using C++ (The first edition). China Machine Press, 2009.

[5] Stanley B. Lippman. C++ Primer (The fifth edition). Addison-Wesley Professional, 2012.

[6] Scott Meyers. Effective C++: 55 specific ways to improve your programs and designs (The third edition), Publishing House of Electronics Industry, 2011.

Written by: Rong Jin **Reviewed by: Qiaoli Zhuang**

Date: 2021.05.20

Syllabus of Lectures on Special Topics

Course Name/Title：　Lectures on Special Topics　　　　　　Course Code: 62599

Course Type：　Specialized Compulsory Course

Total Teaching Hours：16（Classroom Hours: 16 Laboratory Hours or Tutorial Hours：　0）

Course Credit：1.0

I Course Introduction

This course is a basic course for computer majors. The course is for the students of computer major in junior grade. From the computer frontier technology, application field, technical problems and other aspects, from the perspective of information processing and software design, the latest research direction and research results of computer related disciplines are introduced.

The course is mainly carried out in the way of topic sharing. The main task of the course is to share and introduce the research direction, research ideas, methods and research results by the lecturer, or to introduce and share the problems and achievements by the students. Based on these, the students can understand the technical characteristics and development frontier of related majors, and explore and analyze the problems through literature analysis. The feature of the course is that it covers the latest research results and application examples of related professional fields. In the teaching process, by mining the popular computer related technologies and their applications in real life, we can guide students to have an insight into the importance and development prospects of computer related majors, and then cultivate students' professional quality and scientific spirit. At the same time, through the introduction of China's development and advantages in the field of related software technology, carry forward patriotism.

II Course objective

Course objective 1：Through learning the development process, application status, and prospect of software related technology, students' interest in professional learning is stimulated, and their professionalism and scientific spirit are cultivated; Through explaining the development and advantages of our country in the field of related software technology, carry forward the patriotic feelings.

Course objective 2：Through studying this course, students can understand the technical characteristics and development frontier of related majors；

Course objective 3：Through studying this course, students can understand the basic ideas and process of research, and are able to actively explore problems, consult literature and analyze problems;；

Course objective 4：Through the study of this course, we pay attention to the cultivation of students' communication ability during the process of exploration, and enable students to carry out subject research step by step from the aspects of expounding problems, expressing views, and sharing achievements.

III Correlations between Course Objectives and Graduation Requirements

Table 1 Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1：Problem knowledge	2-2 Be able to conduct literature analysis, in order to find a variety of solutions and improvement methods for computer complex engineering problems, and correctly describe the methods used.	Course objectives 3
Graduation Requirement 8：Professional norms	8-3 Understand the social responsibility of computer engineers for public safety, health and well-being, as well as environmental protection, and be able to consciously abide by and perform corresponding responsibilities in computer engineering practice.	Course objectives 1，3
Graduation Requirement 10：Communication	10-1 Be able to make effective written and oral representations on complex engineering problems, and also can communicate effectively with others, including writing reports and designing documents, making statements, clearly expressing or responding to instructions.	Course objectives 4

IV Correlations between Course Content and Course Objectives

Table 2 Correlations between theory Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 Software development and its applications** (1) The process of software development； (2) Typical software	(1) Show the software development stages； (2) Understand the characteristics of different software technologies and their application contexts； (3) Discuss the impact of software technologies on the development of	4	Classroom teaching; discussion	Course Objectives 1，2，3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	technologies and their applications.	society; Discuss the role of software technology in promoting social development, establish professional confidence and stimulate interest in professional technology learning; (4) To understand the development of related fields in China, and stimulate national pride and patriotism.			
2	**Chapter 2 Intelligent Software development and its applications** (1) The development of intelligent software； (2) Typical intelligent software technologies and their applications.	(1) Show the characteristics and advantages of intelligent software; (2) Understand the characteristics of different software technologies and their application contexts； (3) Discuss the impact of intelligent software technologies on the development of society; Discuss the role of software technology in promoting social development, establish professional confidence and stimulate interest in professional technology learning; (4) To understand the development of related fields in China, and stimulate national pride and patriotism. (5)	4	Classroom teaching; discussion	Course Objectives 1，2，3
3	**Chapter 3 Software Analysis and Measurement** (1) Software analysis and verification； (2) Software testing and measurement.	(1) Software analysis and verification； (2) Software testing and measurement.	4	Classroom teaching; discussion	Course Objectives 1，2，3
4	**Chapter 4 Discussion of software development**	Through group discussion on the development prospects and challenges of different software technologies, we can cultivate the spirit of scientific exploration	4	Discussion、oral presentation	Course Objectives 1、2、3、4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		and critical spirit.			

V Period Distribution and Teaching Modes

（1）Period Distribution of theory courses

Table 3 Period distribution of theory courses

Teaching Modes / Period / Course Contents	Classroom Teaching	Exercise Courses	Discussion		Total
Chapter 1 Software development and its applications	4				4
Chapter 2 Intelligent Software development and its applications	4				4
Chapter 3 Software Analysis and Measurement	4				4
Chapter 4 Discussion of software development			4		4
Total	**12**		**4**		**16**

（2）Teaching Mode

This course is a basic course, in order to make students understand the related technology research and the latest application development of computer major. We must let students take the initiative to explore problems, consult literature and analyze problems after class, and then effectively complete the learning work of subject topics. In terms of teaching methods, we should pay attention to the initiative of students, and pay attention to arrange certain links of after-school access, classroom sharing and Research Report, so as to help students understand the content of classroom teaching.

VI Assessment and criterion

Table 4 Course Assessment

Assessment Methods or Approaches	Requirement	Weighting	Evaluation of Course Objectives
Classroom performance	Attendance Other performances (answering	10%	Course Objectives 2

	questions, discussion)		
Discussion performance	According to the class discussion / preparation report: from the report content, ideas, expression of three aspects of evaluation.	40%	Course Objectives 2，4
Final (Report)	According to the paper evaluation, theoretical basis, topic selection and research methods, analysis and summary, document writing, etc.	50%	Course Objectives 2，3

VIII Textbook and References

（1）Textbook

无

（2）References

[1] 王伟主编：《计算机科学前沿技术》，清华大学出版社，2012 年出版

[2] 王长胜,主编：《中国信息年鉴》,中国信息年鉴期刊社, 2008 年出版

[3] 章毓晋编著：《图像处理和分析》. 清华大学出版社，1999 年出版.

[4] （美）冈萨雷斯等编著： 数字图像处理（第二版）（英文版）. 电子工业出版社，2007 年出版.

Written by：Mingyue Jiang **Reviewed by：Dongming Xiang**

Date：2021.5.20

Syllabus of Digital Electronic Technology

Course Name/Title： Digital Electronic Technology **Course code：61905**

Course Type： Specialized Compulsory Course

Total Teaching Hours： 80 (Classroom Hours:64 ,Laboratory Hours:16)

Course Credit：5

I Course Introduction

Digital Electronic Technology is one of the basic subject courses for major of computer science and technology. This course mainly focuses on then physical circuit realization of logic operation. It is a key course for students to correctly understand the computing principles of modern digital electronic computer and their physical structure. Through the study, students can initially understand the physical connotation of computing symbols and algorithms, the physical implementation of the state machine and the principle equivalence of software and the specificity of the performance architecture. At the same time, the analysis and comprehensive application ability of the basic digital logic circuits are obtained to lay the necessary knowledge foundation for the study of subsequent courses.

This course comprehensively applies a mathematical knowledge that realizes the recursion of the human computational process to a simple physical structure. It is a course of profound epistemological value. Through the in-depth study of this course, it will be very helpful to establish the correct outlook on world and life.

II Course Objective

Course Objective 1: Understand the ultimate physical conversion of the human computing process and strengthen the understanding of the materialist world view.

Course Objective 2: Master the basic knowledge of logic algebra and can use it to express, solve and optimize the digital circuit problems in engineering.

Course Objective 3: Master the functions and basic application of various logic gates, triggers, small and medium scale integrated circuits, use them to analyze, compare and design combinational and sequential logic circuits involved in engineering problems, independently design experimental schemes and complete the course experiments, analyze and summarize the experimental process, obtain reasonable and effective data and conclusions, write experimental reports, and feedback to the design practice of complex engineering.

Course objective 4: Understand the working principle and key parameters of logic gate, trigger and small scale combinational / sequential integrated circuit, analog to digital / digital to analog converter and memory, select appropriate devices and their peripheral circuit parameters according to engineering applications, debug digital electronic system, and initially have the ability to design or optimize complex digital electronic control system.

Course Objective 5: Be familiar with websites such as common electronic engineer communities, technology forums, R & D forums and library electronic resources. Cultivate social ability, draw lessons from other's engineering experience, understand relevant technical standards and application dynamics, and develop the spirit of social cooperation.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1	1.1 Use mathematics, natural sciences, engineering foundations, and expertise for the appropriate representation of computer complex engineering problems.	Course Objective 2,3
Graduation Requirement 2	2.3 Use the basic principles of mathematics, natural science and computer science to analyze the main factors affecting the computer complex system, demonstrate the rationality of the solution and obtain effective conclusions.	Course Objective 3,4
Graduation Requirement 2	6.1 Familiar with computer-related technical standards, intellectual property rights, information security specifications, industrial policies, laws and regulations, and understand the responsibilities.	Course Objective 4,5

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	An Introduction to the General (1) Digital and analog quantities (2) Logic Level and Digital Waveform (3) basic logic function (4) combinatorial logic and timing logic (5)programmable components (6)Common instruments	(1) Understand the course content, teaching methods and assessment methods; (2)Understands the physical expression of continuous functions and discrete sequence; (3) Adroitly express the physical expression form and process of the logical quantity; (4) Adroitly express the basic logic function operation;	4	Lectures, discussion	Course Objective 1,2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		(5) Understand the basic categories, functions, and basic analysis design process of digital logic circuits. (6) Gradually establish conceptual connections between physical expression, mathematical forms and computational models.			
2	Digital system, operation, and encoding (1) Digital system (2) Digital Conversion (3) Binary operations (4)Complement and symbols (5) Decimal encoding (6)Binary encoding (7) Error-correction code	(1) Proficient in the conversion of the two-ten-eight-hexadecimal system; (2) Proficient in binary complement operation; (3)Understand expression format of the floating-point number; (4)Understand the BCD encoding system; (5) Understand the encoding rules of the cyclic code system; (6) Understand the principles of the error correction code (7) Establish the symbol system concept of the alphabet-0,1 to string, and extend to the philosophy concept of Chinese ancient Zhouyi, and understand the principle of all symbol systems equivalent to each other.	4	Lectures	Course Objective 1,2
3	Logic gate circuit (1) Not-gate, and gate and or gate (2)Exclusive or and nor gate (3) PLC circuit (4)Common 74 logic gate	(1) Proficient expressing the logical algebraic operation functions of various logic gates; (2) Familiar with ANSI/IEEE standard logic symbols; (3) The concept of programmable logic can be clearly expressed; (4) Correctly draw the timing relationship between the input and outputs of logic gates; (5) The electrical characteristics of CMOS devices and TTL devices can be clearly compared;	4	Lectures	Course Objective 2,3
4	Boolean Algebra and Logical Simplification (1) Boolean operations and	(1) The basic formula of Boolean algebra can be applied for the expression deduction;	6	Lectures	Course Objective 2,3,4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	expressions (2) Boolean Algebra Basic Formula (3) DeMorgan's Theorem (4) Logical circuit analysis (5)Logical Circuit simpl ification (6)Standard Boolean expression (7) Truth table (8)Karnaugh-map simplification	(2)Proficient in applying DeMorgen's threom for expression transformation and simplification; (3)Conversion of Logic expression to POS an SOP form; (4)Conversion between truth table and logic expression; (5)Proficient in Karnaugh map simplification			
5	Combinational Logic circuit (1)Basic combinational logic circuit (2)Implementation of combinational logic circuit (3) Nand and nor gates (4)Waveform operation	(1) Proficient analyze the basic combinational logic circuits; (2)According to the logic expression or truth table, skillfully apply the combinational logic circuit design process to design logic circuit and simulation; (3) Implement logic functions with Nand gate.	6	Lectures	Course Objective 2,3,4,5
6	Combinational logic function (1)Half adder and full adder (2) Parallel adder (3)Look-forward adder (4) Comparator (5) Decoder (6) Encoder (7) Data selector (8) Demultiplexer (9) Parity checker	(1) The difference between half adder and full adder, serial input adder and the look-forward input adder can be clearly explained; (2) Use comparator and its extension method to compare two binary numbers; (3) Knows how to implement binary decoders; (4) Master the process of implementing logic functions using a data selector; (5) Knows how to use a seven-segment decoder in the display system; (6) Adroitly apply priority encoders;	12	Lectures	Course Objective 1,2,3,4,5
7	Latches, triggers and timers (1) Latch (2) trigger (3) trigger properties (4) trigger application (5) 555 timer (6) single stable circuit (7)Multi-harmonic oscillator	(1) Use gate to form basic latches (2)Understand the difference between latches and triggers (3)Master circuit structure and characteristic function of R-S,D latch; (4) Proficient in the J-K trigger circuit structure, trigger mode and characteristic equation;	10	Lectures	Course Objective 1,2,3,4,5

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		(5) Use triggers to achieve simple sequential logic circuits; (6) Proficient in 555 timer and realize the circuit structure form, working process and parameter calculation of the single stable circuit and multi-harmonic oscillator.			
8	Shift register (1) Shift register data I/O (2)Bidirectional shift register (3) Ring counter (5)Application of shift register	(1) The input and output mode of the data in shift register can be clearly explained; (2) Master the process and method of realizing the sequence signal generator using ring counter; (3)Use the shift register for the time-delay operation; (4)Use the shift register for serial-parallel data conversion	4	Lectures	Course Objective 1,2,3,4,5
9	Counters (1) Asynchronous counter (2) Synchronous counter (3) Reversible counter (4)Synchronous counter designation (5) Counter Cascading (6) Counter application	(1)Correctly express the difference between asynchronous and synchronous counters; (2)Draw timing waveform of sequential circuit; (3)Master common counters; (4)Master the process of changing the modulus of the counter; (5)Master the cascading mode and extension of the counters; (6)Uses reversible counters to achieve reversible binary sequence generators; (7)Master the method of designing counters for specific state sequences.	12	Lectures	Course Objective 1,2,3,4,5
10	Signal conversion and processing (1)Anolog-digital conversion method (2)Digital-to-anolog conversion method (3) DSP Introduction	(1)Analog-to-digital conversion can be explained; (2)Elaborate the necessity of filters; (3) Describe the sampling process; (4)The working principles, characteristics and working process of common ADC can be elaborated; (5)The working process of DAC can be elaborated correctly; (6)The most critical elements of digital	2	Lectures, discussion	Course Objective 1,2,3,4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		signal processing can be described.			

V Period Distribution and Teaching Modes

(I)Course period distribution

Class room hours：

Modes / Period / Teaching content	Lecture	Exercise	Discussion	Remarks	Total
1Introduction	3		1		4
2Digital system, operation, and encoding	4				4
3Logic gate	4				4
4 Boolean Algebra and Logical Expression	5	1			6
5Combinational logic circuits	5	1			6
6 Combinational logic function	7	1			8
7Lathes,flip-flop and timers	12	1			13
8Shift register	4				4
9Counters	10	1			11
10Signal conversion	3		1		4
Total	57	5	2		64

Note: Teachers should flexibly arrange classroom discussion and exercises during course.

Laboratory period distribution：

Experiment Project Name	Content feed	Main Equipment Or experimental environment	Experiment period	Number of per group	xperimental properties (Basic / Integrated / Design / Research Innovation)	Issuing requirements (required / optional)
Experiment1: Parameters test of logic gate	Test of voltage transmission characteristics and output characteristics of TTL Nand	Digital test box, oscilloscope and multimeter	2	1	Basic	Required

	gate					
Experiment 2: Logic function test of integrated logic gates	Logical function testing of TTL Nand gate and other gates; implement OR gate and Xor gate with Nand gate	Digital test box, oscilloscope	2	1	Basic	Required
Experiment 3：Design of combinational logic circuit	Design an alarm control circuit; Design a street lamp control circuit; Design a one-bit full adder	Digital test box, multimeter	2	1	Design	Required
Experiment 4：Data selector and its application design	Design three-variable logic function and blood type matching circuit with 74LS153,74LS151.	Digital test box, multimeter	2	1	Design	Required
Experimental 5: Trigger and its application	Function test of D and JK trigger; T, trigger composed of D trigger;	Digital test box, oscilloscope, and function signal generator	2	1	Basic	Required
Experiment 6: Integrated counter and its application (I)	Hardware implementation of the 4-bit asynchronous binary counters; design BCD5421 code with74LS161.	Digital test box, oscilloscope, and function signal generator	2	1	Integrated	Required
Experiment 7: Integrated	Design and implementation	Digital test box, oscilloscope, and	2	1	Integrated	Required

counter and its application (II)	of N and controllable counters.	function signal generator				
Experiment 8: 555 timer and its appplication	Logic function test of the 555 timer, design a monostable trigger and a multiharmonic oscillator with 555 timer.	Digital test box, oscilloscope, multimeter, and function signal generator	2	1	Basic	Required

(II)Teaching methods

The course should be taught by lectures, online video review, online studying, online discussion, classroom discussion. During the lectures teachers should adapt to the characteristics of students in this major, integrate curriculum knowledge points with other professional courses such as computing theory, and strengthen students ' understanding of knowledge concept. Teachers should use the stylus, presentation and other modern educational means as far as possible and record classroom explanation videos for students to review after class. This course should make full use of the school's existing online curriculum platform to access all course resources (presentations, homework, roll-in, online discussion, etc.) online for students to self-study and organize teaching. Classroom discussion should be arranged first, randomly selected students to speak on stage, for questions and discussion. Each student must prepare a speech on the topic.

VI Assessment

(Assessment content and methods must correlate with course objectives.)

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Course homework	no less than 12	20	course objectives 1,2,3,4
Experimental results	Experimental completion status	20	course objectives 2,3,4,5
Final examination	Rating criteria	60	course objectives 2,3,4

VII Textbooks and References

（I）Textbook:

[1]Thomas L. Floyd：《Digital Fundamentals（Eleventh Edition）》，电子工业出版社,2017.

（II）**Reference:**

[1]Thomas L. Floyd 著, 余璆，熊洁 译:《数字电子技术（第十一版）》，电子工业出版社，2019.

[1] 阎石等编著: 《数字电子技术基础》(第 6 版), 高等教育出版社, 2010.

[2] 康华光编著: 《电子技术基础.数字部分》(第 6 版), 高等教育出版社, 2014.

Written by：Kong yanyan **Reviewed by: Jiang xusheng**

Date：2021.05.20

Syllabus of Curriculum Design for Digital Electronic Technology

Course Name/Title：Curriculum Design for Digital Electronic Technology**Course code：61601**

Course Type：Specialized Compulsory Course

Total Teaching Hours：20 (Classroom Hours:4 ,Laboratory Hours:16)

Course Credit：1

ICourse Introduction

Curriculum Design for Digital Electronic Technology is a specialized compulsory coursefor computer science and technology major. It is a practical teaching part based on the theory of digital electronic technology. It is one of the professional basic courses to cultivate students ' basic skills of electronic technology, and carry out follow-up professional course learning and graduation design. Through the study of this course, students are required to master the basic concepts, basic theories and analysis and design methods of digital electronic technology. According to design tasks, they canindependently design schemes, build unit circuits, complete joint debugging, obtain test results, summarize and write design reports, to achieve the course objectives.

IICourse Objective

Course objective 1: Understand the relevant science and technology culture and technology development trends in the field of digital electronic technology application; Understand informatization, intelligence and automation in the development of digital electronic technology; Perceptual science and technology is the embodiment of the primary productivity in digital electronic technology. Strengthen sense of responsibility as a strong country in science and technology, and establish lofty ideals.

Course objective 2: Understand the basic theory, basic analysis and design methods of digital electronic technology, understand the practical significance of digital system design in engineering application. Preliminary learn to consult the component manual, and reasonably choose electronic components. Understand the various factors affecting design objectives and technical scheme, have a certain comprehensive design ability of digital circuit.

Courseobjective 3: Learn welding technology and complete the design and production of unit circuits independently. Familiar with the common EDA software. Be able to use multisim and other tools software to conduct circuit simulation and analysis, experience the advantages of EDA software simulation, and understand the difference between software simulation and the actual hardware implementation.

Course Objective 4: Master the method of circuit testing of common electronic instruments and learn to eliminate circuit faults. Have the debugging skills of digital circuits, use software and hardware coordination thought to improve the system realization. Comprehensively applythe theory and method of digital systemanalyze and solve engineering application problems.

Course Objective 5: Independently complete the design content of this course, analyze and summarize the experimental process, get reasonable and effective data and conclusions, write design reports and feed back to the design practice of complex engineering, so as to improve the ability to analyze and solve practical problems independently. With a strong sense of digital information, put forward and implement new ideas on circuit design and application combined with intelligent information processing technology.

IIICorrelations between Course Objectives and Graduation Requirements

Table 1 Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives	Support Analysis Description
Graduation Requirements 3: Design / Development solutions	3.1 Master the whole-cycle development process, basic design / development methods and technology of the computer system and software, understand various factors affecting the design objectives and technical solutions;	Course objectives 1,2	Understand the basic theory and design methods of digital electronic technology, learn to consult the component manual, and reasonably choose the electronic components.
	3.4 Adopt the coordination idea of software and hardware, integrate, improve and improve the system,use drawings documents and physical objects to present the design results;	Course objectives1 、4	Master the method of common electronic instruments for circuit testing, have the debugging skills of digital circuits, and use the coordination idea of software and hardware to improve and improve the system realization.
Graduation Requirements 5: Using modern tools	5.1 Use software and hardware simulation tools to verify the computer-related theories, conduct simulation and analyze the system design scheme, understand its limitations;	Course objectives1 、3	Complete the design and production of the unit circuit, conduct the circuit simulation and analysis with multisim and other software, experience the advantages of the software simulation,understand the difference with the actual hardware implementation.
Graduation Requirements 10: Communication	10.1 Make effective written and oral statements on complex engineering issues, and effectively communicate with others, including writing reports and design documents, presentation, clear expression or response to	Course objectives1 、5	Analyze and summarize the experimental process, obtain reasonable and effective data and conclusions, write the design report and feedback to the design practice of complex engineering.

	instructions;		

IV Correlations between Course Content and Course Objectives

Table 2 Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Teaching content 1. Guide the topic selection and scheme design.** (1) Introduce design topics, working principles, design requirements; (2) Requires students to consult the information by themselves, design the corresponding circuits and scheme demonstration, and reasonable selectrelevant components; (3) Circuit simulation and analysis with Multisim and other tool software.	(1) Master the basic theory and basic skills of digital electronic technology, and thus have the ability of comprehensive design digital circuits; (2) Independently think, consult materials and books, and design solutions. (3) Experience the advantages of EDA software simulation,understand the difference between software simulation and the actual hardware implementation. (4) Understand the history of domestic chips in the difficult development of the technology blockade, and carry forward patriotism.	4	Lectures、Discussion、Computer	Course objectives 1、2、3
2	**Design and production of teaching** content **2 module circuit** (1) Learn the welding technology; (2) Completes the test selection of the components; (3)Build the design circuit physically, and complete the preliminary test of the module circuit.	(1) Master the basic welding technology; (2) Independently build the circuit and use common electronic instruments for circuit debugging.	8	Lectures、Discussion、Video	Course objectives 1、3、4
3	**Teaching content 3. Overall debugging and data** record **of the system circuit** (1) Use signal source, oscilloscope and other experimental equipment to debug the circuit of each module	(1) Master the comprehensive debugging method of the digital system; (2) Learned to eliminate circuit faults himself, with the craftsman spirit of studying hard; (3) Master the use of basic experimental equipment such as signal source and	4	Lectures、Experiment	Course objectives 1、4、5

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	overall; (2)Record the relevant parameters and indicators.	oscilloscope, and can effectively record the experimental data.			
4	**Teaching content 4. Test, defense and report writing** (1)Assessment and test the theoretical knowledge such as circuit principles related to the course content, (2) Summarize the design content, organize the relevant data, and write the design report.	(1) Master the basic theoretical knowledge of digital electronic technology; (2) Summarize and analyze the experimental results, form a complete and effective design report. (3) Perception of "science and technology is the first productivity" in digital electronic technology. Strengthen the sense of responsibility as a strong country in science and technology, and establish lofty ideals.	4	Computer、Exam	Course objectives 1、2、5

V Period Distribution and Teaching Modes

（一）Period Distribution

Table 3 Course content period distribution

Teaching content	Content feed	Main equipmentOr the experimental environment	Period	Number of per group	Experimental properties (Basic / Integrated / Design / Research Innovation)	Issuing requirements (required / optional
Guide the topic selection scheme design and demonstration	Introduce design topic, working principle, design requirements, design report format, etc.; learn to design circuit with EDA software.	Computers equipped with EDA tools software such as Multisim.	4	1	Integrated	Required
Design and production of the	Introduce the production methods	Multimeter, electric	8	1	Integrated	Required

module circuit	and precautions; select the appropriate components for the unit circuit construction.	pyroiron, inclined pliers and other experimental equipment.				
Overall commissioning and data record of the system circuit	Introduce the commissioning method, conduct the overall circuit debugging, record relevant parameters, and test the corresponding indicators.	DC stable voltage power supply, signal source, oscilloscope and other equipment.	4	1	Integrated	Required
Test defense and report writing	Summary and defense, and write a design report, including: design purpose, requirements, indicators, block diagram, selection of components, circuit assembly, commissioning results analysis and experience, etc.	Computer	4	1	Integrated	Required
Total			20			

（二）Teaching methods

This course is a comprehensive and practical course, and pays attention to the cultivation of design ability and hands-on ability. On the basis of understanding basic theory, students are guided to independently design circuits and hands-on production, after repeated debugging, finally realize a comprehensive digital system. Topic selection guidance, working principles and welding technology are mainly classroom teaching; meanwhile, effective solutions are obtained through classroom discussion; the construction and production of unit circuits, debugging of module circuits and joint debugging of system circuits are done by students. During the process of debugging, gradually learn to investigate faults independently, improve the ability to analyze and solve problems, so as to improve the engineering practice ability.

VIAssessment

（一）Assessment method

This course adopts a comprehensive evaluation method, including classroom performance, design works, design reports, and defense tests, as shown in Table 4. Total course assessment score (percentage system) = usual score * 10% + design works * 50% + design report * 20% + defense test * 20%.

Table 4 Assessment method

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
usual score	Attendance, classroom interaction, and laboratory usage specification.	10%	Course Objectives 1、2、3、4、5
design works	Production process, performance indicators, etc.	50%	Course Objectives1、2、3、4、5
design report	Write the experimental report and submit it on time.	20%	Course Objectives1、2、3、5
defense test	Closed volume exam	20%	Course Objectives1、2、4、5

（二）Evaluation criteria

The assessment method of this course includes normal performance, design works, design report and defense. See the table 5~8in the evaluation criteria

Table 5 Normal Performance Evaluation Standard

Term	Ratio	Excellent	Good	Medium level	Pass	Fail
Atten-dance	25%	No absence, late, early leave	No absence, late, early leave situation is not more than 2 times	Not more than 1 absence, or 3 late or early leave	Not more than 2 absor 4 late or early	More than 3 absor late or early over 4
Class-room interac-tion	25%	Listen carefully, work actively, can complete the design content independently and have good	Listen carefully, work actively, complete design requirements on time and correctly express your views	Listen carefully, and can complete the design and understand the design concept on time with the help of the	Basically complete the design tasks and understand the design philosophy	Inconcentration, incomplete basic design requirements

		communication skills		teacher		
Labora-tory usage specific ation	25%	Compliance with laboratory rules and regulations and proper use of laboratory equipment	Ability to observe laboratory system and use equipment	Ability to comply with the laboratory system and use the experimental equipment properly when reminded by the teacher	Basically compliance with the laboratory system and unclear specification of equipment use	Failure to comply with the laboratory rules and regulations and damage to the equipment
Work is submit-ted on time	25%	Submit high quality work	Submit the work on time	Basically submission on time	Failure to submit on time requires urging	No submission

Table 6 Design Works Evaluation Standard

Term	Ratio	Excellent	Good	Medium level	Pass	Fail
Production process	10%	Welding point standard, reasonable wiring specification	Welding point standard, reasonable wiring	Standard relative to welding point, acceptable wiring	Coarse welding points and inadequate standard wiring	False welding and wiring are messy, not standard
Performance indicator point（1）	30%	Normal experiment phenomenon and accurate data	Normal experiment phenomenon and deviation of the data	Basically stable phenomenon and inaccurate data	Instabilityphenomenon, unmeasurable data	No experimental phenomenon, no data
Performance indicator point（2）	30%	Normal experiment phenomenon and accurate data	Normal experiment phenomenon and deviation of the data	Basically stable phenomenon and inaccurate data	Instability phenomenon, unmeasurabledata	No experimental phenomenon, no data

Overall Performance	25%	System is running normally and with accurate data	System is running normally, with measurable data	System operation is basically stable	System instability, unmeasurable data	System is not running, without data
Completion Time	5%	Located in the top 15% of the total number of people	Located in the 16%-35% of the total number of people	Located in the 36%-60%of the total number of people	Located in the61%-85%of the total number of people	Not completed at the specified time

Table 7Design ReportEvaluation Standard

Term	Excellent	Good	Medium level	Pass	Fail
Design Report	Reasonable structure, clear level, organized content arrangement, accurate problem description, full discussion, rigorous discussion, and strong logic. The text is smooth, no wrong words, the chart and marking are clear.	The structure is reasonable, clear level, organized content arrangement, accurate problem description, full, rigorous discussion, and strong logic. The text is smooth, no wrong words, the chart and marking are clear.	he structure is basically reasonable, the level is basically clear, the content arrangement is organized, the problem description is basically accurate, the analysis is basically accurate, and it has a certain logic.	The level is not clear enough, the content arrangement is not rational enough, the problem description is basically accurate, and the logic of the analysis process is not strong enough.	The content is messy, the lack of organization, there are obvious logical errors.

Table 8DefenseEvaluation Standard

Term	Excellent	Good	Medium level	Pass	Fail
Defense	Accurate concept, clear thinking, and can correctly answer the main	Correct concept, clear thinking, can better answer the main questions	The idea is general, to answer the main questions is still	Cannot answer some major questions or answer errors,	Unable to answer questions or be completely

	questions related to the design.	related to the design.	correct.	not very familiar with the design content.	unfamiliar with the design content.

VII. Recommended teaching materials and references

（一）Textbooks

[1] Yan guohong:《Guideof Digital Electronic Technology Course Design》,Self handoutof ZSTU。

（二）Refenrences

[1] 余瓔等译:《数字电子技术》（第十一版）, 电子工业出版社,2019 年出版。

[2] 寇戈,蒋立平编著:《模拟电路与数字电路》,电子工业出版社,2015 年出版。

[3] 阎石主编:《数字电子技术基础》（第六版）,高等教育出版社,2016 年出版。

[4] 康华光主编:《电子技术基础:数字部分》（第六版）,高等教育出版社,2014 年出版。

（三）Website

[1] http://www.icourse163.org/course/HUST-1001909001

Written by：Kong yanyan **Reviewed by: Jiang xusheng**

Date：2021.5.20

Syllabus of Data Structures and Algorithms

Course Name/Title：Data Structures and Algorithms **Course code**：**62947**

Course Type：(General Course，Basic Course ，**Specialized Course**)(**Compulsory Course**，Optional Course)

Total Teaching Hours： 64 (Classroom Hours: 56 Laboratory Hours or Tutorial Hours： 8)

Course Credit：4

I Course Introduction

Data structure and algorithm is a specialized course for undergraduate students majoring in computer science and technology. The course is an introduction to the basic algorithm analysis methods, some classic and useful data structures as well as the relevant algorithms, and also some classic and popular algorithms, such as the sorting algorithm. The study of this course enables students to mast features and operation of logical structures in various common data structures, design of common storage structures and the realization of typical algorithms on different storage structures in an all-round way, to acquire the ability to analyze and solve problems with knowledge of data structure and through this course. Based on this, the primary goal of the course is to cultivate students' abstract thinking ability and to construct a good foundation of design and implement complex programs, and ultimately to help the students grasp the way of solving practical problems.

II Course Objective

Course Objective 1: To cultivate the students' spirit of criticism via comparison analysis of different algorithms; To inspire students to creatively design data structures and algorithm for solving real-life problems; To establish the concept of curriculum confidence and national confidence, to cultivate students to actively explore and grasp the background and essence of the problem; Master the methods and means of solving practical problems with the knowledge learned, and explore the ways to solve problems from various angles.

Course Objective 2: To master the core concepts of classic data structures, such as the linear data structures, the tree data structures, the graph data structure, etc.; To understand the logic characteristic and storage representation of these data structures, and also the algorithm designs and implementations based on them.

Course Objective 3: To master the way of design data structures, and the ability of selecting appropriate data structure for solving problems; To cultivate the principle of designing data structures and algorithms, and the ability of designing data structure and algorithms for solving practical problems. Be familiar with the way of implementing data structure and algorithms with program languages; To be able to analyze problems, design data structures and algorithms, and finally implement algorithms; To improve the innovation and exploration abilities, the practical ability, and also the capability of independent thinking.

Course Objective 4: To understand the concepts of time and space complexity and to master the ability of algorithm analysis and evaluation; To understand the relationships between algorithm evaluation and the sustainable development of the society.

III Correlations between Course Objectives and Graduation Requirements

Table 1 Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1	1.2 Be able to make abstractions over computer engineering problems, and can establish appropriate mathematical models and solve them.	Course Objective 1,2
Graduation Requirement 4	4.1 Be able to analyze and study critical algorithms and modules in computer science, and also can verify and test them empirically.	Course Objective 3
Graduation Requirement 7	7.2 Be able to evaluate and reasonably judge the hidden danger of computer engineering practice that may cause damage to human and environment according to the actual project.	Course Objective 4

IV Correlations between Course Content and Course Objectives

（一）Theory contents（56 Periods）

Table 2 Theory Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 INTRODUCTION** (1) the aims and goals of this text; (2) some programming concepts and discrete	(1) See that how a program performs for reasonably large input is just as important as its performance on moderate amounts of input. (2) Review good programming style. (3) Summarize the basic mathematical	2	Classroom teaching; case studies; discussion	Course Objectives 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	mathematics.	background needed for the rest of the book. (4) Briefly review recursion.			
2	**Chapter 2 ALGORITHM ANALYSIS** (1) Definitions and concepts; (2) Time complexity analysis; (3) Space complexity analysis.	(1) Explain how to estimate the time and space required for a program. (2) Discuss the way of reducing the running time of a program from days or years to fractions of a second. (3) Demonstrate case studies to show the effects of varying algorithms. (4) To cultivate the students' spirit of criticism via comparison analysis of different algorithms.	6	Classroom teaching; case studies; discussion	Course Objectives 1,2,4
3	**Chapter 3 LISTS, STACKS, AND QUEUES** (1) Array List and linked list; (2) Stack; (3) Queue.	(1) Introduce the concept of Abstract Data Types (ADTs). (2) Show how to efficiently perform operations on lists. (3) Introduce the stack ADT and its use in implementing recursion. (4) Introduce the queue ADT and its use in operating systems and algorithm design. (5) To inspire students to creatively design data structures and algorithm for solving real-life problems.	6	Classroom teaching; case studies; discussion	Course Objectives 1,2,3,4
4	**Chapter 4 TREES** (1) Binary tree; (2) Tree traversal; (3) Binary search tree; (4) AVL tree; (4) Splaying tree; (5) B-tree.	(1) Explain how trees are used to implement the file system of several popular operating systems. (2) Explain how trees can be used to evaluate arithmetic expressions. (3) Show how to use trees to support searching operations in O(log n) average time, and how to refine these ideas to obtain O(log n) worst-case bounds. We will also see how to implement these operations when the data is stored on a disk. (4) Introduce different categories of binary trees and their usage contexts. (5) To cultivate the students' spirit of exploring new findings from daily life, and design solutions for solving problems.	8	Classroom teaching; case studies; discussion	Course Objectives 1,2,3,4
5	**Chapter 5**	(1) Show several methods of implementing	6	Classroom	Course

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	HASHING (1) The ADT of hash table; (2) Hash functions; (3) Application of hash tables.	the hash table. (2) Compare these methods analytically. (3) Show numerous applications of hashing. (4) Compare hash tables with binary search trees.		teaching; case studies; discussion	Objectives 2,3,4
6	**Chapter 6 PRIORITY QUEUES** (1) The ADT of priority queue; (2) implementation of priority queue; (3) application of priority queue.	(1) Efficient implementation of the priority queue ADT. (2) Uses of priority queues. (3) Advanced implementations of priority queues.	6	Classroom teaching; case studies; discussion	Course Objectives 2,4
7	**Chapter 7 SORTING** (1) Insertion sort; (2) Shell sort; (3) Heapsort; (4) Merge sort; (4) Quick sort;	(1) Introduce the idea of different sorting algorithms; (2) Explain the implementation of sorting algorithms; (3) Discuss and compare different sorting algorithms from the perspective of time complexity and space complexity. (4) To cultivate the students' spirit of criticism via comparison analysis of different algorithms.	8	Classroom teaching; case studies; discussion	Course Objectives 1,2,3,4
8	**Chapter 8 GRAPH ALGORITHMS** (1) Basic concepts; (2) Representations; (3) Traversal; (4) Topological sort; (5) Shortest path algorithm; (5) Spanning trees;	(1) Introduce the concepts of graph, and demonstrate different ways of representing graphs; (2) Show several real-life problems, which can be converted to problems on graphs; (3) Explain and discuss algorithms to solve several common graph problems. (4) Show how the proper choice of data structures can drastically reduce the running time of these algorithms. (5) To cultivate the student’s ability of abstracting real-life problems and designing solutions for solving problems.	8	Classroom teaching; case studies; discussion	Course Objectives 1,2,3,4
9	**Chapter 9 ALGORITHM DESIGN TECHNIQUES** (1) Greedy algorithms; (2) Divide and conquer;	(1) Explain the idea and general procedure of algorithms; (2) Show the algorithm implementation; (3) Discuss, in general terms, the time and	6	Classroom teaching; case studies;	Course Objectives 2,3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(3) Dynamic programming.	space complexity, where appropriate.		discussion	

(二) Experimental contents（8 Periods）

Table 3 Correlations between experimental Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Experiment 1： List** (1) Implement functions of the array based list; (2) Implement a desk calculator.	(1) To master the design and implementation of array list and stack; (2) to be able to implement array list and stack, and can use them to solve problems.	2	Classroom teaching; experiments	Course Objectives 2,3
2	**Experiment 2： Tree** (1) Draw binary tree based on the traversal sequences; (2) For each given binary search tree, you are supposed to tell if it is a legal red-black tree.	(1)To master the design and implementation of binary tree; (2) to be able to implement binary, and can use different tree data structures to solve problems.	2	Classroom teaching; case studies; discussion	Course Objectives 2,3
3	Experiment 3：Sort (1) Implement and analyze various sorting algorithms; Select two or more sorting algorithms (insertion sort, bubble sort, selection sort, shellsort, mergsort, and quicksort, and ..), (2) try to design some strategies to improve existing algorithms.	(1) To master the design and implementation of different sorting algorithms; (2) to be able to understand the advantages and disadvantages of different sorting algorithms, and can find strategies to improve some of existing algorithms.	2	Classroom teaching; case studies; discussion	Course Objectives 2,3,4
4	Experiment 4： Graph (1) Identification of an acyclic graph; (2) Find and print shortest paths	(1) To master the design and implementation of graph; (2) to be able to implement shortest path algorithms, and can use the graph data structure and relevant algorithms to solve problems.	2	Classroom teaching; case studies; discussion	Course Objectives 2,3

V Period Distribution and Teaching Modes

（一）Period distribution of theory courses

Table 4 Period distribution of theory course

Teaching / Period / Course content	Theory teaching	Exercise course	Discussion		All
Chapter 1 INTRODUCTION	2				2
Chapter 2 ALGORITHM ANALYSIS	4	1	1		6
Chapter 3 LISTS, STACKS, AND	7	1	1		9
Chapter 4 TREES	6	1	1		8
Chapter 5 HASHING	4				4
Chapter 6 PRIORITY QUEUES	4	1			5
Chapter 7 SORTING	7	1	1		9
Chapter 8 GRAPH ALGORITHMS	6	1	1		8
Chapter 9 ALGORITHM DESIGN	4		1		5
All	44	6	6		56

（二）Period distribution of experiments

Table 5 Period distribution of experimental course

Name	Contents	Environments	Period	Group size	Attributes（Basic /Comprehensive/ Design/ Creative Study）	Requirements (Compulsory /Optional)
List	Implement functions of the array based list; Implement a desk calculator.	c/c++	2	1	Basic	Compulsory
Tree	Draw binary tree based on the traversal sequences; For each given binary search tree, you are supposed to tell if it is a legal red-black tree.	c/c++	2	1	Basic	Compulsory
Sort	Implement and analyze	c/c++	2	1	Basic,	Compulsory

	various sorting algorithms; Select two or more sorting algorithms (insertion sort, bubble sort, selection sort, shellsort, mergsort, and quicksort, and ..), try to design some strategies to improve existing algorithms.				Creative Study	
Graph	Identification of an acyclic graph; Find and print shortest paths.	c/c++	2	1	Comprehensive	Compulsory

（三）Teaching Mode

This course contains both theoretical and practical contents, and the content is complex and has a certain degree of difficulty. During the teaching process, the combination of case study-driven, heuristic and discussion teaching methods is adopted, focusing on mobilizing the enthusiasm of students, and changing the original "full room" process into a process of joint exploration by teachers and students under the guidance of teachers. In addition, properly arranging some homework and exercises to help students understand the content of classroom teaching.

In order to make students fully understand the key and difficult contents of this course, additional questions can be arranged according to students' different mastering situations and practical abilities. The aim of this is to to further strengthen their practical ability, help students understand and master relevant concepts and principles, and realize students' diverse needs.

VI Assessment

Table 6 Course assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Classroom performance	Attendance Other performances (answering questions, discussion)	10%	Course Objectives 2,3
Homework	Submit homework on time	10%	Course Objectives 2,3,4
Experiments	Submit the report and source codes on time	30%	Course Objectives 2,3,4
Final exam	closed-book exam	50%	Course Objectives 2,3,4

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

Textbook

[1] Mark Allen Weiss, Data Structures and Algorithm Analysis in C, Addison-Wesley,2010

References

[1] 维斯（美国）(冯舜玺译)，数据结构与算法分析：C 语言描述(原书第 2 版). 北京：机械工业出版社, 2004

[2] Clifford A.Shaffer．A Practical Introduction to Data Structures and Algorithm Analysis（影 印版），第 2 版．北京：电子工业出版社，2002

[3] Horowitz E，Sahni S．Fundamentals of Data Structures．Pitmen Publishing Limited,1976

[4] Thomas H.Cormen．算法导论[英文版]，第 2 版．北京：高等教育出版社，2002

[5] Robert Sedgewick．算法 V（C 实现）：图算法（第三版 影印版）．北京：中国电力出版 社，2003

Written bv：Mingyue Jiang **Reviewed by：Qi Sun**

Date：2021.5.20

Syllabus of Course Project for Data Structures and Algorithms

Course Name/Title：Course Project for Data Structures and Algorithms   Course code: 62982

Course Type：(General Course，Basic Course ，**Specialized Course**)(**Compulsory Course**，Optional Course)

Total Teaching Hours： 20   (Classroom Hours: 0 Laboratory Hours or Tutorial Hours：20)

Course Credit：1

I Course Introduction

Data structure and algorithm is a specialized course for undergraduate students majoring in computer science and technology. The course is an introduction to the basic algorithm analysis methods, some classic and useful data structures as well as the relevant algorithms, and also some classic and popular algorithms, such as the sorting algorithm. The study of this course enables students to mast features and operation of logical structures in various common data structures, design of common storage structures and the realization of typical algorithms on different storage structures in an all-round way, to acquire the ability to analyze and solve problems with knowledge of data structure and through this course. Based on this, the primary goal of the course is to cultivate students' abstract thinking ability and to construct a good foundation of design and implement complex programs, and ultimately to help the students grasp the way of solving practical problems.

II Course Objective

Course Objective 1: To cultivate the students' spirit of criticism via comparison analysis of different algorithms; To inspire students to creatively design data structures and algorithm for solving real-life problems; Master the methods and means of solving practical problems with the knowledge learned, and explore the ways to solve problems from various angles..

Course Objective 2: Be familiar with linear data structure, tree data structure and mesh data structure, be able to master the typical algorithms of these logical structures, and have a good grasp of the complexity of the algorithm, realize the theory and algorithm learned in comprehensive data structure and algorithm, analyze and solve some application problems in practical program software.

Course Objective 3: Be able to choose the appropriate data structure and algorithm, and solve the practical problems through design and development; According to the actual situation of complex engineering problems, clear design objectives, let the students be able to analyze problems, carry out scheme design and realize the scheme under the constraints of society, health, safety, law, culture and environment.

Course Objective 4: To build a solid theoretical foundation for students to independently analyze and solve practical problems related to data structure selection and algorithm design and analysis in the future; To cultivate students with the ability of preliminary innovative design and good sense of teamwork and communication skills.

III Correlations between Course Objectives and Graduation Requirements

Table 1 Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 3：Design/develop solutions	3-3 According to the design scheme, fully consider the cost performance, follow the software engineering specification, and realize the computer system that meets the requirements;.	Course Objective 1,2
Graduation Requirement 4：Research	4-3 Be able to analyze and interpret experimental data, and get reasonable and effective conclusions by information synthesis.	Course Objective 3
Graduation Requirement 9：Individuals and teams	9-2 Be able to manage, coordinate and organize the work of team members as team leader.	Course Objective 4

IV Correlations between Course Content and Course Objectives

Table 2 Correlations between experimental Course Content and Course Objectives

(each student is required to select one project)

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Travel information advisor** Please design and implement a program to provide travel information for users with different	(1) To master the design and implementation of graph; (2) to be able to leverage graph algorithms to solve real-life problems. (3) To inspire students to creatively design data structures and algorithm for	2	Classroom teaching; experiments	Course Objectives 1,2,3,4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	preferences, including (1) shortest travel time; (2) lowest travel cost; (3) minimum number of transition.	solving real-life problems.			
2	**Experiment 2： How to fill oils** With highways available, driving a car from Hangzhou to any other city is easy. But since the tank capacity of a car is limited, we have to find oil stations on the way from time to time. Different oil station may give different price. You are asked to carefully design the cheapest way for filling oils.	(1) To master the design and implementation of different data structures and algorithms; (2) to be able to select proper data structures and algorithms for satisfying the given requirements. (3) To inspire students to creatively design data structures and algorithm for solving real-life problems. To cultivate the students' spirit of criticism via comparison analysis of different algorithms.	2	Classroom teaching; case studies; discussion	Course Objectives 1,2,3,4
3	Experiment 3 ： string compression and decompression Your program has to implement the following functionalities: (1) Given a string as input, the program output is a sequence of bits (compression); (2) Given a sequence of bits as input, the program output is a string (decompression). You have to design algorithms, select or design data structures, and prepare initial data to support the execution of your program.	(1) To master Huffman tree; (2) to be able to design and implement Huffman tree to support string compression and depression. (3) To inspire students to creatively design data structures and algorithm for solving real-life problems.	2	Classroom teaching; case studies; discussion	Course Objectives 1,2,3,4

V Period Distribution and Teaching Modes

（一）Period distribution of experiments

Table 3 Period distribution of experiments

Name	Contents	Environments	Period	Group size	Attributes（Basic /Comprehensive/ Design/ Creative Study）	Requirements(Compulsory /Optional)
Course design	Understanding and selecting projects.	c/c++	2	1	Comprehensive	Compulsory
Course design	Requirement analysis and overall system design.	c/c++	2	1	Comprehensive	Compulsory
Course design	Data structure design, algorithm design and implementation.	c/c++	16	1	Comprehensive	Compulsory
Total			20			

（二）Teaching Mode

This experiment is mainly based on laboratory study. The teacher needs to explain to the students the nature, tasks, requirements, course arrangement and progress, contents to be assessed, experimental rules and laboratory safety system of the course.

Each student has a computer, and the students must complete the designated experimental project. In order to cultivate students' practical operation ability and teamwork spirit, students can ask two people to complete the topic selection, preview, experiment and experiment report writing.

VI Assessment

Table 4 Course assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Classroom performance	Attendance Other performances (answering questions, discussion)	10%	Course Objectives 2,3
Operational performance	Solve problem and complete	10%	Course Objectives 2,3

	the project on time.		
Oral presentation	Prepare a ppt based on system design, implementation and solutions, and do a presentation in English.	30%	Course Objectives 2,3,4
Final report	Write an experimental report based on the project.	50%	Course Objectives 1、2,3,4

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

（一）Textbook

无

（二）References

[1] 维斯（美国）(冯舜玺译)，数据结构与算法分析：C 语言描述(原书第 2 版). 北京：机械工业出版社, 2004

[2] Clifford A.Shaffer．A Practical Introduction to Data Structures and Algorithm Analysis（影 印版），第 2 版．北京：电子工业出版社，2002

[3] Horowitz E，Sahni S．Fundamentals of Data Structures．Pitmen Publishing Limited,1976

[4] Thomas H.Cormen．算法导论[英文版]，第 2 版．北京：高等教育出版社，2002

[5] Robert Sedgewick．算法 V（C 实现）：图算法（第三版 影印版）．北京：中国电力出版 社，2003

Written bv：Mingyue Jiang **Reviewed by：Qi Sun**

Date：2021.5.20

Syllabus of Introduction to Database System

Course Name/Title： Introduction to Database System **Course code：62940**

Course Type：(Specialized Course)(Compulsory Course)

Total Teaching Hours： 48 (Classroom Hours:33 Laboratory Hours or Tutorial Hours 15)

Course Credit： 3

I Course Introduction

Introduction to Database System is a compulsory specialized course for the Computer Specialized undergraduates .It is a course based on both basic principles and knowledge. Students are required to have an overview of the basic concepts of database theory include the major features of databases and database management systems, data models, different categories of database languages, the different categories of people who interact with databases, three level architecture, data independence, the basic structure of the relational model, integrity constraints in relational model, master the use of the basic relational algebra operations, use SQL skillfully, learn how to use E-R diagram to represent conceptual data model and translate the E-R model to logical model, learn how to normalize the form to improve the quality of the database, learn how to maintain and protect the database and so on.

II Course Objective

Course objective 1： Through this course study, students should understand the concepts of Database systems, the hierarchical data model, network data model, the relational data model, integrity, and so on. After the course study, the students must grasp the fundamental knowledge about database in order to study database programming in further course. Based on the concept of integrity, students should know the operations must be conducted under rules.

Course objective 2： After this course study, the students should be able to use E-R model to design tables. They can decompose relations based on the normalization theory. They can create tables with integrities. Based on normalization, students will know that they should be in the professional norm when they are in the future career.

Course objective 3： After this course study, the students should be able to install, profile the Mysql. They gain the abilities to use language of SQL to retrieve the information from Databases. They can write the correct SQL statements for practical cases. They know how to create and use views and index to improve the performance.

Course objective 4： After this course study, the students should be able to analyze the real world and feel good taste for English version of text books. They should be able to take actions in the group effectively.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation requirement 1： Problem analyzing	2.1 be able to distinguish the compound computer engineering problem and express the requirement and process of the compound computer engineering problem explicitly after investigation.	Course Objective 1
Graduation requirement 2： Design to resolve the problem	3.3 be able to consider the cost performance according to the design; obey the software engineering normalization to realize the required computer system.	Course Objective 2
Graduation requirement 3： Use tools	5.2 be able to use integrated tools and open-sourced software to develop, debug and test the system and understand the limitations at the same time.	Course Objective 3
Graduation requirement 4：project management	11.1 understand the principles and methods in the management of engineering project and take actions in the group effectively.	Course Objective 4

IV Correlations between Course Content and Course Objectives

No.	Course content	Teaching requirement	period	Teaching methods	Course objectives
1	**Unit 1：Introduction**	(7) Fundamental Database Concepts Be able to describe what is database management systems (DBMS). Tell the History of database systems.	3	Lecture, on line lecture, discussi	Course objective 1

No.	Course content	Teaching requirement	period	Teaching methods	Course objectives
2		(8) Database Users Tell the difference between concepts of End users (casual users, naïve users), application programmers and Database Administrators. (9) Overview of Relational and Object Relational DBMS Be able to tell about the relational model, basic SQL query language, the object-relational model, programs to access a database, database design, integrity, view and security. From the concept of constrains, guide students to establish the core values of Chinese socialism for honesty and civilization. Dishonesty should be tested by rules. So students should understand the rules and obey the disciplines.		on	
2	**Unit 2：Relational Model**	(5) Be able to describe relational models of database. (6) Be able to define table consisting column names which have types and domains, keys, superkeys, candidate keys, null values. (7) Be able to express in relational algebra on relational database. On the fact that the father of relational model was given the Turing Award, encourage students establish the core values of Chinese socialism and devote themselves to their professional job in the future.	3	Lecture, on line lecture, discussion	Course objective 1、4

No.	Course content	Teaching requirement	period	Teaching methods	Course objectives
3	**Unit 3：ER model**	(1) Be able to define E-R Concepts Be able to tell about Entities, Attributes, and E-R Diagrams; Relationships among Entities (2) Be able to distinguish Details of E-R Modeling including: 1-1, N-1, N-N Relationship; Transformation rules. (3) Case study. Be able to calculate Cardinality of attributes; be able to tell the definition of Weak Entity; Case study of simple airline reservation database. The conversion from E-R diagram to relational model must be operated under some rules. It indicate that students should work following the processional rules in their future carrer.	6	Lecture, on line lecture, discussion	Course objective 2、4
4	**Unit 4：relational database design theory**	(1) Functional dependency Be able to define the concept of redundancy, anomaly, normalization, functional dependency (FD). (2) BCNF Be able to define the nontrivial FD, unavoidable FD. (3) Multi_valued dependency Be able to distinguish the multi_valued dependency. (4) 4NF,3NF Be able to tell the difference of 4NF, 3NF. Be able to decompose following the decomposing rules.	3	Lecture, on line lecture, discussion	Course objective 2、4

No.	Course content	Teaching requirement	period	Teaching methods	Course objectives
5	**Unit 5: Basic SQL Query Language**	(1) SQL history Be able to describe SQL capabilities, SQL history and dialects. (2) Select statement Ba able to tell the standard typographical conventions for SQL statement syntax; be able to write simple select statement. (3) Subqueries Be able to use The IN predicate, The Quantified Comparison Predicate, The EXISTS Predicate. (4) Advanced SQL Syntax Be able to use INTERSECT, UNION, EXCEPT operators; JOIN forms. (5) Set Functions in SQL Be able to describe the set functions such as COUNT, SUM, AVG, MAX, and MIN. (6) Groups of Rows in SQL Be able to use Group by, having search_condition. (7) Create , Insert, Update, and Delete Statements Be able to write statement to Create table, insert rows, update rows and delete rows statements. Through the from clause, encourage students to establish their sense of belongings and national pride. Let them know that the motherland will always be their strong backing.	9	Lecture, on line lecture, discussion	Course objective 3
6	**Unit 6：operation on a Database with algebra**	(1) Algebra language Be able to distinguish operators on	3	Lecture, on line	Course objective

No.	Course content	Teaching requirement	period	Teaching methods	Course objectives
7		relation. (2) Operate on relations with algebra Be able to tell the algebra operators with the function of select,projection,cross-product,natural join,set operation.		lecture, discussion	1、2
7	**Unit 7：constraints, trigger**	(1) Integrity Constraints Be able to define the concept of Integrity constraints in Create table statement; primary keys, foreign keys and referential integrity. (2) triggers be able to write statement to create and drop triggers;		Lecture, on line lecture, discussion	Course objective 3
8	**Unit 8：Indexing and view**	(1) Views Be able to write statement to create and drop views; the value of view; the usage of views. (2) Concept of Indexing Be able to describe database index. Be able to Create index statements	3	Lecture on line lecture ,discussion	Course objective 3
9	**Unit 9：transaction and security**	(3) transaction be able to describe the concept of concurrency. (4) security be able to describe backup, recovery, deadlock.	3	Lecture, on line lecture, discussion	Course objective 1、4

V Period Distribution and Teaching Modes

（1）Period Distribution of class hours

Fig Period Distribution of class hours for theoretical content

Teaching method / class hours / content	lecture	Exercise lesson	discussion	备注	total
Unit 1：Introduction	3				3
Unit 2：Relational Model	2	1			3
Unit 3：ER model	2		1		3
Unit 4：relational database design theory	3				3
Unit 5：Basic SQL Query Language	7	1	1		9
Unit 6：operation on a Database with algebra	3				3
Unit 7：constraints, trigger	3				3
Unit 8：Indexing and view	**3**				**3**
Unit 9：transaction and security	**3**				**3**
total	**29**	**2**	**2**		**33**

Fig Period Distribution of practice hours

Name of experiment	Content point	Experiment environment	hours	Person for each group	property（basic/comprehensive/design/innovation）	requirement(required/optional)
Experiment 1 DBMS data definition	(1) Setup database with create (2) drop table	Computer with mysql	3	1	basic	required
Experiment 2 query operation of DBMS	(1) Various select statement	Computer with mysql	3	1	basic	required
Experiment 3 compound query operation of DBMS	Statement to operate data in a comprehensive way	Computer with mysql	3	1	comprehensive	required
Experiment 4 constrains and triggers	(1) Know how to set up constrains (2) Know how to	Computer with mysql	3	1	comprehensive	required

	create triggers (3) Know how to delete triggers					
Experiment 5 index and view	(1) Know how to set up index (2) Know how to set up view	Computer with mysql	3	1	basic	required

（2）teaching method

In this course, teacher should lecture in English. It is better to use case teaching. Teacher should encourage students to use online learning after class and design an example database such as student information management. Teacher should adopt question –result mode to give students strong impression. Teacher should use multimedia classroom to teach this course. The amount of time should be left to discussion for all students. The discussion class can be grouped. One group may quest; the other may answer the question.

VI Assessment

(Assessment content and methods must correlate with course objectives.)

Assessment methods		Assessment requirements	Assessment weighting	Evaluation of Course objectives
Daily work	Class performance	Evaluate according to the attendance and presentation	30%	Course objectivity 1、2、3、4
	homework	Evaluate according to the homework quality		
	project	Evaluate according to the quality and completeness		
	Online-learning	Evaluate according to the numbers of viewing video, the lasting time for the viewing of video.		
experiment		Evaluate on the practice performance, experiment report and experiment operation.	30%	Course objectivity 3、4
Final exam (Without reference)		Evaluate on the accuracy and completeness of the answer	40%	Course objectivity 1、2、3

VII Textbooks and References

（1）Textbook

Jeffrey D.Ullman,A First Course in DATABASE SYSTEMS,China Machine Press,2013

（2）Reference book

[2] ABRAHAM sIBERSCHATZ,Database System Concepts,高等教育出版社,2014

[3] C.J.Date An introduction to database systems.科学出版社，2013。

Written by：Li liu **Reviewed by: Wei shen**

Date：2021.05.20

《认知实习》教学大纲

课程中文名称：认知实习　　　　　　　　　　课程代码：62696

课程英文名称：Cognition Practice

课程类别与性质：（学科基础课）（必修）

总学时：20 学时（其中课外实践 20 学时）

学　分：1

先修课程：C 程序设计

适用专业：计算机科学与技术

开课系(室)：计算机科学与技术系

一、课程简介

认知实习是计算机科学与技术及相关专业本科生的一门专业基础课程，其主要内容是通过组织学生到大型企事业单位、计算机科技公司、软件公司开展实地参观、学习，通过本次认知实习，了解所学理论知识在实际中的应用情况及新的技术应用发展方向，拓宽知识面，为后续的学习打下良好的基础，加深对专业的理解。学生针对计算机专业理论、技术和应用进行实地参观、学习，树立正确职业道德观，培养社会责任感和使命感。通过了解企业文化培养团队合作精神。

二、课程教学目标

课程目标 1：通过组织学生到大型企事业单位、计算机科技公司、软件公司，针对计算机专业理论、技术和应用进行实地参观、学习，树立正确职业道德观，培养社会责任感和使命感；通过了解企业文化培养团队合作精神。

课程目标 2：通过对计算机公司、软件公司实地参观、学习，了解、学习企业中计算机系统及软件产品全周期开发流程、基本设计/开发方法和技术，了解影响设计目标和技术方案的各种因素，了解企业中团队协作开发的基本方法。

课程目标 3：通过计算机公司和软件公司技术人员的实地讲解，熟悉与计算机相关的技术标准、知识产权、信息安全规范、产业政策和法律法规，并理解应承担的社会责任和法律责任，培养社会责任感和使命感。

课程目标 4：了解计算机公司、软件公司员工职业道德和规范，理解诚信公正的工程职业道德和规范的重要性，并能在计算机工程实践中自觉遵守。

三、课程教学目标与毕业要求的对应关系

毕业要求	毕业要求指标点	课程目标
毕业要求 3	3-1 掌握计算机系统及软件产品全周期开发流程、基本设计/开发方法和技术，了解影响设计目标和技术方案的各种因素。	课程目标 1、2
毕业要求 6	6-1 熟悉与计算机相关的技术标准、知识产权、信息安全规范、产业政策和法律法规，并理解应承担的责任。	课程目标 1、3
毕业要求 8	8-2 理解诚信公正的工程职业道德和规范，并能在计算机工程实践中自觉遵守。	课程目标 1、4

四、实习内容

通过在业内比较知名的计算机公司、软件公司实地参观、学习，在公司实地开展座谈、讲座等活动，了解、学习计算机系统及软件产品全周期开发流程、基本设计/开发方法和技术，了解影响设计目标和技术方案的各种因素。

通过计算机公司和软件公司技术人员的实地讲解，为学生现场答疑，熟悉与计算机相关的技术标准、知识产权、信息安全规范、产业政策和法律法规，并理解应承担的社会责任和法律责任，培养社会责任感和使命感。

通过现场讲解，了解计算机公司、软件公司员工职业道德和规范，理解诚信公正的工程职业道德和规范的重要性，并能在计算机工程实践中自觉遵守。

学生撰写认知实习报告，将所闻所见按照格式整理成文，主要从计算机系统及软件产品开发技术、信息安全规范、产业政策、法律法规、职业道德规范和计算机技术应用发展等方面总结，撰写学习体会。

通过发放问卷，及时了解学生认知实习情况，并做出教学方式方法的调整，及时答疑解惑。

五、课程学时分配及教学方法

(一) 课程学时分配：

内容	课时
参观计算机公司、软件公司	8
讲座	5
视频学习、问卷调查、回答问题	3
撰写实习报告	4

（二）教学方法：

通过在业内比较知名的计算机公司、软件公司实地参观、学习，在公司实地开展座谈、讲座、观看视频、现场答疑等活动，了解、学习计算机系统及软件产品全周期开发流程、基本设计/开发方法和技术，了解影响设计目标和技术方案的各种因素；熟悉与计算机相关的技术标准、知识产权、信息安全规范、产业政策和法律法规，并理解应承担的社会责任和法律责任，培养社会责任感和使命感。通过现场讲解，了解计算机公司、软件公司员工职业道德和规范，理解诚信公正的工程职业道德和规范的重要性，并能在计算机工程实践中自觉遵守。

学生撰写认知实习报告，将所闻所见按照格式整理成文，主要从计算机系统及软件产品开发技术、信息安全规范、产业政策、法律法规、职业道德规范和计算机技术应用发展等方面总结，撰写学习体会。

学生通过网络与企业员工进行现场互动，通过发放问卷，及时了解学生认知实习情况，并做出教学方式方法的调整，及时答疑解惑。

六、课程考核

考核方式或途径	考核要求	考核权重	对应课程目标
平时成绩	考勤、网络互动、实习表现、调查问卷	40	课程目标 1，4
认知实习报告	结构合理、层次清楚，内容安排有条理，问题描述准确，论述充分、严谨，逻辑性强。文字表述流畅、 无错别字。	60	课程目标 1，2，3，4

七、推荐教材及参考资料

（一）网络资源：（本课程网址或主要网络资源链接）

[1]网络资源 https://www.mosoteach.cn

执笔：许立成　　　　审稿：夏劲松　　　　审定：信息学院教学委员会

制（修）订时间：2021 年 5 月 20 日

Syllabus of Computer Architecture Principles

Course Name/Title： Computer Architecture Principles **Course code：** 62921

Course Type：Specialized Course，Compulsory Course

Total Teaching Hours： 64 hours (Classroom Hours: 56 hours Laboratory Hours or Tutorial Hours: 8 hours)

Course Credit： 4

I Course Introduction

This course is designed to introduce computer science students to the basics of computer organization and architecture. Topics include an overview of computer evolution and performance, bus, memory, I/O, computer arithmetic, instruction set, RISC and parallel processing, micro-programmed control unit. This course is a major core course for computer science students. Through the study of this course, student will exhibit competence in a wide variety of topics pertaining to computer organization and architecture, including: Understanding all basic performance characteristics of computer systems (including processor, memory, interconnection parts etc.); Recognize the relationship between various components and their interconnection; Understand the organization and structure of modern-day computers; Become more aware of the recent developments in computer technology. In addition, it lays a solid theoretical foundation for students' follow-up courses and the ability to independently analyze and solve practical problems in the computer field in the future.

II Course Objective

Course Objective1: Cultivate students' teamwork spirit by explaining how the various components of the computer work together; Enable students to establish dialectical thinking, by telling the design methods of the computer; Cultivate students' patriotism, enable them to establish lofty ideals, and the determination and confidence to fight for the prosperity of the country, by introducing the development of computer technology.

Course Objective 2: Familiar with the basic concepts and coding methods of various information inside the computer, master the basic theories and methods of computer arithmetic and logic operations.

Course Objective 3: Master the basic working principles of the main components of the computer (ALU, memory, and controller) and the basic principles of their interconnection with each other and the outside world. Understand the basic principles of RISC and pipelines.

Understand the basic analysis methods and calculation methods of the comparison and evaluation of the main components of the computer and the whole machine.

Course Objective 4: Master the design method and configuration method of the main components of the computer and the whole machine, understand the optimization strategy of the computer organization and architecture, and be able to evaluate the reliability and execution efficiency of the computer performance.

Course Objective 5: Understand the evolution and development of computers and their main components, understand the development status of computers and their main components in my country, and enhance the sense of responsibility for national rejuvenation and social progress.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1: Engineering Knowledge	1-2 Be able to abstract the processing process of computer engineering problems, establish appropriate mathematical models and solve them.	Course Objective 1,2
Graduation Requirement 2: Problem Analysis	2-3 Be able to use basic principles of mathematics, natural sciences and computer science to analyze the main factors affecting computer complex systems, demonstrate the rationality of solutions and obtain effective conclusions.	Course Objective 1,3
Graduation Requirement 3: Environment and Sustainable Development	7-2 Be able to evaluate and reasonably judge the hidden dangers of computer engineering practices that may cause harm to humans and the environment for actual projects.	Course Objective 1,4
Graduation Requirement 4: Professional Norms	8-1 Understand China's national conditions, have a certain degree of humanity, safeguard national interests, and have a sense of responsibility to promote national rejuvenation and social progress.	Course Objective 1,5

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 Overview**	(1) Define the basic concepts of architecture,	6	Classroom	Course

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(1) Organization and Architecture, Structure and Function (2) A Brief History of Computers (3) Designing for Performance (4) Performance Assessment	organization, structure, function of the computer; (2) Describe the development of the computer has gone through four stages and feature of each stage; (3) Explain Moore’s law; (4) List the approaches to improve microprocessor speed and balance the performance of different components; (5) Analyze the process and parameters of performance assessment; (6) Display the domestically developed Huawei Kirin processor and the Chinese supercomputer, cultivate students' patriotism, enable them to establish lofty ideals.		teaching and discussion	Objectives 1,3,5
2	**Chapter 2 A Top Level View of Computer Function and Interconnection** (1) Computer Components (2) Computer Function (3) Interconnection Structures (4) Bus Interconnection	(1) Define the top-level components and basic instruction cycle and I/O function; (2) Describe the process of program execution and Interrupt; (3) Describe the interconnection structure top-level components; (4) List the elements of bus design; (5) Explain how the various components of the computer work together for Cultivating students’ teamwork spirit.	4	Classroom teaching and discussion	Course Objectives 1,3,4
3	**Chapter 3 Computer Arithmetic** (1) The Arithmetic and Logic Unit (2) Integer Representation (3) Integer Arithmeti (4) Floating-Point Representation (5) Floating-Point Arithmetic	(1) Introduce ALU hardware knowledge,Am2901; (2) Describe the weight expansion method for number system notation and the number converting method; (3) List three kinds of integer representation and Integer addition and subtraction; (4) Explain the feature of twos complement representation; (5) Describe multiplication of unsigned and signed, the Booth’s algorithm; (6) Describe restoring twos Complement Division and addition and subtraction alternate method;	12	Classroom teaching and discussion	Course Objectives 2,3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		(7) Explain the division achieved by multiplication; (8) Describe floating-point numbers arithmetic;			
4	**Chapter 4 Cache Memory and Internal Memory Technology** (1) Semiconductor Main Memory (2) Error Correction (3) Computer Memory System (4) Cache Memory Principles (5) Elements of Cache Design	(1) Introduce the organization of DRAM and SRAN; (2) Explain memory capacity extension methods and interleaved memory; (3) Describe the hard failure and soft error, the Hamming code; (4) List the key characteristics of computer memory system; (5) Define the memory hierarchy; (6) List mapping function and replacement algorithms; (7) Describe write policy, line size, number of caches and unified versus split caches; (8) Explain the hierarchical structure of computer storage systems , mapping methods for enabling students to establish dialectical thinking.	8	Classroom teaching and discussion	Course Objectives 1,3,4
5	**Chapter 5 Instruction Sets** (1) Machine Instruction Characteristics (2) Types of Operands and Operations (3) Addressing (4) Instruction Formats	(1) Introduce machine instruction characteristics and element of instruction set design; (2) Describe types of operands; (3) Explain common Instruction Set Operations; (4) List advantage and disadvantage of different addressing mode; (5) Analyze how to design instruction format; (6) List some type of instruction formats;	4	Classroom teaching and discussion	Course Objectives 3,4
6	**Chapter 6 Processor Structure and Function** (1) Processor and Register Organization (2) The Instruction Cycle and Instruction Pipelining (3) Instruction Execution Characteristics (4) Reduced Instruction Set	(1) Introduce processor organization and how to design register organization; (2) Define instruction pipelining and . explain how to evaluate pipeline performance; (3) Explain instruction execution; (4) Explain reduced instruction set architecture and how to use RISC pipelining, the RISC versus CISC controversy.;	10	Classroom teaching and discussion	Course Objectives 1,3,4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	Architecture (5) Instruction-Level Parallelism and Superscalar Processors	(5) Define superscalar processor and approach, instruction-level parallelism and machine parallelism; (6) List design issues for superscalar implementation; (7) List design methods for enabling students to establish dialectical thinking.			
7	**Chapter 7 Control Unit Operation** (1) Micro-operations (2) Control of the Processor (3) Hardwired Implementation (4) Basic Concepts of Microprogrammed Control (5) Microinstruction Sequencing (6) Microinstruction Execution (7) TEC-2 Controller	(1) Define the concept of Micro-operations; (2) Introduce the process of implementing the Instruction by way of Micro-Operations; (3) Define control unit functional requirements and control signals; (4) Introduce internal processor organization of control unit; (5) Explain design considerations for microinstruction sequencing; (6) List method of address generation; (7) Explain microinstruction encoding; (8) Introduce organization of AM2910; (8) Explain implement process of a micro program in TEC-2；	12	Classroom teaching and discussion	Course Objectives 3,4

V Period Distribution and Teaching Modes

a. Course hours allocation

(1) Class hour distribution table of theoretical teaching content

Teaching Mode / Teaching Hours / Teaching Content	Theoretical Teaching	Exercise Class	Discussion Class	Remarks	Subtotal
Chapter 1 Overview	6				6
Chapter 2 A Top Level View of Computer	3	1			4
Chapter 3 Computer Arithmetic	12	1			12
Chapter 4 Cache Memory and Internal Memory Technology	8				8
Chapter 5 Instruction Sets	3	1			4
Chapter 6 Processor Structure and Function	10				10

Chapter 7 Control Unit Operation	10	1	1		12
Total	**51**	**4**	**1**		**56**

(2) Class hour distribution table of experimental teaching content

Experiment Name	Synopsis	Main Equipment or Experimental Environment	Experiment Hours	Persons per group	Experiment Attributes	Request (Required or Optional)
Experiment 1: Arithmetic Unit	Implement the arithmetic logic operation of the arithmetic unit and	TEC-2 simulation software, TEC-2 model machine，oscilloscope and PC	2	2	Basic	Required
Experiment 2: Main memory expansion	Implement memory expansion configuration and verify it	TEC-2 simulation software, TEC-2 model machine , 6116 chips and PC	2	2	Comprehensive	Required
Experiment 3: Microprogram controller	Design the microprogram of a new instruction and verify it	TEC-2 simulation software, TEC-2 model machine and PC	4	2	Design	Required

b. Teaching method

This course is a very theoretical and practical course, the content is complex and cumbersome. the key content should be paid attention to highlight in the teaching method. A certain assignment and exercises should be arranged and explained to help students to understand the content of classroom teaching.

In order to enable students to fully understand the focus of this course, difficult content, the use of multimedia and providing some supporting information could improve teaching efficiency. Appropriate arrangement of extra-curricular practice could help students understand the relevant concepts and principles.

VI Assessment

Assessment Methods or Approaches		Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
regular grade	Classroom performance	Evaluate according to the attendance rate, classroom performance	30%	Course Objective 1,2,,3,4
	Homework	Evaluate according to the times of handing in, accuracy and quality of assignments		

Experiment grade	Evaluate according to classroom performance, program design, and experimental report quality	20%	Course Objective 2, 3, 4
Final exam grade (close book)	Evaluate according to the correctness of questions and problem-solving ideas	50%	Course Objective 2, 3, 4

VII Textbooks and References

1. Textbook

[1] William Stallings. Computer Organization and Architecture Designing for Performance (Eighth Edition). Pearson Education.

[2] Junsong Li. Guide for Experiment of Computer Architecture Principles. school printing,

2. References

[1] Carl Hamacher. Computer Organization(Fifth Edition). McGraw-Hill

[2] Arnold S.Berger. Hardware and Computer Organization. Elsevier

Written by：Junsong Li **Reviewed by: Jianlong Xu**

Date：2021.05.20

Syllabus of Curriculum Design for Computer Architecture Principles

Course Name/Title： Curriculum Design for Computer Architecture Principles

Course code： 62988

Course Type： Specialized Course，Compulsory Course

Total Teaching Hours： 20 (Laboratory Hours or Tutorial Hours: 20)

Course Credit： 1

I Course Introduction

Curriculum Design for Computer Architecture Principles is a compulsory practical course for student of computer science and technology. Through learning of this course, students should deeply understand the execution process of the various instructions and the composition of the controller, moreover further grasp specific knowledge of the design microprogram instruction and the concept of dynamic microprogramming , complete computer instruction system design and debug. This should improve students' ability to analyze and solve problems related to computer composition and architecture and computer-related applications.

II Course Objective

Course Objective1: Cultivate students' teamwork spirit and sense of innovation by setting up microcode to control the coordination of the various components of the computer and teaming up to design new instructions；Cultivate students' scientific research spirit, by designing high-quality instruction microprograms with debugging and testing their functions.

Course Objective 2: Understand and master the components and functions of the TEC-2 model machine, their interconnection relations, and the TEC-2 instruction format. Familiar with the instruction design process of TEC-2 and be able to design corresponding microprograms according to the functional requirements of specific instructions. Understand the related factors that affect microprograms and microinstruction functions and performance.

Course Objective 3: Understand and master the related knowledge of TEC-2 instruction system and its microprogram design. Familiar with the hardware and software environment of TEC-2 for instruction design and development, based on this environment, formulate and implement instruction design plan.

Course Objective 4: Familiar with the use of TEC-2 simulation software and hardware control, compile test programs, realize the debugging of microprograms, and verify and analyze test results. According to the analysis results, understand the limitations of TEC-2's hardware and software environment.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1: Design/Develop Solutions	3-1 Master the full-cycle development process, basic design/development and technology of computer systems and software products, and understand various factors that affect design goals and technical solutions.	Course Objective 1,2
Graduation Requirement 2: Research	4-2 Based on scientific principles and using scientific methods, for the overall realization of complex computer systems, be able to formulate experimental plans, build experimental systems, and conduct experiments.	Course Objective 1,3
Graduation Requirement 3: Use Modern Tools	5-1 Be able to use software and hardware simulation tools to verify computer-related theories, simulate and analyze system design schemes, and understand its limitations.	Course Objective 1,4

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**1. Design Instruction Format** (1) Instruction execution process (2) Controller organization and working principle (3) Microprogramming	(1) Analyze and understand the various components of TEC-2 and their interconnection; (2) Analyze the function and composition of each component of TEC-2; (3) Define and explain the instruction format of TEC-2; (4) Analyze the functions of the new instructions; (5) Design the instruction format of the new instructions; (6) Teaming up to design new instructions in order to cultivate students' teamwork spirit and sense of innovation by teaming up to design new instructions	4	Classroom teaching and discussion	Course Objectives 1,2,3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
2	**2. Design Microprogram** (1) Microinstruction format (2) AM2910 chip (3) TEC-2 simulation software	(1) Design micro-instruction function, according to the designed instruction format; (2) Design the microcode of the microinstruction, according to the function of the microinstruction; (3) Set and select the microprogram entry address of the instruction; (4) Load the microprogram into the control memory by using TEC-2 simulation software; (5) Setting up microcode to control the coordination of the various components of the computer in order to cultivate students' sense of innovation by.	8	Classroom teaching and discussion	Course Objectives 1,2,3
3	**3. Test Instruction** (1) Test procedure of instruction function (2) The basic method of instruction function test (3) TEC-2 simulation software	(1) Master the software testing procedures; (2) Design test procedures for new instructions; (3) Set up experimental data for testing; (4) Display the test results and analyze the performance of the instructions; (5) Repeated testing and modification of microprograms to cultivate students' scientific research spirit.	8	Classroom teaching and discussion	Course Objectives 1,2,3,4

V Period Distribution and Teaching Modes

a. Course hours allocation

Class hour distribution table of experimental teaching content is as follows:

Experiment Name	Synopsis	Main Equipment or Experimental Environment	Experiment Hours	Persons per group	Experiment Attributes	Request (Required or Optional)
1. Design Instruction Format	Analyze the instruction function; Design several new machine instruction formats.	Hardware laboratory, TEC-2 experimental system and simulation	4	1-2	Design	compulsory

		software				
2.Design Microprogram	According to the format of the TEC-2 machine instruction and the usage of AM2910 chip, design microinstruction; Allocate the entry address of microprogram and load microprogram into the control memory.	Hardware laboratory, TEC-2 experimental system and simulation software	8	1-2	Design e	compulsory
3.Test Instruction	Learn the basic steps and methods of testing instruction function, and design test procedures; Design experimental data, test and debug instruction function; Complete experimental report.	Hardware laboratory, TEC-2 experimental system and simulation software	8	1-2	Design	compulsory

b. Teaching method

This experiment is mainly done in the laboratory. Teachers are required to clarify the nature of the course, tasks, requirements, curriculum and progress, the need to assess the content, experimental specification and laboratory safety and so on.

One computer per student, students must complete the specified experimental project. In order to cultivate students hands-on ability and team spirit, students can ask two people to complete the preview, experimental and experimental report writing.

This course can be previewed and reviewed in the spare time through experimental materials. According to the textbook recommendation, students can access the related materials from website, read these materials by self-studying to consolidate knowledge learned in class. Students can independently learn interactive simulations software, complete the relevant operation and analyze the results.

VI Assessment

Assessment Methods	Assessment Requirements	Assessment	Evaluation of Course

or Approaches		Weighting	Objectives
Classroom performance	Attendance, classroom interaction, and laboratory use specifications.	20%	Course Objective 1, 2, 3, 4
Design works	Evaluate according to the instruction design scheme and the comprehensive performance of the instruction.	30%	Course Objective 1, 2, 3, 4
Design report	Write experiment report and submit it on time.	30%	Course Objective 2, 3, 4
Oral exam	Open-book exam	20%	Course Objective 2, 3, 4

VII Textbooks and References

1. Textbook

[1] Junsong Li. Guide for Experiment of Computer Architecture Principles. school printing, 2017

2. References

[1] William Stallings. Computer Organization and Architecture Designing for Performance (Eighth Edition). Pearson Education, 2010.

[2] 王诚. 计算机组成原理实验指导书. 清华大学出版社，2005.

Written by：Junsong Li **Reviewed by: Jianlong Xu**

Date：2021.05.20

Syllabus of Compiler Principles

Course Name/Title： Compiler Principles **Course code：62902**

Course Type： (Specialized Course)(Compulsory Course)

Total Teaching Hours： 48 (Classroom Hours: 39 Laboratory Hours or Tutorial Hours 9)

Course Credit： 3

I Course Introduction

Compiler Principles is an important professional course in computer science. In the computer software architecture, programming language is an important field, which includes three aspects: language application (programming), language translation (compiler), language design (study language syntax, semantics, and theory and implementation of a programming language). Compiler serves as a connecting link role in the architecture. This course introduces basic design ideas and main technical methods and some of the automatic construction tools. Students can systematically learn the compiler theory, such as: identification of finite automata theory, formal languages, semantic check, runtime storage management, code optimization and code generation. Students are expected to be familiar with the whole process of the compiler construction and can apply the techniques and ideas in general software design.

II Course Objective

Course Objective 1: Enhance students' sense of responsibility of "powerful country in science and technology" by introducing the evolution of compiler technology. Emphasizing the significant impact of basic software such as compiler and stimulate students' patriotic enthusiasm and cultivate their spirit of hard study and craftsmanship

Course Objective 2: Master the basic principles and methods of lexical analysis, syntax analysis, and semantic analysis. Abstract and identify a language and express it using formal methods such as regular expression and context free grammar. Express problems in an abstract way, and have the thinking method to solve a class of problems.

Course Objective 3: Understand the software structure of the compiler system and the design and implementation of module and interface. Be able to compare different representation models and implementation methods. For a given high-level language, have the ability to analyze the requirements and design the lexical analysis, syntax analysis, semantic analysis and other modules properly. Can choose appropriate implementation methods.

Course Objective 4: Cultivate the research consciousness in the process of solving compilation problems. Learn the research methods of processing ambiguity, conflict and uncertainty in classic compiler problems. Understand the thinking method of determining the research object, solve the relevant uncertainty by querying relevant information and scientific experiments, experience the fun of success through research, and further strengthen the research consciousness. Cultivate students' practical ability and innovative consciousness.

Course Objective 5: Select and apply appropriate technology and resources such as lex and YACC to solve the problems. Evaluate their limitations. Understand the problems of basic software such as compiler, stimulate patriotic enthusiasm, and cultivate students' spirit of assiduous research and craftsmanship.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
GR 1:Engineering Knowledge	1-3 Be able to use basic engineering knowledge, professional knowledge and mathematical models to derive engineering problems and analyze the influencing factors.	Course Objective 1,2
GR 2:Analyse	2-3 apply basic principles of mathematics, natural sciences and computer science, analyze the main factors affecting computer complex systems, demonstrate the rationality of solutions and obtain effective conclusions.	Course Objective 3
GR 4:Research	4-2 Based on scientific principles and using scientific methods, for the overall realization of complex computer systems, be able to formulate experimental plans, build experimental systems, and conduct experiments.	Course Objective 4
GR 5:Morden Tools	5-2 Be able to use integrated development tools, open source and third-party resources to develop, debug and test computer systems, and understand their limitations	Course Objective 5

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Unit 1 Introduction** (1)Language processor, compiler structure, language classification and its influence on compiler. (2) Modeling and code optimization in compiler design and implementation. (3)Optimization of computer architecture, software productivity tools.	(10) Grasp the requirements and challenge of mapping a language into another equivalent language; understand the function of tools, the history and trends of the field. (11) Can retell the structure of the compiler and its data structure, preliminary understanding and experience the fun of automatic calculation. (12) Through the description of the evolution route of compiler technology, enhance students' sense of responsibility of "science and technology power".	3	Lecture、recitation	1、2、3
2	**Unit 2 SCANNING** (1)Introduction of the mathematical model and the algorithm used to identify tokens. (2) Regular expression and normal set, state transition graph, deterministic finite automata (DFA), uncertain finite automata (NFA), transformation algorithm of finite automata from NFA to DFA, minimization algorithm of DFA. (3) The principle of lexical analysis design and automatic generation (Lex) of lexical analysis.	(1) Can specify the pattern of tokens with regular expressions and simulate the recognition process of tokens with finite automata. (2) Can eliminate the uncertainty and know the methods of transforming NFA into DFA. Can minimize the number of states in a DFA (3) Be able to use Lex to generate lexical analyzer automatically, and understand the advantages and limitations of tool. (4) Understanding the structural changes of morphology require the automatic reconstruction of lexical analyzer. Stimulate students' interest in solving problems independently, and cultivate students' professional qualities and eagerness to explore the unknown.	9	Lecture、recitation	1、2、3
3	**Unit 3：Context-Free Grammars and Parsing** (1) Formal specification of grammar and language, concepts of grammar, language and derivation. (2) Context free grammar (CFG), the solution of language, the judgment of	(1) Be able to use formal definition of grammar and language. (2) Understand basic concepts of grammar, language, normal derivation; Master the solving method of language, the judgment method of grammar ambiguity and the analysis method of sentence pattern. (3) The method of eliminating ambiguity,	6	Lecture、recitation	1、2、3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	grammar ambiguity and the analysis of sentence patterns.	eliminating left recursion and extracting left factoring to design a well-formed grammar.			
4	**Unit 4：Top-Down Parsing** (1)The Basic Method of LL(1) Parsing; (2)The LL(1) Parsing Table and Algorithm; (3) predictive sets: First Sets; Follow Sets; Constructing LL(1) Parsing Tables;	(1) Can use top-down analysis to solve parsing problems, can compare the two top-down analyzer implementation: predictive analysis and retrospective analysis. (2) Can use predictive sets FIRST and FOLLOW, which can help to construct predictive parsing tables. Master LL(1) algorithm.	6	Lecture、recitation	1、2、3
5	**Unit 5：Bottom-Up Parsing** (1)overview of bottom-up parsing; (2) generic Shift-Reduce framework (3)definition , category of conflict and related solutions (4)Simple LR method; (5) the SLR(1) parsing algorithm; The LR(1) and LALR(1) Parsing algorithm	(1) Objectives：Understand the principle of the bottom up analysis method. Can use generic Shift-Reduce framework to construct a syntax analyzer. Judge the LR (0) grammar and construct LR (0) analysis table and analysis method. Understand the SLR (1) the method of algorithm. Understand and grasp the LR (1) and LALR (1) algorithm. Use the generator tool YACC to build a Parser. (2) Can retell the principle of bottom-up analysis. (3) Can use generic Shift-Reduce framework to construct a syntax analyzer. Can judge the LR (0) grammar and construct LR (0) analysis table and analysis method. (4) Understand the principle of SLR (1) method and analysis table algorithm (5) Understand and master LR (1) and LALR (1) algorithm. Be able to use YACC to build a Parser. (6) Can compare different grammatical analysis methods, cultivate dialectical view of things, analyze problems from different angles and simplify complex problems.	9	Lecture、recitation	1、2、3
6	**Unit 6：Semantic Analysis**	(7) Objectives ： Learn to associate	6	Lecture、	1、2、3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	and Runtime Environments (1)Attributes and attribute grammars; dependency graphs and evaluation order; （2）Synthesized and inherited attributes; （3）The computation of attributes during parsing; The dependence of attributes computation on the syntax.	information with a language construct by attaching attributes to grammar symbols representing the construct. Use semantic rules cleverly. Use synthesized and inherited attributes definition to distinguish the computation of attributes during parsing; properly use dependency graphs to get evaluation order; Understand symbol table; Scope rules and block structure; Runtime Environments.		recitation	

V Period Distribution and Teaching Modes

2. Theory Lecture period distribution

Teaching mode / hours / contents	lecture	exercise	Discussion	remark	S total
Unit 1：Introduction of compiler	3				3
Unit 2：SCANNING	8	1			9
Unit 3：Context-Free Grammars and Parsing	6				6
Unit 4：Top-Down Parsing	6				6
Unit 5：Bottom-Up Parsing	8	1			9
Unit 6：Semantic Analysis and Runtime Environments	6				6
Total	**37**	**2**			**39**

3. Experiment period distribution

Unit	Content	Lab site	Hours	No.per Group	Experiment type	Compulsory

Experiment 1	Patten matching: **Theoretical knowledge:** Master the writing method of normal expression. **Experimental skills:** Be able to express the word in the normal form according to the definition of the specified language. Be able to use Lex tools to construct lexical analyzer, and be ready to use the basic environment. Be able to design experimental steps.	Software Lab. of Information College	3	1	comprehensive	Yes
Experiment 2	Construction of recursive descent parser **Theoretical knowledge:** master the requirements of grammar rules that can be implemented by recursive descent algorithm **Experimental knowledge:** can use recursive descent method to construct a parser.	Software Lab. of Information College	3	1	design	Yes
Experiment 3	Construction of parser based on YACC Learn to use yacc tool to define and build parser	Software Lab. of Information College	3	1	design	Yes

2. Teaching methods

1. We recognize that few students will build, or even maintain, a compiler for a major programming language. Yet the models, theory, and algorithms associated with a compiler can be applied to a wide range of problems in software design and software development. So the emphasis the idea, method and algorithm is very important.

2. Individualization of teaching.

In the individualized teaching strategy, the teacher makes clear the theme, publishes a certain learning task in this course, and lets the students find the relevant content on the

Internet, prepare the solution and implementation algorithm. Make students establish clear learning objectives, study textbooks and other methods, and lay a good foundation for the completion of learning tasks. At the same time, we should teach students in accordance with their aptitude, give them more free space, and guide them to solve projects according to their own interests. For the less advanced students, they also have the joy of completing simple tasks.

3. Research, explore guide strategy - extracurricular design.

The teacher provides the skeleton frame of the project, guides students to explore the familiar objects in depth after class. For example, a simple calculator can be implemented with C ++ classes, and how to do symbol deconstruction in the syntax processor, and how to backtrack the look-up mark, etc. Make students feel the charm of compiler design.

4. Situational interactive self-learning strategies.

Lex and YACC are important tools of compiler technology. It takes lots of skill to master. Students need to refer to a large number of documents besides text book. The teacher should partition difficult problems, create problem situation, and guide the students to do self-learning in groups. Cultivate the concept of student-centered in stead of traditional passive learning. Encouraging autonomous learning, active learning, independent exploration, and self-improvement.

VI Assessment

Assessment Methods or Approaches		**Assessment Requirements**	**Assessment Weighting**	**Evaluation of Course Objectives**
Usual per-formance	the rates of attendance	Evaluate according to class attendance, participation and practice.	40%	Course Objectives 1、2、3、4、5
	homework	Evaluate according to the hand in, accuracy, writing quality of homework.		
	experiment	Evaluate according to the experimental design, experimental operation and experimental report quality.		
Final Exam (close-book)		Assess the correctness, the steps and the ideas of solving the questions.	60%	Course Objectives 2、3

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

1. Textbook

Kenneth C. Louden：COMPILER CONSTRUCTION Principles and Practice ，China Machine Press，2002.

2. References

[1] Aho, Sethi, and Ullman. Compilers:Principles,Techniques and Tools, Second Edition, China Machine Press，2011.

[2] 蒋宗礼、姜守旭编著:《编译原理》，高等教育出版社，2010 年出版

[3] 王生原，董渊，张素琴，吕映芝，蒋维杜等编:《编译原理》（第 3 版），清华大学出版社， 2015 年出版。

Written by：Xia jinsong **Reviewed by: Lin wang**

Date：2021.05.20

Syllabus of Assembly Language

Course Name/Title： Assembly Language **Course code：** 62509

Course Type： (Specialized Basic Course) (Optional Course)

Total Teaching Hours： 48 (Classroom Hours:39 Laboratory Hours:9)

Course Credit： 3

I Course Introduction

Assembly Language is designed for undergraduate students majoring in Computer Science and Technology, and is a major elective course. The undergraduate students who choose Assembly Language are required to master programming language; in this way, they can deepen their understanding of assembly based on C language, which they are relatively familiar with. More important, assembly language is a kind of high-level language represented at the machine level. The implementation of the program and the work progress of computer can be closely linked by mastering the corresponding of these two languages, directly reflecting the inherent characteristics of the assembly language, which means it is the combination that can easily unify “program” and “machine”. Give the differences in microarchitecture through further explanation of the different assembly code, and it can provide pilot knowledge for some follow-up courses, such as Compiler Principles, Operating System, Computer Architecture, as well as an understanding of the role and position of various courses for students from the perspective of the whole system.

II Course Objective

Course Objective 1: Connect assembly language with other professional courses to form a complete understanding of computer system, let students connect different views and ideas, find and understand the dialectical relationship between computer professional courses, and recognize the development space and bottleneck of IT industry in China, and establish the determination and confidence to learn professional well and strive for the prosperity and strength of the country.

Course Objective 2: Understand the basic components and structures of the computer, and master the representation of fixed-point data, data conversion and the corresponding operation method, and the common set of X86 instruction set, and master the program structure and programming specification of assembly language.

Course Objective 3: Understand the advantages and application scenarios of assembly language compared with other high-level languages, and master the key contents of assembly language programming, such as data representation and operation, loop and branch, subprogram structure.

Course Objective 4: Master the development process, methods and techniques of assembly language programming, and understand the various factors that affect the function and performance objectives of assembly language programs.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1: Engineering Knowledge	1-1 Be able to apply mathematics, natural science, engineering fundamental and professional knowledge to the proper expression of computer complex engineering problems.	Course Objective 1,2
Graduation Requirement 2: Problem Analysis	2-1 Be able to identify computer complex engineering problems, and clearly express the needs and key processes of computer complex engineering problem through research.	Course Objective 1,3
Graduation Requirement 3: Design / Develop Solutions	3-1 Master the whole cycle of development process and basic design / development methods and technologies of computer system and software products, and understand various factors affecting design objectives and technical solutions.	Course Objective 1,4

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Ch1 Basic Concepts 1) assembly language advantages and applications. 2) data representation. 3) Boolean operations	1) list advantages and disadvantages of assembly language; 2) list the common applications of assembly language 3) describe detailed data representation of integers and characters in computers 4) explain Boolean operations such as and, or, not, and exclusive or, and Boolean expressions	3	Classroom teaching and discussion	Course Objectives 2
2	Ch2 x86 Processor Architecture 1) x86 architecture details 2) x86 memory management	1) list x86 processor’s three primary modes of operation 2) distinguish basic program execution	3	Classroom teaching and discussion	Course Objectives 1,2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	3) components of a typical x86 computer	environment of x86 processor 3) describe x86 processor memory management 4) explain how the x86 integrates with other components by examining a typical motherboard configuration 5) recognize the development space and bottleneck of IT industry in China, and establish the determination and confidence to learn professional well and strive for the prosperity and strength of the country			
3	Ch3 Assembly Language Fundamentals 1) basic elements of assembly language 2) assembling, linking, and running programs 3) defining data 4) symbolic constants	1)define integer, character, and string constants and expressions 2) describe reserved words, identifiers, directives, and instructions 3) describe the assemble – link - execute cycle of a program 4) define symbolic constants using EQUAL-SIGN, EQU, and TEXTEQU directives 5) connect assembly language with high level languages to form a complete understanding of computer system, let students find and understand the dialectical relationship between computer professional courses	6	Classroom teaching and discussion	Course Objectives 1,2
4	Ch4 Data Transfers, Addressing, and Arithmetic 1) data transfer instructions 2) addition and subtraction 3) data-related operators and directives 4) indirect addressing 5) JMP and LOOP instructions	1) list three types of instruction operand: immediate, register, and memory operands 2) describe the data transfer instructions: MOV, LAHF, XCHG instructions 3) distinguish direct addressing, indirect addressing 4) describe the syntax of addition and subtraction instructions: ADD, SUB, INC, DEC, NEG instructions 5) distinguish the data-related operators and directives: OFFSET, ALIGN, PTR, TYPE, LENGTHOF, SIZEOF,	6	Classroom teaching and discussion	Course Objectives 2,3,4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		LABEL 6) describe array and pointer operands 7) describe the syntax of JMP and LOOP instructions			
5	Ch5 Procedures 1) linking to a external library 2) stack operations 3) defining and using procedures	1) describe how to link to a external library 2) explain the runtime stack and stack operations 3) describe the syntax of push and pop instructions 4) define a procedure using proc directive and use a procedure by call and ret instructions	3	Classroom teaching and discussion	Course Objectives 2,3,4
6	Ch6 Conditional Processing 1) Boolean and comparison instructions 2) conditional jumps 3) conditional loop instructions 4) conditional structures	1) distinguish CPU flags: zero, sign, overflow, parity flags 2) describe the syntax of Boolean and comparison instructions: AND, OR, NOT, XOR, TEST, CMP instructions 3) explain conditional jump structures and applications 4) describe the syntax of conditional jump instructions 5) distinguish conditional loop instructions: LOOPZ, LOOPNZ 6) implement the block-structured IF, WHILE, Table-Driven selection statements 7) compare assembly language to high level languages about flow control, let students find and understand the dialectical relationship between different languages	6	Classroom teaching and discussion	Course Objectives 1,2,3,4
7	Ch7 Integer Arithmetic 1) shift and rotate instructions and applications 2) multiplication and division instructions 3) extended addition and subtraction 4) ascii and unpacked decimal arithmetic	1) distinguish the syntax and applications of shift and rotate instructions: SHL, SHR, SAL, SAR, ROL, ROR, RCL, RCR, SHLD, SHRD 2) describe the syntax and applications of multiplication and division instructions: MUL, DIV, IMUL,	6	Classroom teaching and discussion	Course Objectives 2,3,4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		IDIV 3) explain the extended addition and subtraction application with ADC and SBB instructions 4) describe ascii and unpacked decimal arithmetic operations with AAA, AAS, AAM, AAD instructions			
8	Ch8 Advanced Procedures 1) stack frames 2) recursion 3) INVOKE, ADDR, PROC and PROTO directives	1) explain stack frames and related operations 2) describe how to access stack parameters and local variables 3) describe enter and leave instructions to manage stack frames 4) describe how to declare local variables by LOCAL directive 5) explain recursion and recursion programming method 6) describe how to declare and use a procedure with proc, invoke and proto directives and ADDR operator 7) distinguish the formal and actual parameters of a procedure 8) compare assembly language to high level languages about parameters passing, let students find and understand the dialectical relationship between different languages	6	Classroom teaching and discussion	Course Objectives 1,3,4

V Period Distribution and Teaching Modes

a. Course hours allocation

(1) Class hour distribution table of theoretical teaching content

Teaching Content \ Teaching Hours \ Teaching Mode	Theoretical Teaching	Exercise Class	Discussion Class	Remarks	Subtotal
Ch1 Basic Concepts	3				3
Ch2 x86 Processor Architecture	2	1			3
Ch3 Assembly Language Fundamentals	6				6

Ch4 Data Transfers, Addressing, and Arithmetic	5	1			6
Ch5 Procedures	3				3
Ch6 Conditional Processing	6				6
Ch7 Integer Arithmetic	5	1			6
Ch8 Advanced Procedures	5		1		6
Total	**35**	**3**	**1**		**39**

(2) Class hour distribution table of experimental teaching content

Experiment Name	**Synopsis**	**Main Equipment or Experimental Environment**	**Experiment Hours**	**Persons per group**	**Experiment Attributes**	**Request (Required or Optional)**
Experiment 1: Programming with MS Visual Studio	Complete the basic programming in assembly language using MS Visual studio	Microcomputer with MS Visual Studio software	3	1	Basic	Required
Experiment 2: Branch and loop programming	Complete the Conditional jump and loop programming in assembly language	Microcomputer with MS Visual Studio software	3	1	Comprehensive	Required
Experiment 3: Procedure programming	Complete modular programming by declaring and calling procedures in assembly language	Microcomputer with MS Visual Studio software	3	1	Comprehensive	Required

b. Teaching method

The combination of lectures and live demonstrations will be taken as teaching methods, allowing students to have a visual concept. Combined with the practical background, students are arranged in writing assembly program in order to further train the students' practical ability, design ability and solving problem ability.

VI Assessment

Assessment Methods or Approaches		Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
regular	Classroom	Evaluate according to the attendance rate,	30%	Course

grade	performance	classroom performance, classroom exercises		Objective 1, 2, 3
	Homework	Evaluate according to the times of handing in, accuracy and quality of assignments		
Experiment grade		Evaluate according to classroom performance, program design, and experimental report quality	20%	Course Objective 2, 3, 4
Final exam grade (open book)		Evaluate according to the correctness of questions and problem-solving ideas	50%	Course Objective 2, 3, 4

VII Textbooks and References

（1）Textbooks：

Kip R Irvine, Assembly Language for X86 Processors（photoetching）, Tsinghua university press,

(2) References

a) Wang Shuang, *Assembly Language*, Tsinghua university press
b) Randall Hyde, The Art of Assembly Language, Tsinghua university press

Written by：Fu Feng **Reviewed by: Jia Yubo**

Date：2021.05.20

Syllabus of Fundamental of Information Security

Course Name/Title：Fundamental of Information Security **Course code**：62619

Course Type：(Basic Course)(Optional Course)

Total Teaching Hours：32 (Classroom Hours: 32 Laboratory Hours or Tutorial Hours 0)

Course Credit：2

I Course Introduction

With the rapid development of information technology, information security technology is increasingly becoming an important topic. The fundamental of information security is one of the recognized mainstream technologies in the computer science, and it is also a comprehensive discipline that is rapidly developing and widely used in the information technology community. The purpose of this course is to introduce a general knowledge of information security and the topics cover the concepts, theories and methods of information security. The specific course objectives are as follows:

II Course Objective

Enable the students to understand the important of information science, encourage students to devote themselves to scientific research, help to establish innovative consciousness and patriotic enthusiasm.

Course Objective1: enable the students to describe the security service mechanism of information security, understand the principles of symmetric cryptography, asymmetric cryptography and secure protocols, design secure algorithms and protocols for information systems in practice.

Course Objective 2: enable the students to describe the common attack methods and defense measures, assess the security of symmetric cryptography, asymmetric cryptography and secure protocols, analyze the security of the existing algorithms and protocols in practice.

Course Objective3: enable the students to assess the vulnerability of existing applications and identify the potential hazards to human beings and computer system.

Course Objective4: enable the students to develop the skills in reading research papers and ability of preliminary scientific research, and thereby form the habits of self-study and lifelong learning.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1	1.1 Be able to build mathematical model for the problems in computer Engineering	Course Objective 1

	field.	
Graduation Requirement 2	2.2 Be able to find different solutions to complex engineering problems in computer application, through literature review and describe the solutions correctly.	Course Objective 2
Graduation Requirement 6	6-2 Be able to be familiar with the related standards, regulations policy, and understand the responsibility they should take.	Course Objective 3
Graduation Requirement 12	12-2 Understand the importance of lifelong learning, form a sense of lifelong learning, and adapt to sustainable career development.	Course Objective 4

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 Introduction** 1.1 The concept of information security. 1.2 The history of information security 1.3 Common attack methods and defenses mechanisms	(1) Review the history and origin of information science; (2) Explain the challenge and opportunity China is facing. Enable the students to understand the important of information science, helps to establish innovative consciousness and patriotic enthusiasm.	2	Teaching in class or Panel discussion	Course Objective 1、2、3、4
2	**Chapter 2 Classical cryptography** 2.1 The concept of classical ciphers 2.2 analysis of classical ciphers.	(1) Explain the concept of classical ciphers (2) Introduce the statistical analysis method for classical ciphers.	2	Teaching in class or Panel discussion	Course Objective 1、 2、3
3	**Chapter3 Symmetric cryptography** 3.1 The concept of symmetric ciphers	(1) Introduce the definitions of symmetric ciphers (2) Explain construction of LFSR, DES in details.	4	Teaching in class or Panel discussion	Course Objective 1、2、3

Num.	**Course Content**	**Teaching Requirement**	**Period**	**Teaching modes**	**Course Objectives**
	3.2 LFSR and DES				
4	**Chapter 4 Public cryptography** 4.1 The concept of public key cryptography 4.2 DH and RSA	(1) Introduce the concept of public key cryptography (2) Explain DH and RSA (3) Enable the students to understand the uniqueness of public key cryptography, helps to establish innovative consciousness and patriotic enthusiasm.	8	Teaching in class or Panel discussion	Course Objective 1、2、3、4
5	**Chapter 5 Digital signature** 5.1 The concept of digital signature 5.2 RSA signature scheme	(1) Introduce the concept of digital signature (2) Explain RSA signature.	2	Teaching in class or Panel discussion	Course Objective 1、2、3
6	**Chapter 6 Hash function** 6.1 The concept of hash function 6.2 SHA-1	(1) Introduce the concept of hash function (2) Explain in details the construction of SHA-1.	2	Teaching in class or Panel discussion	Course Objective 1、2、3
7	**Chapter 7 Message authentication code(MAC)** 7.1 The concept of message authentication code 7.2 HMAC and CBC-MAC	(1) Introduce the concept of MAC (2) Explain in details the construction of HMAC and CBC-MAC.	2	Teaching in class or Panel discussion	Course Objective 1、2、3
8	**Chapter8 Public-key infrastructure (PKI)** 8.1 The concept of message authentication code 8.2 PKI	(1) Introduce the concept of PKI (2) Explain in details the mechanism of PKI.	2	Teaching in class or Panel discussion	Course Objective 1、2、3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
9	**Chapter9 Secure protocols** 9.1 The concept of message authentication code 9.2 SSL and IPSEC	(1) Introduce the concept of secure protocols (2) Explain in details the mechanism of SSL and IPSEC.	4	Teaching in class or Panel discussion	Course Objective 1、2、3
10	**Chapter 10 Advanced topics** 10.1 Cloud computing 10.2 Secure computation 10.3 Homomorphic encryption	(1) Introduce the cutting edge topics (2) Enable the students to establish innovative consciousness and increase interest in exploring truth.	4	Teaching in class or Panel discussion	Course Objective 1、2、3、4

V Period Distribution and Teaching Modes

(1) Period Distribution

Teaching methods / Teaching hours / Teaching content	Teaching in class	Exercise class	Panel discussion	Remark	Total
Chapter 1 Introduction	2				2
Chapter 2 Classical cryptography	2				2
Chapter 3 Symmetric cryptography	4				4
Chapter 4 Public key cryptography	8				8
Chapter 5 Digital signature	2				2
Chapter 6 Hash function	2				2
Chapter 7 Message authentication code	2				2
Chapter 8 PKI	2				2
Chapter 9 Secure protocols	4				4
Chapter 10 Advanced topics	4				4
Total	**32**				**32**

(2) Teaching methods

In addition to the multimedia tool, the traditional teaching method, e.g., blackboard writing, should be adopted to help the students in understanding the content. The exercises in class and class discussions are strongly encouraged. Also, homework is important to help the students to consolidate the knowledge. The teacher should encourage students to read related academic papers and explore the cutting-edge technology in the field of information security

VI Assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Performance in class	Attendance. Totally 10 points. -0.5 point for being one late once， and -1 point for being absence once. Performance in class. Totally 10 points. 1 point will be deducted if the question is not answered. Denote by A2 the point.	30%	Course Objective 1、2、3、4
Homework	According to the number of submission. 50 points bonus for the students who submit every time. -10 points for the students with one missed submission. According to the correctness of homework. 40+ points for the excellent, 40 points for the good, 35 points for the average, 30 points for the fair, and <30 points for the fail. Denote by B2 to point. ③ and 10 points bonus for students with a total of more than 5 A+s; -3 points deduction for students with one late submission. (2).-5 points deduction for students with 1 absence and students with 3 absences will fail.		
Project	According to the quality of the project.	20%	Course objective 3、4
Final exam (close book)	According to correctness of the answers.	50%	Course objective 1、2

VII Textbooks and References

(1) Textbook

[1] Christof Paar, Jan Pelzl. Understanding Cryptography: A Textbook for Students and Practitioners. Springer, 2012.

(2) References

[1] William Stallings, Cryptography and Network Security (5th Edition). Publishing House of Electronics Industry, 2012.

[2] Christof Paar, Jan Pelzl. Understanding Cryptography: A Textbook for Students and Practitioners. Springer, 2010.

[3] Jonathan Katz and Yehuda Lindell. Introduction to Modern Cryptography. Chapman and Hall/CRC, 2007.

Written by：Hai Huang **Reviewed by: Zhiyu Zhou**

Date：2021.05.20

Syllabus of Analysis and Design of Algorithms

Course Name/Title： Analysis and design of Algorithms **Course code：** 62916

Course Type：(General Course，**Basic Course**，Specialized Course)(**Optional Course**)

Total Teaching Hours： 48(Classroom Hours: 33 Laboratory Hours or Tutorial Hours: 15)
Course Credit： 3

I Course Introduction

Algorithm design and analysis is one of core problems of computer science, and also an important professional basic course for graduates of computer science & technology and related professional. This course is designed to systematacially introduce computer algorithm design methods and analysis technologies. Students study common algorithm design strategies in this course. Topics include sorting and searching, advanced data structures, graph algorithm, numerical algorithm, distributed algorithms, computational geometry string matching and NP-completeness. Students will comprehend and become acquainted with basic skills of algorithm design and analysis by researching on common and representative algorithms.

II Course Objective

Course Objective1: Special emphasis will be placed on problem solving, unifying ideas, proof techniques, the "scientific method", as well as striving for elegance, insights, and generalizability in developing algorithms and proofs.

Course Objective2: This course aims to cultivate students' initial capacity to analyze the complexity of algorithms, exercise their logical thinking ability and enable them to understand of the development of theory.

Course Objective3: Students will be provided with independent research capability and ability to integrate theory with practice. . Student will also learn to solve practical problems with algorithms.

III **Correlations between Course Objectives and Graduation Requirements**

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1	1.2 Students have acquired a reasonable professional knowledge structure. They can apply the knowledge and methods of discrete mathematics, algorithm, data structure and programming to Computational Thinking, to the analysis and implementation of basic algorithm problems and the analysis and understanding of the working principle or mechanism of complex software systems.	Course Objective 1,2
Graduation Requirement 2	3.3 Be able to realize the computer system that meets the requirements according to the design scheme, fully considering the cost performance, and following the software engineering specification.	Course Objective 2,3
Graduation Requirement 7	7.2 Be able to assess the efficiency of resource utilization, the scheme of waste disposal and the safety precautions and identify the potential hazards to human beings and the environment in the lifecycle of computer-related products for complex computer application problems.	Course Objective 2,3

(Note: Basic courses and specialized courses must correlate with the graduation requirements as specified in the Program outline. The correlated graduation requirement index point must be put before the descriptive phrases or sentences. General courses are exempted from this rule.)

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	2.1 Introduction	Review of asymptotic Functions Review of basic data structures Review of basic algorithms	3h	Classroom,Online	Course Objective 1,2
2	2.2 Sorting and searching	• Review of classical sorting	4h	Classroom,Online	Course Objective

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		• Interpolation Search • Specialized sorting methods • Deterministic K^{th} selection • Lower bounds on max & min • Majority detection • Meta algorithms			2,3
3	2.3 Advanced data structures	Skip lists • Amortized analysis • Fibonacci heaps • Perfect hashing, cuckoo hashing	3h	Classroom,Online	Course Objective 2,3
4	2.4 Graph algorithms	Lowest common ancestor • Minimum spanning trees • Shortest paths trees • Radius-cost tradeoffs • Steiner trees • Minimum matchings • Network flows • Degree-constrained trees	4h	Classroom,Online	Course Objective 1,2,3
6	2.6Distributed algorithms	•Asynchronous consensus impossibility • Leader election in a ring • Leader election in graphs • Distributed MSTs	4h	Classroom,Online	Course Objective 2,3
7	2.7 Computational geometry	• Lower bounds • Chan's convex hull algorithm • Segment intersection • Planar subdivision search • Voronoi diagrams • Nearest neighbors	6h	Classroom,Online	Course Objective 2,3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		• Geometric minimum spanning trees • Delaunay triangulations • Minimum density trees • Minimum bounding box • distance between convex polygons • Smallest Enclosing Circle • Triangulation of polygons • Collinear subsets • Probabilistic analysis			
8	2.8 String matching	• Knuth-Morris-Pratt • Boyer-Moore • Edit distance • Longest increasing subsequence • Smith-Waterman algorithm	3h	Classroom,Online	Course Objective 2,3
9	2.9 NP-completeness	Polynomial time and intractability • Space and time complexity • Problem reductions • NP-completeness of satisfiability • Independent sets • Graph colorability • Travelling salesperson problem • Approximation heuristics	3h	Classroom,Online	Course Objective 2,3

V Period Distribution and Teaching Modes

3. Teaching Modes

6) Teaching modes include Classroom lectures, online courses, classroom discussions, experiments, etc.
7) During the teaching processes of the course, we are going to take multiple platforms, the ACM OJ(Online Judgement)，the CHIDE environment, the LINUX environment, the COMMAND SHELL platform.
8) The course handouts are distributed at lecture time. Some of them are available on the

Web of the home page which will be informed when it is ready. The homework is given out weekly in class through handout or through email.

9) Special stress should be put on the introduction of modern educational technology with an optimal integration of various teaching media.

10) Make reasonable use of presentation instruments, inquiring instruments, interaction instruments and design instruments to effectively improve the teaching quality.

VI Assessment

Lab Hours	**15** hours
Lab Venue	In the software lab of building 10[#]
Lab Approach	Students are required to work out their projects outside of class, and lab hours are mainly used for discussions on common problems met by students in finishing the projects.
Additional Tutoring	N/A
Practices / Labs	Students are required to work out 5 programming projects. **(1) Project #1** **To realize sorting and searching, above all Interpolation Search;** **(2) Project #2** **To study and analysis Fibonacci heaps;** **(3) Project #3** **To work out Minimum spanning trees** **(4) Project #4** **To complete the Linear programming** **(5) Project #5** **To achieve Geometric minimum spanning trees**
Platforms	Operating System: Microsoft Windows Programming Language and Compiler: C/VC
Skill Development	None

VII、Autonomous Learning

Lectures and exercises with assignments. The assignments are a very important part of the course. Each week an assignment has to be solved. After the exercise class you have one week to solve the assignment and hand it in at the beginning of the next exercise class (or send it by mail/fax). Late hand-ins is not accepted. The assignments will be checked. A number of assignments involve programming in C. Although the programming language is not a crucial part of this course it is strongly recommended that you implement and run all programming exercises. The recommended way to do this is to use the server of the university. The details on The outlook to teaching and studying will be explained as follows:

- ✧ **Learning by doing ;**
- ✧ **International view ;**
- ✧ **Innovating and Exploring;**
- ✧ **knowledge integration ;**
- ✧ **Human-centric values ;**

VIII Suggestions for Teaching

You are responsible for material presented in lectures, including oral comments made by the lecturer.

You are also responsible for material presented in recitations. Attendance in recitation has been well correlated in the past with exam performance. Recitations also give you a more personalized opportunity to ask questions and interact with the course staff. Your recitation instructor, together with the lecturers, will assign your final grade.

IX **Suggestions for Course Learning**

(Mainly include tips for learning methods, course preparation information and requirements, and requirements for assignments, autonomous learning, discussion, etc.)
After having completed the course the student will be able to:

- Explain and derive the complexity of algorithms for basic and collective communication operations
- Apply different methods and performance measures to analyze algorithms with respect to cost and scalability, including using the so-called iso-efficiency function.

- Describe the basic methods of problem and data partitioning for efficient memory utilization and minimization of communication costs in parallel computers, such as recursive blocked and tiled algorithms for dense matrices and hybrid data layouts.
- Perform design and analysis of parallel algorithms in some of the fields mentioned under contents
- Use and be familiar with standard software in numerical linear algebra such as BLAS, LAPACK, and ScaLAPACK

X Assessment

Students will complete written homework assignments, programming projects, a midterm and a final. Each will be assigned a point value. Homework and program averages will be computed as total points earned divided by total points possible. These grading components will be weighted as follows. Letter grades for the course will be subject to my evaluation of your overall class performance. Please keep your own record of your grades so that you will know your standing in the course and can double-check my records. All grades will be available on web of our School. The specific contents are showed as follows:

Grading	Class Participation	10%
	Weekly Written Assignments	20%
	Midterm Examination	0%
	Final Examination	40%
	Programming Assignments	30%
	Total	100%

XI Textbooks and References

4. Textbook

Thomas .H. Cormen. Ronald. L. Rivest. Charles. E. Leiserson. Clifford. Stein., Introduction to Algorithms(Third Edition), The MIT Press, 2009.

In previous semesters the course has used the first or second edition of this text. We will be using material and exercise numbering from the third edition, making earlier editions unsuitable as substitutes.

5. References

The class will be relying mostly on the textbook, but additional material will occasionally be drawn from the following:

1. Thomas H Cormen, Charles E. Leiserson, Ronald L. Rivest and Cliff Stein, Introduction to Algorithms (second Edition) , The MIT Press, 2001
2. **李文书**，胡洁，骆淑云等编著[M]. 数据结构与算法应用实践教程[M]（第二版）. 北京：北京大学出版社. (2016.12) (**ISBN**: 978-7-301-27833-8) (**21 世纪全国应用型本科计算机案例型规划教材**)
3. **李文书**，赵悦等著. 数字图像处理算法及应用[M]. 北京：北京大学出版社. (2012.12) (**ISBN**: 978-7-301-21607-1) (**21 世纪全国本科院校电气信息类创新型应用人才培养规划教材**)
4. **李文书**，何利力，叶海荣等著. 算法设计、分析与应用教程[M]. 北京：北京大学出版社. (2014.07) (**ISBN**: 978-7-301-24352-7)

Written by：Ye Hairong **Reviewed by:Xianchuang Su**

Date：2021.05.20

Syllabus of C# Programming

Course Name/Title：C# Programming **Course code**：62980

Course Type：Specialized Course, Optional Course

Total Teaching Hours： 48 (Classroom Hours: 32 Laboratory Hours: 16)

Course Credit：3.0

I、 Course Introduction

C# is one of the world's fastest-growing object-oriented programming languages. "C# Programming" is an important and fundamental course for computer science majors. The course will introduce students to the unique syntax of C#, the fundamentals of object oriented programming and how to solve complex problems with object oriented programs. As a comprehensive and powerful API library, .Net Framework will also be introduced and practiced to build Windows form, IO, and database-based applications. Emphasis will be placed on understanding not only the syntactical features of the language, but how to effectively use the design of the language to develop robust software.

II、 Course Objective

In this course, students will learn the syntax of C# language and modern object-oriented programming methodology. They will be using the .NET framework and Visual Studio on which they can conveniently write, test and debug applications, including console applications, Windows form applications and database-based applications.

Upon the completion of this course students will be able to:

Course Objective 1: Master the C# programming fundamentals including data types, control statements, methods, strings and arrays. Master the newly introduced features, including delegate, Lambda expression, generics, etc.

Course Objective 2: Firmly establish object-oriented programming concepts and apply them in C# programs, including inheritance, encapsulation, polymorphism and generics. Be able to create complicated object models. Understand the usage and advantage of interfaces in class hierarchy. Know how to apply design patterns in C# programs.

Course Objective 3: Master the common .Net class library, including generic collections, files and streams, Windows Forms, ADO.NET, etc. Master the fundamentals of functional programming with LINQ.

Course Objective 4: Be able to utilize IDEs such as Visual Studio to create console, desktop and database manipulation applications, and to refactor and debug these applications.

Course Objective 5: Through this course, develop the habit of diligent and independent thinking, lifelong learning. Be able to develop the applications of practical value, and lay a solid foundation for the application of C# and .Net in the future.

III、 Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Indicator point of Graduation Requirements	Course Objectives	Description of Correlations
Graduation Requirement 1	1.2 Students have acquired a reasonable professional knowledge structure. They can apply the knowledge and methods of discrete mathematics, algorithm, data structure and programming to Computational Thinking, to the analysis and implementation of basic algorithm problems and the analysis and understanding of the working principle or mechanism of complex software systems.	Course Objective 1,2	The Course is to grasp the C# language fundamentals, to utilize the .Net Framework to design, write, debug and run C# programs. The Course is also a demonstration of object-oriented programming methodology. The analysis and implementation of some algorithm and some data structure will be covered and emphasized, e.g. the linked list, dictionary.
Graduation Requirement 2	2-1 Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.	Course Objective 2,3	C# language and .Net Framework provides flexible means to implement several solutions for a complex engineering problem, which may have different effects. Through plenty of programming practices, the students can learn to distinguish the difference between the solutions and choose the optimal one.
Graduation Requirement 3	3-2 Design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.	Course Objective 3,4,5	The solutions may differ in time, space and security. The course encourage students to try developing various solutions and compare them. The better choices will be lectured and showed later for more profound understanding.
Graduation Requirement 5	5.2 Be able to develop, debug and test computer systems, and understand their limitations by using integrated development tools, open source and third-party resources.	Course Objective 4	Besides Visual Studio and .Net Framework, the students are free and welcome to try any other IDEs, other .Net implementations, and third party libraries to implement the programming tasks. Some open-source libraries in Github and

			Gitee will be presented in the course to show the capabilities of the .Net platform.

IV、 Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	1 Introduction to Computers, the Internet and Visual C# 2 Introduction to Visual Studio and Visual Programming 3 Introduction to C# App Programming 4 Introduction to Classes, Objects, Methods and strings 5 Algorithm Development and Control Statements: Part 1 6 Control Statements: Part 2	To know the state of popular programming languages. To understand Microsoft's .NET initiative. To understand the structure of simple C# program. To know the basics of Visual Studio IDE. To use C# value types. To understand the distinction between value types and reference types. To use the operators to form complex statements. To use selection statements, repetition statements to form complex code blocks. By comparing the object-oriented programming methodology with the national and social management system, the students are guided to think about the differences of development modes. The confidence in home culture and system are trained.	4	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 1
2	7 Methods: A Deeper Look 8 Arrays; Introduction to Exception Handling 10 Classes and Objects: A Deeper Look 11 Object-Oriented Programming: Inheritance 12 OOP: Polymorphism and Interfaces 13 Exception Handling: A Deeper Look	To understand the basic concept of object-oriented programming methodology and interface-based programming in C#. To understand data structures and encapsulation. To declare a class, implement it and use it. To define parameters using the out, ref and params keywords. To derive one class from another. To define and implement interfaces. To use overridden methods to effect polymorphism. To manage the object life cycle. To manipulate strings. To	9	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		catch, throw, and rethrow exceptions. Cultivate team spirit and the sense of service.			
3	9 Introduction to LINQ and the List Collection 20 Generics 21 Generic Collections; Functional Programming with LINQ/PLINQ	To understand delegates and predefined delegate types. To use delegates and lambda expressions to pass methods to other methods for execution at a later time. To use anonymous methods including lambda expressions. To publish and subscribe to events. To understand collection interfaces and types. To use generic or non-generic lists, dictionaries, and sets. To create generic classes and methods. To query an array or a list using LINQ technique and iterate over the query results.	7	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 3
4	14 Graphical User Interfaces with Windows Forms: Part 1 15 Graphical User Interfaces with Windows Forms: Part 2	To create graphical user interfaces. To process events in response to user interactions with GUI controls. To create and manipulate various controls. Through the wide application of the C# language and. NET technology in a number of fields, students are encouraged to study hard to cultivate the spirit of craftsmen.	4		course objective 3,4, 5
5	16 Strings and Characters: A Deeper Look 17 Files and Streams	To use various methods of classes string. To obtain information about files and directories. To use streams, readers and writers to perform I/O operations. To perform object serialization. Cultivate team spirit and the sense of service.	4	Classroom lectures, online courses, classroom discussions, experiments, etc	course objective 3,4,5
6	22 Databases and LINQ	To understand the relational database	4	Classroom	course

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		model of ADO.NET. To use database access objects. To use an ADO.NET Entity Data Model to create classes for interacting with a database via LINQ to Entities. To use LINQ to retrieve and manipulate data from a database.		lectures, online courses, classroom discussions, experiments, etc	objective 3,4,5

Correlations between course experimentation contents and course objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Experimentation 1. C# programming fundamentals	To understand the distinction between value types and reference types. To transfer among primitive data types. To use the strings. To understand the features related to parameters. To use the arrays. To define, instantiate and use classes.	3	Programming experiments in lab	course objective 1, 4
2	Experimentation 2. Object-oriented programming	To master the classes and objects more skillfully. To build families of related classes using inheritance. To establish a polymorphic interface into class hierarchies. To implement methods called by delegates with lambda expressions.	3	Programming experiments in lab	course objective 2,4
3	Experimentation 3. Inheritance, polymorphism and delegates	To create a derived class that inherits attributes and behaviors from a base class, using virtual and abstract methods, overriding methods and properties. To use constructors in inheritance hierarchies. Practicing the definition, instantiation and invocation of delegates. Making the objects of a class comparable with interface IComparable and IComparer.	3	Programming experiments in lab	course objective 2,4,5
4	Experimentation 4.	To understand the framework of Windows	3	Programmi	course

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	Windows Form App Design	Form applications. To master the basic properties and events of forms and controls. To design drawing applications.		ng experiments in lab	objective 3,4,5
5	Experimentation 5. Files and Generics	To understand the concepts of file processing and I/O operations. To know how to read and write data in files. To persist objects with serialization. To use generic classes in data structure and algorithms. To master the basic usage of common generic collections. To apply multiple constraints on a type parameter.	3	Programming experiments in lab	course objective 3,4,5
6	Experimentation 6. ADO.NET	To know how to communicate with a relational database using the connected layer of ADO.NET, which is represented by connection objects, command objects, transaction objects, and data reader objects. To understand the in-memory datasets. To understand the concepts of LINQ. To know the Entity Framework.	3	Programming experiments in lab	course objective 3,4,5

V、 Period Distribution and Teaching Modes

4. Period Distribution

	Lecture	Experiment	Discussion	Attribute	Total
Introduction to C# and .NET framework	4	2			6
Object-oriented Programming	9	3			12
Delegate, Collections and Generics	7	3			10
Windows forms	4	2			6
Files and streams	4	3			7
ADO.NET	4	3			7
Total	32	16			48

5. Teaching Modes

1) Teaching modes include classroom lectures, online courses, classroom discussions,

experiments, comprehensive demonstration, etc.

2) During the teaching processes of the course, we are going to take multiple platforms, the SuperStar platform， the Visual Studio environment, the .Net Framework.

3) Case teaching is a important method that can help students acquire knowledge by means of analyzing and discussing C# program case.

4) The course handouts are distributed at lecture time. Some of them are available on the Web of the home page which will be informed when it is ready.

5) Special stress should be put on the introduction of modern educational technology with an optimal integration of various teaching media.

VI、 Assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
conventional assignments	8 times	30%	Course Objective 1,2,3,5
lab experiments	6 times	20%	Course Objective 1,2,3,4
final exam	Closed Exam	50%	Course Objective 1,2,3

Course grading

The final examination will be in the form of written test. The types of questions are choice problems and 4 writing programs problems.

Final grade (100 points)=Homework*30% + Lab experiment *20%+Final examination*50%

Grading criteria of different assessments

Content of Assessment	Methods of Assessment	Grading criteria
Final examination	Open-book examination	Standard answers
Class attendance and homework	Rollcall, and Statistics on homework	(1) According to correctness of homework. In addition, 10 points bonus for students with a total of more than 5 A+s; -3 points deduction for students with one late submission. (2) -5 points deduction for students with 1 absence and students with 3 absences will fail.
Lab experiment	Programs and documents	(1) According to correctness of programs. (2) According to the results of the programs.

Final examination assessments

Questions type	What are covered？	Criteria	Percentage	Course Objectives

Choice problems	Every parts of C# programming are included. It is important and necessary to master the fundamentals of C# language and .Net Framework	Results match	30%	Course Objectives 1、2、3
Writing programs	Finishing writing programs according to a design scheme.	Standard answers	70%	Course Objectives 1、2、3

VII、 Degree of Completion of Course Objectives

考核方式 / 权重 W_{ijk} / 课程目标	**Ordinary Achievements** R_1 **=50%**			**Final examination** R_3 **=50%**
	Classroom S_{11} **=0.20**	**Homework** S_{12} **=0.40**	**Experiment** S_{13} **=0.40**	Open-book S_{21} **=1**
Course objective 1	0.20	0.30	0.10	0.35
Course objective 2	0.00	0.30	0.10	0.40
Course objective 3	0.20	0.30	0.30	0.20
Course objective 4	0.20	0.00	0.50	0.00
Course objective 5	0.40	0.10	0.00	0.05

VIII、 Textbooks and References

6. Textbook

Paul Deitel, Harvey Deitel. Visual C# HOW TO PROGRAM. Pearson, 2017. ISBN: 978-013-460154-0.

7. References

[1] Mark J. Price. C# 7.1 and .NET Core 2.0 – Modern Cross-Platform Development. Packt Publishing Ltd., third edition, 2017.

[2] Mikael Olsson, C# 8 Quick Syntax Reference, Apress Media, LLC, third edition, 2020.

8. Resources

[1] https://mooc1.chaoxing.com/course/207934771.html

[2] https://docs.microsoft.com/en-us/dotnet/csharp/

[3] https://github.com/pdeitel/VCSharpHTP6

Written by：Zheng Junhong **Reviewed by: Fu Feng**

Date：20210520

Syllabus of Java Programming

Course Name/Title：Java Programming

Course Code： 62987

Course Type： (Specialized Basic Course)(Optional Course)

Total Teaching Hours： 48 (Classroom Hours: 32, Laboratory Hours or Tutorial Hours: 16)

Course Credit： 3

I Course Introduction

Java programming is an optional course for undergraduates majoring in computer science and technology. Its main content is to learn the basic statements, grammar and Java object-oriented programming methods and design process. Theoretically, except requiring students to master the basic statements and grammar of Java language, they should also understand the similarities and differences between Java and C/C++, including Java's object-oriented technology, input and output streams, exception handling, multithreading, graphical user interface, database development and network programming. Technically, it is required to be able to use JDK1.8 to edit, compile and debug Java programs more proficiently, and be able to develop applications with certain practical value.

II Course Objective

Course objective 1: Adhere to the combination of knowledge transfer and value guidance, use themes and content that cultivate college students' political beliefs, values, and spiritual pursuits, conduct professional identity and mission education, and cultivate students' professional pride and social responsibility.

Course objective 2: Master the basic knowledge of the Java programming language and the basic coding standards of Java programming. Adhere to the combination of knowledge transfer and value guidance, use themes and content that cultivate college students' political beliefs, values, and spiritual pursuits, conduct professional identity and mission education, and cultivate students' professional pride and social responsibility.

Course objective 3: Master the Java design method and have the basic skills to develop applications using the Java language.

Course objective 4: Understand the idea of object-oriented programming, train students to use "object-oriented thinking" in computer programming, inspire students' sense of innovation, and improve students' ability to analyze and solve problems in the process of programming. Build a good foundation for the further courses study and employment.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation requirement 2: Question Analysis	2-1 Identify problems for complex computer engineering, and clearly express the needs and key processes of complex computer engineering problems through investigations.	Course Objectives 1, 2
Graduation requirement 3: Design/Develop Solutions	3-1 Master the full-cycle development process, basic design/development methods and technologies of computer systems and software products, and understand various factors that affect design goals and technical solutions.	Course Objectives 1, 3
Graduation requirement 4: Research	4-2 Formulate experimental plans, build experimental systems, and conduct experiments based on scientific principles and scientific methods for the overall realization of complex computer systems.	Course Objectives 1, 4

IV Correlations between Course Content and Course Objectives

IV.1 Correlations between Theoretical Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 Overview of Java Language** (1) Develop history of Java language (2) Characteristics of Java language (3) Java virtual machine and Java running system (4) Install and use of JDK development environment	(1) Introduce the development process of the Java language and the reasons why it has been quickly promoted; (2) List the basic characteristics of the Java language: object-oriented, platform-independent, support for multi-threading, security, etc.; (3) Understand the Java virtual machine and Java operating system; (4) Master the installation and use of JDK development environment, as well as the installation and use of JCreator and	2	Lectures and discussion	Course Objectives 1, 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(5) Classification of Java program	Eclipse development tools; (5) Show examples to master the editing, compiling and running processes of Java application and Java applet. (6) During the epidemic, a computer simulation program implemented in Java visually appeals to everyone why they should stay at home instead of going out. With the theme of software defining the future world and the role of programmers in the epidemic, the education of professional identity and mission responsibility is carried out to cultivate students' professional pride and social responsibility.			
2	**Chapter 2 Java Language Basic** (1) Basic grammar of Java language (2) Define and reference of package (3) Common packages of Java	(1) Explain the basic grammar of Java language. (2) Distinguish the similarities and differences between Java syntax and C/C++. (3) Specify the definition and reference of the package, and list the commonly used packages of the Java system. (4) Comprehend the programming specifications of Java programs. (5) Aiming at the recruitment conditions for Java software engineers, Ali and Huawei coding standards, with the theme of software company coding standards and Java engineers' testimonials, conduct professional standard education and cultivate students to develop standardized coding habits.	2	Lectures, discussion, and experiment	Course Objective 1, 2, 3
3	**Chapter 3 Java Object-oriented programming** (1) Relation between class and object, definition and usage (2) Distinguish between	(1) Review related concepts such as classes and objects, as well as the basic composition of classes (2) Details the creation and use of objects; (3) Explain the Java garbage collection mechanism (4) Define methods and distinguish different	2	Lectures and discussion	Course Objective 2, 3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	object and object reference (3) Definition of method and parameter passing	ways of parameter passing (5) Write and debug Java programs based on object-oriented design (6) Distinguish the similarities and differences between Java object-oriented and C/C++; (7) The codes written by Ali programmers are collected by the National Museum, and they are responsible for the education of professional identity and mission, and cultivate students' sense of pride and social responsibility.			
4	**Chapter 4 Inheritance and Polymorphism** (1) Inheritance mechanism of Java (2) Initialization process of sub-class object (3) Abstract class, and abstract method (4) Interface	(1) Explain the inheritance mechanism of Java language; (2) Show the initialization process of subclass objects; (3) Distinguish final keywords to modify variables, methods and classes; (4) Explain abstract concepts, define and use abstract methods and abstract classes; (5) Introduce the essence of the interface, define and use the interface; (6) Master inner classes and anonymous inner classes.	4	Lectures and discussion	Course Objective 2, 3
5	**Chapter 5 Common Data Structures** (1) Definition and use of one-dimensional array, two-dimensional array and multi-dimensional array (2) Class String, StringBuffer and StringBuilder, and the difference between them (3) The inheritance relationship of the class set interface, typical set data	(1) Master the definition and use of one-dimensional arrays, two-dimensional arrays and multi-dimensional arrays; (2) Master the classes String, StringBuffer and StringBuilder, and the differences between them; (3) Master the definition and use of typical collection classes such as Vector and ArrayList; (4) Show the use of Java API documentation; (5) Explain the essence of generics and customize generics.	3	Lectures and discussion	Course Objective 2, 3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	structures such as Vector and Linklist (4) The definition of generics and custom generics				
6	**Chapter 6 Java Exception Handling** (1) Principle of Java exception mechanism (2) Classification of exception, and common exception classes in Java (3) Definition, throwing and capture of exception	(1) Define the concept of Java exception handling and distinguish the classification of exceptions (2) Familiar with common system exception classes of Java (3) Detailed definition, throwing and capture of Java exceptions (4) Master the Java custom exception programming process	2	Lectures, discussion and experiment	Course Objective 2 ,3
7	**Chapter 7 Java I/O Stream** (1) Related concepts of software quality assurance (2) The basic principle of IO operation, the inheritance relationship of the classes in the java.io package (3) Use the File class for file operations and the use of file operation streams (4) Master the three standard streams supported by the System class for IO (5) Use of buffer stream, data stream, and print stream (6) The role of object serialization and the	(1) Introduce the basic principles of IO operation and divide the classification of flows (2) Analyze the inheritance relationship and hierarchy of the classes in the java.io package (3) Use the File class to perform file operations, and implement file read and write operations through the file operation flow (4) Introduce the three standard streams supported by the System class for IO (5) Master the use of buffer stream, data stream, and print stream (6) Understand the role of object serialization and the use of object streams (7) According to the functional requirements, select the stream	4	Lectures, discussion and experiment	Course Objective 2, 3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	use of object streams (7) Choice of stream				
8	**Chapter 8 Java Applet** (1) The difference between applet and application (2) The life cycle of applet (3) Commonly used HTML tags and attributes related to applet.	(1) Understand the difference between applets and applications; (2) Familiar with the life cycle of applets; (3) Master the commonly used HTML tags and attributes related to applets.	1	Lectures and discussion	Course Objective 2, 3
9	**Chapter 9 Graphical User Interface Programming** (1) AWT and Swing, (2) Common components in Swing (3) Layout manager (4) The principle of the event handling mechanism and the realization of the event model.	(1) Distinguish between AWT and Swing; (2) List commonly used components in Swing; (3) Learn to set up a layout device; (4) Understand the principle of the Java event handling mechanism and how to use the listener.	1	Lectures, discussion and experiment	Course Objective 2, 3
10	**Chapter 10 Java Multithreading** (1) Threads and related basic concepts (2) Two implementation methods and differences of Java multithreading (3) The life cycle of a thread (4) The basic scheduling method of multithreading (5) The concept of synchronization and deadlock	(1) Understand threads and related basic concepts; (2) Master the two implementation methods and differences of Java multithreading; (3) Understand the life cycle of threads; (4) List the basic scheduling methods of multithreading; (5) Under the understanding of the concept of synchronization and deadlock, learn how to synchronize and break deadlock.	4	Lectures, discussion and experiment	Course Objective 2, 3, 4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
11	**Chapter 11 Java Networking** (1) Basic knowledge of network programming, (2) The relationship between IP address and InetAddress class (3) Use URL to locate network resources (4) Communication programming between client and server based on TCP and UDP protocol.	(1) Understand the concepts and methods of implementing network communication in Java; (2) Master the relationship between IP address and InetAddress class; (3) Use URL to locate network resources; (4) Master the communication programming method between client and server based on TCP and UDP protocol.	3	Lectures, discussion and experiment	Course Objective 2, 3, 4
12	**Chapter 12 Java Database** (1) The concept and driving method of JDBC (2) Basic operation of MySQL (3) Database operation classes and interfaces in the java.sql package (4) Simple database development of JDBC for MySQL	(1) Understand the concept of JDBC and distinguish various driver types; (2) Master the basic principles and implementation methods of using JDBC to access databases. (3) Familiar with the basic operation of MySQL; (4) Use the interfaces and classes provided in the java.sql package to design data applications.	4	Lectures, discussion and experiment	Course Objective 2, 3, 4

IV.2 Correlations between Experimental Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Experiment 1 Familiar with programming environment, editing, compiling and running Java programs (1) Setting of Java programming environment	(1) Familiar with the settings of the Java programming environment (2) Editing, compiling and running process of Java application and Java applet; (3) Design and write Java object-oriented programs; (4) Master the use of Java Swing	2	Lectures, and experiment	Course Objective 2, 3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(2) Classification of Java programs (3) The composition of Java programs and Java programming specifications (4) Java Swing component and layout manager	components, design and write graphical interface programs.			
2	Experiment 2 Simple Exception Handling (1) The concept and classification of Java exception handling (2) Customize exception handling process	(1) Define exceptions by inheriting Excption or its subclasses; (2) Program to throw and catch an instance of an exception class, run the program and observe the execution result	2	Lectures, and experiment	Course Objective 2, 3
3	Experiment 3 Java IO (1) Basic principles of IO operation (2) Basic operations of the File class (3) Selection of stream class	(1) Master the usage of byte character input and output streams; (2) Create objects through the File class, and the file stream class implements simple file operations.	3	Lectures, and experiment	Course Objective 2, 3
4	Experiment 4 Simple Thread Scheduling (1) Use the Thread class and Runnable interface to create threads (2) Use thread scheduling methods to manage and control thread objects.	(1) Use the method of Thread subclass to create a thread; (2) Use methods that implement the Runnable interface to create threads; (3) Use sleep and other methods to implement thread scheduling applications	3	Lectures, and experiment	Course Objective 2, 3, 4
5	Experiment 5 Simple Networking (1) URL to access network resources (2) Socket communication based on TCP	(1) Use URL class and URLConnection class to access network resources; (2) Understand the principle of Socket communication, and master the programming method of TCP communication using ServerSocket class and Socket class.	3	Lectures, and experiment	Course Objective 2, 3, 4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(3) Datagram communication based on UCP	(3) Master the programming method of UDP datagram communication			
6	Experiment 6 Simple Database Programming (1) Understand basic concepts such as database system and client/server model; (2) The definition of SQL language, the syntax of data manipulation and data query statements (3) The interfaces and classes provided by JDBC implement database operations.	(1) Master the use of JDBC to realize simple database management (2) Familiar with the JDBC core API, specify the database driver type, connect to the database, execute SQL statements, process result sets and other operation methods	3	Lectures, and experiment	Course Objective 2, 3, 4

V Period Distribution and Teaching Modes

V.1 Period Distribution for Theoretical Course Content

Teaching methods / Teaching hours / Course content	Theoretical Teaching	Assignment	Discussion	Note	Total
Chapter 1 Overview of Java Language	2				2
Chapter 2 Basics of Java Language	2				2
Chapter 3 Java Object-oriented	2				2
Chapter 4 Inheritance and Polymorphism	3		1		4
Chapter 5 Common Data Structures	3				3
Chapter 6 Java Exception Handling	2				2
Chapter 7 Java I/O Stream	4				4
Chapter 8 Graphical User Interface	1				1
Chapter 9 Java applet	1				1
Chapter 10 Java Multithreading	3		1		4

Chapter 11 Java Networking	3				3
Chapter 12 Java Database	3		1		4
Total	29		3		32

V.2 Period Distribution for Experimental Course Content

Name	Contents	Environments	Period	Group size	Attributes（Basic /Comprehensive/ Design/ Creative Study）	Requirements(Compulsory /Optional)
Experiment 1 Familiar with programming environment, editing, compiling and running Java programs	(1) The setting of programming environment, the editing, compiling and running process of Java Application and Java Applet. (2) Design and write Java Swing programs.	Eclipse, or other Java IDE, JDK 1.8	2	1	Basic	Compulsory
Experiment 2 Simple Exception Handling	Define an exception by yourself, programmatically create and throw an instance of an exception class, run the program and observe the execution result.	Eclipse, or other Java IDE, JDK 1.8	2	1	Basic	Compulsory
Experiment 3 Java IO	Write a graphical interface program, accept 5 floating point data and a file directory name input by the user, save the 5 data in a file, read out from the file and sort from largest to smallest, and then append again Save to file.	Eclipse, or other Java IDE, JDK 1.8	3	1	Basic	Compulsory
Experiment 4 Simple Thread Scheduling	(1) Use multi-threading technology to write a graphical interface program that displays a red bouncing ball	Eclipse, or other Java IDE, JDK 1.8	3	1	Basic	Compulsory

	program. When the ball hits the frame, it should bounce back from the frame and move 45° in the opposite direction. (2) Design and implementation of simple games based on multithreading (Choose one of the above)					
Experiment 5 Simple Networking	(1) Write a program to access a website and save a specified page locally. (2) Write a simple chat program (Choose one of the above)	Eclipse, or other Java IDE, JDK 1.8	3	1	Comprehensive	Compulsory
Experiment 6 Simple Database Programming	Use JDBC to write a simple database management system.	Eclipse, or other Java IDE, JDK 1.8	3	1	Comprehensive	Compulsory
Total			16			

V.3 Teaching Mode

This course is a very practical course. It must adhere to the principle of equal emphasis on theory and practice. On the basis of clarifying basic knowledge, special attention should be paid to case teaching. Course teaching fully implements the teaching idea of "learning within practicing, practicing within learning" and "projects as the carrier", and diversified teaching organization forms. According to different course contents, various kinds like "integrated teaching" and "example-based teaching" are adopted. "Integrated teaching" fully embodies the organic combination of learning and doing; "Example-based teaching" tried a thematic teaching organization method based on the main line "typical program examples→grammar, practical knowledge points→imitation programming→practice improvement". To encourage students to quickly master the key points of each topic in Java, so that students can experience the joy of success as soon as possible, enhance their confidence in learning, and stimulate curiosity, teachers should use classroom discussion, case analysis, and system analysis and design of actual application software to enhance students' confidence and ability in Java programming.

VI Assessment

The comprehensive assessment of this course includes classroom performance, online assignment, and experiment. Assessment score (Hundred Marking System) = usual grade * 50% + experiment grade * 50%.

The final grade is assessed by Five-Levels Marking System: A-Excellent (90 - 100), B-Good (80 - 89), C-Medium (70 - 79), D-Passing (60 - 69), E-Failure (Below 60).

<table>
<tr><th colspan="2">Assessment Methods or Approaches</th><th>Assessment Requirements</th><th>Assessment Weighting</th><th>Evaluation of Course Objectives</th></tr>
<tr><td rowspan="2">Usual grade</td><td>Classroom performance</td><td>Attendance, class interaction, class exercise</td><td rowspan="2">50%</td><td rowspan="2">Course Objectives 1, 2, 3</td></tr>
<tr><td>Online Assignment</td><td>Online assignment submit and scores</td></tr>
<tr><td colspan="2">Experiment grade</td><td>Classroom performance, experimental solution design, experimental operation, and experimental report</td><td>50%</td><td>Course Objectives 1, 2, 3</td></tr>
</table>

Note: 1. Assessment methods or approaches mainly include classroom performance, phase achievement, summary report, and testing defense, etc.

2. Assessment requirements include class interaction and testing requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

1. Textbooks

[1] Y. Daniel Liang. Introduction to Java Programming, China Machine Press, 2011.

2. References

[1] Yubo Jia, Qi Sun, Jing Shen, Chunxia Xu. Java Programming Foundation, Posts & Telecom press, 2013.

[2] Bruce Eckel. Thinking in JAVA. Pearson, 2006.

[3] Avinash C.Kak. Programming with Objects: A Comparative Presentation of Object Oriented Programming C++ and Java. Wiley-IEEE Press, 2008.

Written by：Tingting Wu **Reviewed by: Qi Sun**

Date：2021.05.20

Syllabus of Visual Programming B

Course Name/Title: Visual Programming B **Course code:** 62923

Course Type：(Basic Course, Specialized Course)(Optional Course)

Total Teaching Hours: 32 (Classroom Hours:24 Laboratory Hours:8)

Course Credit: 2

I Course Introduction

It is a new programming method, which is an important basic course for computer and related majors. It is an important way to cultivate and develop students' ability to analyze and solve problems, and to teach visual programming technology. The teaching goal of this course is to enable students to consolidate, deepen and expand their existing computer and information technology knowledge through systematic learning; Master the content and method of "object oriented and visual programming"; Combine their own professional knowledge and visual programming knowledge and skills, make them adapt to the requirements of the new era in the future study and work. Through classroom teaching and computer experiment, students can achieve the goal of knowledge and skills. And Through the introduction of a new programming method to solve the application of human-computer interaction in related fields, cultivate and develop students' ability to analyze and solve problems, and meet students' increasing spirit of hard work and craftsmanship.

II Course Objective

Course Objective 1: Through the explanation, from the application of ancient binary system to modern windows programming technology, we can promote patriotism; Through the introduction of a new programming method to solve the application of human-computer interaction in related fields, cultivate and develop students' ability to analyze and solve problems, and meet students' increasing spirit of hard work and craftsmanship.

Course Objective 2: Learn and master the basic knowledge of Visual C++, including the basic knowledge of C++, the development environment of Visual C++, the basic knowledge of windows programming, etc. Learn and master the basic methods of visual programming using Windows API functions, including windows drawing, text input/output, response of Windows applications to keyboard and mouse, application of resources, application of standard controls, etc. Understand the application of windows programming in many fields, and cultivate students' spirit of hard study and craftsmanship.

Course Objective 3: At present, the development platform of window application software is mostly "visual", especially the appearance of Visual C++ greatly promotes the application and development of object-oriented and visual programming technology. Therefore, mastering the contents and methods of "Object-oriented and visual programming" has become one of the requirements for college students' computer application and development ability.

Course Objective 4: Through the study of this course, we should pay attention to cultivate the following qualities of students: flexible and open, rigorous and careful way of thinking, not afraid of difficulties, down-to-earth work attitude, assiduous research, the learning spirit of excellence. The ability to explore the way to solve various problems from multiple perspectives;

Have a good scientific attitude and innovative spirit, and reasonably put forward new ideas, new concepts and new methods. Lay a solid foundation for future work. Cultivate students' practical ability and innovative consciousness.

III Correlations between Course Objectives and Graduation Requirements

Table 1 Correlation between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1: Engineering knowledge	1-2 Be able to establish appropriate mathematical models for complex engineering problems, and use basic theoretical knowledge to solve them.	Course Objective 1,2
Graduation Requirement 5: Using modern tool	5-2 Be able to use integrated development tools, open source and third-party resources to develop, debug and test computer systems, and understand their limitations,	Course Objective 1, 2,3
Graduation Requirement 12: Lifelong learning	12-1 Be able to recognize the necessity of continuous exploration and learning, and have the ability of autonomous learning and self-learning; The awareness of lifelong learning, master the method of autonomous learning, understand and expand knowledge; The ability and way of knowledge;	Course Objective 1, 4

IV Correlations between Course Content and Course Objectives

(1) Relationship between theoretical teaching contents and curriculum objectives（39hrs）

Table 2 Relationship between theoretical teaching contents and curriculum objectives

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 Fundamental of C++** (1) History of C++ (2) Review of C++	(1) The development history of C++; Encourage students to open their own operating system, improve national self-confidence;	2	Classroom teachin	Course Objective 1

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		(2) Explain and understand variables, data types and dynamic memory allocation with real life examples; (3) List the strategies we used in the revolutionary war to understand constructors and destructors;		g, discussion	
2	**Chapter 2 Windows Application Foundation** (1) Windows programming basics (2) Common messages of windows application (3) Design of event-driven in Windows (4) The basic structure of Windows Application	(1) Understand the basic knowledge of windows programming and guide students to understand the process of human-computer interaction; (2) Connecting with reality to guide students to understand the advantages of time driven programming; (3) Understand and apply the basic structure of windows application, and lay a good foundation for learning windows programming well.	2	Classroom teaching, discussion	Course Objective 1,2,3
3	**Chapter 3 Windows graphics device interface and windows drawing** (1) Graphical device interface (2) Drawing tools and colors (3) Common drawing functions (4) Application examples	(1) Understand the concept of graphical device interface; (2) Help students understand the concept of device independence by connecting with practice; (3) The graphic device interface is used to complete the graphic drawing, and the route map of Chinese revolution is drawn in combination with the process of the Red Army's long march in the history of Chinese revolution.	4	Classroom teaching, discussion	Course Objective 1,2,3
4	**Chapter 4 Font and its application** (1) Set device environment for text (2) The output process of text (3) Text operation examples	(1) Deeply understand the device environment of the text, and understand the concept of device independence again; (2) Encourage students to use the output of the text to write different styles of text; Training students to use the knowledge to complete the needs of real life.	2	Classroom teaching, discussion	Course Objective 1,2,3

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
5	**Chapter 5 The corresponding control of keyboard and mouse in Windows Application** (1) The application of keyboard in application program (2) Application examples of keyboard operation (3) The application of mouse in application program (4) Mouse application example	(1) Understand and master the response of windows to keyboard and mouse; (2) According to the requirements to achieve the required keyboard and mouse response; (3) Through practical examples, students are guided to understand and master the windows programming of keyboard and mouse response.	2	Classroom teaching, discussion	Course Objective 1,2,3
6	**Chapter 6 Application of resources in Windows program** (1) Menu and accelerator resources and their applications (2) Bitmap resource and its application (3) Dialog resource and its application (4) Application of icon resources	(1) Understand the types of resources and their applications; (2) Focus on the application of menu and accelerator resources in Windows programming; (3) Focus on mastering the types of dialog resources and their programming differences and connections (4) Understand and master the application of icon resources	4	Classroom teaching, discussion	Course Objective 1,2,3
7	**Chapter 7 Fundamentals of MFC** (1) Introduction of MFC (2) MFC class organization structure and the introduction of the main class (3) Global function and global variable in MFC (4) Application Wizard	(1) Familiar with MFC (2) Understand and master MFC class and windows programming based on MFC	8	Classroom teaching, discussion	Course Objective 1,2,3

(2) Relationship between experimental teaching contents and curriculum objectives (8hrs)

Table 3 Relationship between experimental teaching contents and curriculum objectives

Num.	Teaching Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Lab 1 Windows API programming** (1) Master the basic structure of windows application program (2) Use API function to build window and message loop, and program window function (3) Use API function to draw and output text in user area, and understand windows graphics device interface	(1) Familiar with the framework of windows programming; (2) Familiar with API function and its application; (3) Understand the significance of device independence in Windows programming.	2	Classroom teaching and lab	Course Objective 1,2,3
2	**Lab 2 Windows Resource Application** (1) Using API function to program windows application with various resources; (2) Realize the response of windows application to keyboard and mouse	(1) Understand and master various resources in Windows (2) Understand and realize the programming and implementation of windows resources (3) Combined with the current domestic and foreign situation, we should understand how to make rational use of resources to realize the scientific and technological power	2	Classroom teaching and lab	Course Objective 1,2,3
3	**Lab 3 Comprehensive application of MFC** (1) Using MFC to realize the programming of various windows controls	(1) Focus on the visual realization of various controls and response programming	4	Classroom teaching and lab	Course Objective 1,2,3

V Period Distribution and Teaching Modes

(1) Period distribution of theoretical courses

Table 4 Period distribution of theoretical part

Teaching Mode / Teaching period / Teaching Content	Theory teaching	Exercise	Discussion	Note	Total
Chapter 1 Fundamental of C++	1		1		2
Chapter 2 Windows Application foundation	2				2
Chapter 3 Windows graphics device interface and windows drawing	3		1		4
Chapter 4 Font and its application	1		1		2
Chapter 5 The corresponding control of keyboard and mouse in Windows Application	1		1		2
Chapter 6 Application of resources in Windows program	3		1		4
Chapter 7 Fundamentals of MFC	6		2		8
Total	**17**		**7**		**24**

(2) Period distribution of experimental courses

Table 5 Period distribution of experimental part

Name	Content	Environment	Period	Group size	Attributes (Basic/ Comprehensive / Design / Creative Study)	Requirements (Compulsory /Optional)
Lab 1: Windows API programming	In the visual C++ integrated development environment, the realization of the window, drawing and text output are completed.	Visual C++ IDE	2	1	Basic	Compulsory
Lab 2: Windows Resource	In the visual C++ integrated	Visual C++ IDE	2	1	Comprehensive	Compulsory

Application	development environment to complete the window resources, keyboard and mouse response.					
Lab 3 & 4: Comprehensive application of MFC	In the visual C++ integrated development environment, MFC is used to realize various controls of windows.	Visual C++ IDE	4	1	Comprehensive	Compulsory

(3) Teaching Modes

Change the traditional teaching method of simply instilling knowledge in the classroom, and implement heuristic and discussion teaching. When studying the teaching methods of the course, we should adopt different teaching methods according to the content and the students' conditions; Combine the methods of self-study, explaining key points and difficulties, organizing class discussion or interspersed discussion in teaching; To enable students to master the use of visual programming tools, the operation mechanism of event driven and the basic ideas and methods of programming; Can design, compile, debug simple and practical VC++ program, lay a good foundation for the follow-up computer application course.

Visual programming experiment is an important part of learning visual programming. It can consolidate and deepen classroom teaching content, improve students' practical work ability, cultivate scientific style, and lay a foundation for learning follow-up courses and engaging in practical technical work. Through the basic training of the experimental course, the students can initially have the basic knowledge of windows programming, master the principles and use methods of common development software, apply the software technology foundation to practice, and have the preliminary ability to deal with software development.

VI Assessment and Evaluation

(1) Assessment

Table 6 Course Assessment

Assessment Methods or Approaches		**Assessment Requirements**	**Assessment Weighting**	**Evaluation of Course Objectives**
Performan	Classroom	Attendance	30%	Course

ce	performance	Other performances (answering questions, discussion)		Objectives 1,2,3,4
	Homework	Submit homework on time and quality		
Experiment		According to the classroom performance, experimental scheme design, experimental operation and quality of experimental report evaluation.	20%	Course Objectives 1,2,3,4
Final Exam		Program Design	50%	Course Objectives 1,2,3,4

(2) Assessment criterion

The assessment of the performance is shown in Table 7-9.

Table 7 The Evaluation of the usual performance

Usual performance	Evaluation Criterion	Equation
Classroom performance A	⑦ According to the number of sign in, full attendance is 10 points. 0.5 point will be deducted for one time of being late and 1 point will be deducted for one time of absenteeism; The score is A1. ⑧ Classroom performance, full score 10. 1 point will be deducted if the question is not answered; The score is A2.。 ⑨ Small exercises in class, with a full score of 10. 1 point will be deducted if no answer is given in each exercise; The score is A3.	A=(A1+A2+A3)/30*100
Homework B	According to the number of completions, full completion is 10 points. 0.5 point will be deducted for one time of being late and 1 point will be deducted for one time of no submission; The score is B.	B
Total Score	**Usual Performance=A*40%+B*60%**	

Table 8 Evaluation of experiment

Aspects	The quality of Code	The quality of report	The quality of solution
Weight	50%	20%	30%
Excellent (A)	The code is very clear, with readable comments; results are correct; innovations are involved in	Text format is standard, the report is very clear and easily to be understood.	The problem and the existing solutions are reasonably analyzed; the solution embodies the innovative consciousness, and the result analysis is reasonable

	the experiments.		
Good (B)	The code is clear, with readable comments; results are correct.	Text format is fine; the report is clear and readable.	The problem and the existing solutions are reasonably analyzed; the solution is based on some existing solutions.
Medium (C)	Code format is standard; can achieve experimental objectives; individual conditions are not considered	Text format is not right; some contents are unclear and are difficult to be understood.	The solution is fine, but the analysis is not sufficient.
Pass (D)	Basically achieve the experimental goal	Have required contents.	The solution is workable.
Fail (E)	Do not achieve the experimental goal	Do not fully contain the required contents.	The solution is not workable.
Total	**The quality of Code*50% + The quality of report*20% + The quality of solution*30%**		

Table 9 Evaluation criteria of final exam

Type	**Emphasis on**	**Criteria**	**Percentage**	**Course Objective**
Window implementation	The result of program running is the expected window.	According to the requirements to achieve the basic style and requirements of the window.	40%	Course Objective 1,2
Comprehensive function realization	Including brush, brush, text output, mouse and keyboard response and API implementation of various resources	According to the requirements to achieve the examination of each score point	60%	Course Objective 3,4

VII Textbooks and References

(1) Textbook

Charles Petzold. Programming Windows (The sixth edition). Posts & Telecom Press, 2013.

(2) References

[1] Stanley B. Lippman. C++ Primer (The fifth edition). Addison-Wesley Professional, 2012.
[2] 孙鑫，VC++深入详解，电子工业出版社，2019.
[3] 黄维通、解辉，Visual C++面向对象与可视化程序设计，高等教育出版社出版社，2016.

Written by: Rong Jin **Reviewed by: Junfu Ma**

Date: 2021.05.20

Syllabus of Production Practice

Course Name/Title： Production Practice

Course code： 62724

Course Type： (Specialized Course)(Compulsory Course)

Total Teaching Hours： 40 (Laboratory Hours or Tutorial Hours 40)

Course Credit：2

I Course Introduction

Production practice is an important practical stage for computer science and technology students before entering the graduation project. In production practice, students can apply their ability in the design, development, and maintenance of computer systems in a systematic way; experience industry engineering project and typical software development process in the industry partner Further consolidate the theoretical knowledge learned, strengthen the integration of theory with practice, test and improve one's design and practical ability to prepare and lay the foundation for completing the graduation design task; at the same time, improve the practical ability and employment practice ability in the internship. Lay a solid foundation for practice after graduation. Through internships, students can also understand the needs of the society and enterprises for the knowledge structure and ability of graduates of the major, and adjust their knowledge structure in time to meet the needs of social development as soon as possible.

II Course Objectives

Course objective 1: Cultivate students to be brave in exploration, rigorous reasoning, serious and responsible for their work, meticulous, selfless dedication to the country and the collective, friendship and unity with colleagues, cooperation in tackling key problems, to be able to study things with great concentration, use practice to test theories, and comprehensively consider issues and other science and technology The ideological quality and work style that personnel should have.

Course objective 2： Through the internship and practice of the theoretical knowledge learned by the major, it is possible to write modules and algorithms for solving specific engineering problems.

Course objective 3： Through internships, establish the influence of the relevant entities on the computer system design and software development process, cultivate the awareness of quality control, and gradually cultivate a sense of social responsibility and humanities and social sciences.

Course objective 4：Through the necessary teamwork in the internship process, understand the role of team members in a multi-disciplinary background, carry out effective collaboration, and improve the ability of students to adapt to society as soon as possible.

Course objective 5：Master the knowledge and methods of engineering management and related cost control required for computer professional product development through on-site practice.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation requirement 3：Problem analysis	3.1 Able to design and develop modules or algorithms that meet specific needs for complex computer engineering problems	Course Objective 2
	3.4 Able to consider social, health, safety, legal, cultural and environmental factors in the design process.	Course Objective 3
Graduation requirement 9: Individuals and teams	9.2 Possess certain organizational and management capabilities, can reasonably formulate work plans, assign tasks according to the knowledge and ability characteristics of team members, and coordinate to complete work tasks.	Course Objective 4
Graduation requirement 11: project management	11.1 Able to understand and master engineering management principles and economic decision-making methods.	Course Objective 1，5

IV Correlations between Course Content and Course Objectives

No.	Course content	Teaching requirement	period	Teaching methods	Course objectives
1	1. Internship content guidance	For the content of the internship, teachers provide necessary guidance so that students can understand the content and requirements of the internship	5	Instructed by the instructor of the practice base	Course Objective 1,2
2	2. Teamwork to complete the internship content	Understand and master the needs of Take the group as a unit, complete the	35	Decentralized, mentor	Course Objective

No.	Course content	Teaching requirement	period	Teaching methods	Course objectives
		internship content in groups, give solutions, and implement them, and finally pass the teacher's review		guidance	3,4,5

V Period Distribution and Teaching Modes

Fig Period Distribution of practice hours

Name of experiment	Content point	Experiment environment	hours	Person for each group	property（basic/comprehensive/ design/innovation）	Requirement (required/optional)
Practice 1	Internship content guidance	Determined according to the content of the internship	5	3-5	comprehensive	required
Practice 2	Teamwork to complete the internship content	Determined according to the content of the internship	35	3-5	comprehensive	required

（2）teaching method

Production internships can be carried out in a combination of centralized internships and decentralized internships, which are arranged one academic year before graduation. The centralized internship location is the off-campus practice base of the major, and the scattered internship locations are determined by the students themselves. Before the internship, the students apply to the instructor and proceed after review and approval.

VI Assessment

Internship report/project review + work performance, of which internship report/project review accounted for 70%, and work performance accounted for 30%.

Assessment method	Requirements	Assessment weight	Evaluation goal
Internship report and project review	1. The internship report is written independently, no less than 2000 words, and the internship log is attached 2. Project review. Examples of evaluation criteria,	70%	Course Objective 2,3,4,5

	such as (overall situation (30), stability (15), business logic (15), UI (10), notes (10), complete on time (5), document (5), creativity (5), Project expression (5), the total of the above is 100 points, and the evaluation will be carried out according to each group)		
working performance	Attendance is checked by instructors for centralized internships, and by units for decentralized internships. Evaluation criteria such as initiative, collaboration, and constructiveness.	30%	Course Objective 1

*Description of points deduction for leave requests: Leave requests for centralized internships should be strictly controlled and generally not approved. Under special circumstances, the person who asks for leave may only grant leave at his discretion if he has sufficient reasons. In addition to affecting work performance, the corresponding percentage of points will be deducted for each leave (except for the exams arranged uniformly by the school). (For example, if the internship is 5 days in total, 10 points can be deducted for every half-day leave. If the deduction exceeds 30 points due to the leave, the internship may be judged as unqualified.)

VII Textbooks and References

（1）Reference book

According to the internship content, the internship base teachers will determine the reference books.

（2）Network resources

According to the internship content, the internship base teachers will determine the reference resource.

Written by: Wei Shen **Reviewed by: Jingsong Xia**

Date: 2021.05.20

Syllabus of Computer Network

Course Name/Title：Computer Network **Course code：62917**

Course Type： Specialized Course, Compulsory Course

Total Teaching Hours： 48 (Classroom Hours: 39 Laboratory Hours or Tutorial Hours 9)

Course Credit： 3

I Course Introduction

This is an undergraduate level course on computer networking. The goal is to teach the fundamental concepts and principles of networking, focusing particularly on how they apply to the Internet. We will cover the technologies supporting the Internet, from Ethernet and WiFi through the routing protocols that govern the flow of traffic, and the web technologies that are generating most of it. A major concern is understanding the protocols used on the Internet: how they work, their shortcomings, what the issues are, and what improvements are on the horizon.

II Course Objective

Course Objective 1: By explaining the leading position of Wechat, Beidou satellite navigation system and 5G Network in the international development, stimulate the students' feelings of home and country, sense of pride and sense of mission.

Course Objective 2: Analyze networked systems for performance bottlenecks and reason about improving performance.

Course Objective 3: Describe the mechanisms that make the Internet "work" from start to finish.

Course Objective 4: Design, implement, and simulate networked systems such as server/client applications, protocol stacks, and Internet infrastructure.

Course Objective 5: Recognize state of the art challenges and open questions in networking industry, including datacenters, enterprise networks, content distribution networks, wireless and cellular networks, and traditional network service providers.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
GR1 Knowledge	1-4 Be able to use engineering basic knowledge, professional knowledge and mathematical model for scheme comparison and analysis of computer complex engineering problems.	Course Objective 1,2,3
GR4 Investigation	4-3 Be able to analyze and interpret experimental data, and get reasonable and effective conclusions by information synthesis.	Course Objective 4(50%)
GR5 Modern tool usage	5-1 Be able to use software and hardware simulation tools to verify	Course Objective 4(50%)

	computer-related theories, simulate and analyze system design schemes, and understand its limitations.	
GR7 Environment and sustainability	7-2 Be able to evaluate and reasonably judge the hidden dangers of computer engineering practices that may cause harm to humans and the environment for actual projects.	Course Objective 5

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Unit 1: Introduction** (1)Overview and motivation of Resource sharing; (2)Protocols and standards. (3)Layering, OSI reference model, encapsulation. (4)End-to-end argument. (5)Protocol design issues; (6) Social Issues.	(1)Understand the basic theory of computer network; and the connection between scale and two types of transmission technology (broadcast links and point-to-point links); (2)Software structuring technique as a stack of layers or levels to reduce the design complexity, the leverage and content of every layer; (3)Apply reasoning to assess societal, safety, legal and cultural issues and the consequent responsibilities relevant to Internet.	6	Lecture	1,2 3
2	**Unit 2: Physical Layer** (1)The theoretical basis for data communication (2)Guided transmission media, wireless transmission. Digital modulation and multiplexing.	(1) Apply knowledge of mathematics to signal, analyses the limitation of data communication in different channels. (2) Learn how to choose and design the method to transport bits from one machine to another. (3) Apply the technical indicators of data communication to evaluate various physical media in terms of bandwidth, delay, cost, and ease of installation and maintenance. (4) Use digital modulation and multiplexing for more information capacity, better quality communications. (5) Through group discussion on China's achievements in the field of Beidou satellite navigation system, aroused national pride.	3	Lecture group discussion	1,2 3
3	**Unit 3: Data Links Layer**	(1) Achieve the data transfer algorithms for	6	Lecture	1,2 3,5

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(1)Data Link layer design issues; (2)Error detection and correction, CRC, Internet Checksum; (3) MAC protocols, CSMA/CD, Ethernet, Addressing; (4)Bridges, Spanning Tree, Flooding/Multicasting .	reliable efficient communication in the circumstance of finite data rate with errors and propagation delay. (2) Apply the algorithm in point-to-point or broadcast environment; (3) Solution of reliable transfer. CRC and Checksum algorithm.		group discussion	
4	**Unit 4: The Media Access Control Sublayer** (1)The channel allocation problem; (2)multiple access protocols; (3)Ethernet (4)switch	(1) Achieve the algorithm to allocate a single broadcast channel among competing users. (2) Model the dynamic scheme with correct assumptions. Understand multiple access protocols (3) Solution to isolate collision domain and broadcast domain. The self-learning of bridges	3	Lecture group discussion	1,2 3,5
5	**Unit 5: The Network Layer** (1) The best effort service model. (2)IP header structure. IP delivery review, options and encapsulation. (3)ARP and ICMP. (4)Routing protocols: distance vector vs link-state routing; (5) The Network layer in the Internet.	(1) Model the topology of the network, and appropriate paths choice algorithm, with consciousness of the scale of networks, balance between communication lines and avoiding overloading. (2) Describe the process when the source and destination are in different networks. (3) Compare the addressing solutions for IP. (4) Explain the ARP, ICMP that help the IP in layer mapping and error process.	9	Lecture	1,2 3,5
6	**Unit 6: Transport and Reliable Delivery** (1)Transport introduction; Port numbers, service models, UDP; (2)Intro to reliability; Intro to TCP. Flow control vs.	(1) Compare connection-oriented & connectionless. (2) Enhance the consciousness of the services and (3) Can choose API to tackle issues of reliability, connections and congestion control based on the service of IP layer.	6	Lecture	1,2 3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	congestion control; (3)Window-based and rate-based congestion control;	(4) Describe the Protocols such as TCP and UDP, describe how they use different methods to deal with transfer design and performance tradeoff.			
7	**Unit 7: The Application Layer** (1)DNS, the domain name system, (2) The world wide web	(1) Defining key application-layer concepts. (2) Examine World Wide Web in detail; (3) Using DNS to describe how a distributed system is built.	6	Lecture	1,2 3,5

1	**Experiment 1： LAN establishment** (1)Make twist-wire using clamping tool (2) Make the cross-wired cable and straight through cable according 568A and 568B specification. (3) Test wire using proper tools. (4) Setup a basic LAN with twisted-pair cables through the hub or switch	Recognize physical media. (1) Identify the factors that affect the transmission performance of physical media (2) Using standards to solve problems. (3) Explain the impact of the network performance related to the skill of making a twisted pair. (4) Setup a basic LAN with twisted-pair cables through the hub or switch.	3	Experiment	4
2	**Experiment 2：Observe a network packet using sniffer software** (1) In Ethernet, intercept ARP, ICMP protocol packets (2)Using filter to obtain wanted packets (3) Investigate the time sequence and protocol content of captured packets.	(1) Learn how to install and use capture software (2) Intercepted Ethernet data packets, and describes the frame structure and the datagram Ethernet packet head.	3	Experiment	4
3	**Experiment3：Investigate HTTP**	(1) the server/client application software architecture.	3	Experiment	4

	(1)the basic GET/response interaction (2)HTTP message formats (3)Retrieving large HTML files (4)Retrieving HTML files with embedded objects (5)HTTP authentication and security.	(2) Explain the behavior of a protocol. (3) Detail the design of protocol.			

V Period Distribution and Teaching Modes

4. Theory Lecture period distribution

Teaching mode / hours / contents	lecture	exercise	Discussion	remark	S total
Unit 1: Introduction	6				6
Unit 2: Physical Layer	3				3
Unit 3: Data Links Layer	6				6
Unit 4: The Media Access Control	3				3
Unit 5: The Network Layer	9				9
Unit 6: Transport and Reliable Delivery	6				6
Unit 7: The Application Layer	6				6
Total	**39**				**39**

5. Experiment period distribution

Unit	Content	Lab site	Hours	No.per Group	Experiment type	Compulsory
Experiment 1	LAN establishment **Content:** Practically implement the cross-wired cable and straight through cable using clamping tool, with proper testing.	Network Lab. of Information College	3	1	comprehensive	Yes

Experiment 2	Observe a network packet using sniffer software **Content:** In the environment of Ethernet, intercept ARP, ICMP protocol packets, do protocol analysis.	Network Lab. of Information College	3	1	design	Yes
Experiment 3	**Investigate HTTP** (1)the basic GET/response interaction (2)HTTP message formats (3)Retrieving large HTML files (4)Retrieving HTML files with embedded objects (5)HTTP authentication and security.	Network Lab. of Information College	3	1	design	Yes

1. Teaching methods

In this course, teacher should lecture in English at around 70 percentages. It is better to use case teaching. Teacher should adopt "question –result" mode to give students strong impression.

2. Teaching Ways

Teacher should use multimedia classroom to teach this course. The amount of time should be left to discussion for all students. The discussion class can be grouped. One group may quest; the other may answer the question.

VI Assessment

(Assessment content and methods must correlate with course objectives.)

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
the rates of attendance	Evaluate according to class attendance, participation and practice.	10%	Course Objectives 1、2、3、4、5
homework	3-4 times, Evaluate according to the hand in, accuracy, writing quality of homework.	10%	Course Objectives 2、3、5
experiment	Evaluate according to the experimental design, experimental operation and experimental report quality.	20%	Course Objectives 4

Final Exam (close-book)	Assess the correctness, the steps and the ideas of solving the questions.	60%	Course Objectives 1、2、3、5

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

1. Textbook

J. F. Kurose, K. W. Ross, Computer Networking, A Top-Down Approach Featuring the Internet, 7th edition or later Addison-Wesley.

2. References

[1] KAndrew S.Tanenbaum. Computer Networks (Fifth Edition), China Machine Press.2012

[2] Dimitri Bertsekas and Robert Gallager, Data Networks, 2nd edition or later

[3] W. Stallings, Data and Computer Communications, Prentice Hall, Sixth Edition, 2000, or later.

Written by：Xia jinsong **Reviewed by: Zhou zhiyu**

Date：2021.05.20

Syllabus of Curriculum Design for Computer Network

Course Name/Title：Curriculum Design for Computer Network **Course code：62937**

Course Type： Specialized Course, Compulsory Course

Total Teaching Hours： 20 (1 week)

Course Credit： 1

I Course Introduction

Curriculum Design for Computer Network is a curriculum design course related to *Computer*

Network.

Through a combination of lecture, hands-on labs, and self-study, you will learn how to install, operate, configure, and verify basic IPv4 and IPv6 networks. The course covers configuring network components such as switches, routers, and wireless LAN controllers; managing network devices; and identifying basic security threats. The course also gives you a foundation in network programmability, automation, and software-defined networking.

This course helps you begin the first step to prepare to take the 200-301 Cisco® Certified Network Associate (CCNA®) exam. By passing this one exam, you earn CCNA certification.

II Course Objective

Course Objective 1: Enhance students' sense of responsibility of "powerful country in science and technology" by introducing the evolution of compiler technology. Emphasizing the significant impact of basic software such as compiler and stimulate students' patriotic enthusiasm and cultivate their spirit of hard study and craftsmanship

Course Objective 2: Describe LANs and the role of switches within LANs; Identify and resolve common switched network issues and common problems associated with IPv4 addressing; Describe the TCP/IP Internet layer, IPv4, its addressing scheme, and subnetting; Explain host-to-host communications across switches and routers Explain the basics of dynamic routing protocols and describe components and terms of Open Shortest Path First (OSPF); Describe the operation, benefits, and limitations of static routing; Explain host-to-host communications across switches and routers.

Course Objective 3: Describe the features and functions of the Cisco Internetwork Operating System (IOS®) software, Implement basic configuration on a Cisco router. Explore functions of routing.

Course Objective 4: Describe network and device architectures and introduce virtualization. Validate the network design and debug the system in simulated environment.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
GR 1	1-3 Be able to use basic engineering knowledge, professional knowledge and mathematical models to derive engineering problems and analyze the influencing factors.	Course Objective 1,2

GR 3	3-1 Master the full-cycle development process, basic design/development and technology of computer systems and software products, and understand various factors that affect design goals and technical solutions.	Course Objective 3
GR 5	5-1 Be able to use software and hardware simulation tools to verify the computer related theory, simulate and analyze the system design scheme, and understand its limitations	Course Objective 4

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Usage of simulation tools	(1) The use of software and hardware simulation tools; (2) The limitation of software and hardware simulation tools;	2	Lecture/Experiments	1,4
2	Configure network interfaces	Help students to clarify the task objectives, understand the role of each component in their own design system, as well as the key factors of the design	2	Lecture/Experiments	2,3,4
3	Configure IPv4 Static Routes	Configure and Verify IPv4 Static Routes Implement IPv4 Static Routing	4	Lecture/Experiments	2,3,4
4	Configure RIP protocol	Configure and Verify RIP	4	Lecture/Experiments	2,3,4
5	Design and implementation of engineering network case	Network engineering requirement expression. Design and implementation of solution	2	Lecture/Experiments	2,3,4
6	Project test	Teachers give the project case, students design and implement the system using simulation software in given time.	2	Practice	2,3,4
7	Final written project document	Final written project document should at least include analysis, design, implementation and summary of four parts, illustrated, logical clear, standard format	2	Discussion/Writing	2,3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives

V Period Distribution and Teaching Modes

1. Practice period distribution

Course Content	Content	Device & lab enviroment	Hours	No.per Group	Experiment type	Compulsory
Usage of simulation tools	(1) The use of software and hardware simulation tools; (2) The limitation of software and hardware simulation tools;	network simulation tools	4	1	comprehensive	Compulsory
Configure network interfaces	Help students to clarify the task objectives, understand the role of each component in their own design system, as well as the key factors of the design	network simulation tools	4	1	comprehensive	Compulsory
Configure IPv4 Static Routes	Configure and Verify IPv4 Static Routes Implement IPv4 Static Routing	network simulation tools	8	1	comprehensive	Compulsory
Configure RIP protocol	Configure and Verify RIP	network simulation tools	8	1	comprehensive	Compulsory
Design and implementation of engineering network case	Network requirement expression. Design and implementation	network simulation tools	8	1	comprehensive	Compulsory

	of solution					
Project test	Design and implement a network using simulation tools in limited time.	network simulation tools	4	1	comprehensive	Compulsory
Final written project document	Final written document , including task content, purpose, requirements, indicators, business process design, function design, module design, code and operation combination and analysis	network simulation tools	4	1	comprehensive	Compulsory
总计			40			

2. Teaching methods

1. This course is mainly based on design and hands-on experiments. The teacher needs to explain to the students the nature, tasks, requirements, course arrangement and progress, assessment contents, experimental rules and laboratory safety system of the course.

2. Students must complete the whole process of design, implementation and debug. The project case has the overall complexity, which encourages students to challenge themselves.

VI Assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Final Document	the rationality, integrity and format standardization of the project design	30	Course objects 2,3,4
Project Test	Score according to the answers to the teacher's questions	60	Course objects 2,3,4
class attendance	Score according to the class attendance	10	Course objects 1

VII Textbooks and References

1. Textbook

J. F. Kurose, K. W. Ross, Computer Networking, A Top-Down Approach Featuring the Internet, 7th edition or later Addison-Wesley.

2. References

[1] KAndrew S.Tanenbaum. Computer Networks (Fifth Edition), China Machine Press.2012

[2] Dimitri Bertsekas and Robert Gallager, Data Networks, 2nd edition or later

[3] W. Stallings, Data and Computer Communications, Prentice Hall, Sixth Edition, 2000, or later.

Written by：Xia jinsong **Reviewed by: Zhou zhiyu**

Date：2021.05.20

Syllabus of Operating System A

Course Name/Title： Operating System A* **Course code：** 62903

Course Type： Specialized Course, Compulsory Course

Total Teaching Hours： 48 (Classroom Hours: 39 Laboratory Hours 9)

Course Credit： 3

I Course Introduction

The goal of this course is to introduce general knowledge of operating systems to Computer Science students. Topics cover the concepts of operating systems, Operating System Structure, Process, CPU Scheduling, Process Synchronization, Deadlocks, Memory Management, Virtual Memory, and File System This course will teach students basic knowledge of operating system, basic theory and basic methods, and enhance students' practice skills.

After completing the study of this course, students are required to have an overview of the basic concepts of operating system, and know what and how about operating system, especially in the following:

1. Understanding operating system history, its roles and features in the computer systems; familiar with resource management and scheduling.

2．Knowing about memory management, especially paging and segmentation.

3．Learning about Unix、Windows NT, Linux operating systems, and their architectures.

II Course Objective

Course Objective1: By explaining the advantages and disadvantages of our country in the field of operating system, we can promote patriotism and establish a correct outlook on life and values. Through the introduction of the design methods and ideas of the operating system, students' practical ability and innovative consciousness are cultivated.

Course Objective2: Understanding operating system history, its roles and features in the computer systems; familiar with resource management and scheduling.

Course Objective3: Knowing about Process, CPU Scheduling, Process Synchronization, Deadlocks, memory management, especially paging and segmentation.

Course Objective 4: Learning about Unix、Windows NT, Linux operating systems, and their architectures.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1	A systematic, theory-based understanding of the natural sciences applicable to the discipline	Course Objective 1,2,3
Graduation Requirement 2	Conceptually-based mathematics, numerical analysis, statistics and formal aspects of computer and information science to support analysis and modeling applicable to the discipline	Course Objective 1,2,3
Graduation Requirement 10	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions	Course Objective 1,3,4

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Unit 1：Introduction	1. Concepts of operating system Key points: definitions of operating system, purpose of operating systems. 2. History of operating system Key points: Batch systems, Multiprogrammed Systems, Time-Sharing Systems 3. Overview of types of operating systems Key points: Mainframe Systems, Desktop Systems, Multiprocessor Systems, Distributed Systems, Clustered Systems, Real-Time	3	classroom	Course Objective 1

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		Systems, Handheld Systems.			
2	Unit 2：Operating System Structure	System Components，Operating System Services，System Calls，System Programs，System Structure ，Virtual Machines，System Design and Implementation，System Generation.	3	classroom	Course Objective 1,2
3	Unit 3：Process	1. Process State Key points: process concept, Process State, Process Control Block 2. Process schedule Key points: process scheduling, process scheduler, context switch. 3. Process operations Key points: process creations, process termination. 4. Interprocess communications Key points: direct message communication, indirect message message, Remote Procure call.	3	classroom	Course Objective 2,3
4	Unit 4：CPU Scheduling	1. Scheduling Criteria Key points: Basic Concepts, CPU Scheduler, Dispatcher, Max CPU utilization, Max Throughput, Min Turnaround time, Min Waiting time, Min Response time. 2. Scheduling Algorithms Key points: First-Come, First-Served (FCFS) Scheduling, Shortest-Job-First (SJR) Scheduling, Priority Scheduling, Round Robin (RR), Multilevel Queue Scheduling, Multilevel Feedback Queues.	6	classroom	Course Objective 1,3,4
5	Unit 5：Process Synchronization	1. The concept of synchronization Key points: background of synchronization, bounded buffer produce- consume problem, Critical-Section Problem. 2. Solution to Critical-Section Problem	6	classroom	Course Objective 3,4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		Key points: Dekker's algorithm, Bakery Algorithm, Synchronization Hardware, Semaphores			
6	Unit 6：Deadlocks	1. Deadlock concepts Key points: Deadlock problems. 2. Deadlock Characterization Key points: four conditions about deadlock. 3. Deadlock Prevention Key points: breaking any of four conditions for deadlock. Deadlock Avoidance Key points: Safe State，Resource-Allocation Graph，Banker's Algorithm. Deadlock Detection Key points: Detection-Algorithm，Recovery from Deadlock	6	classroom	Course Objective 3,4
7	Unit 7:Memory Management	1. Early memory management Key points: Swapping, Contiguous Allocation. 2. Paging Key points: logical page, frame, page table, hierarchical page tables. 3. Segmentation Key points: segmentation table, segmentation architecture, segmentation hardware, Segmentation with Paging	6	classroom	Course Objective 3,4
8	Unit 8 ： Virtual Memory	1. Demand Paging Key points: concepts of virtual memory, page faults. 2. Page Replacement Key points: page replacement steps, page replacement algorithms.	3	classroom	Course Objective 3,4
9	Unit 9：File System	1. File-System Interface Key points: File Concept, Access Methods, Directory Structure, File System Mounting, File Sharing, Protection. 2. File System Implementation	3	classroom	Course Objective 1, 3,4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		Key points: File System Structure, File System Implementation, Directory Implementation, Allocation Methods, Free-Space Management.			
10	Experiment 1：process management	Linux shell commands, gcc. , fork（）, process synchronizations.	3	Lab	Course Objective 3,4
11	Experiment 2：process communications	Pipe, process communications by memory sharing. Process communications by message. Signals	3	Lab	Course Objective 3,4
12	Experiment 3：memory management	Design a program to test number of page faults in various page replacement algorithms.	3	Lab	Course Objective 1,3,4

V Period Distribution and Teaching Modes

2. Theory Lecture period distribution

contents / hours / Teaching mode	lecture	exercise	Discussion	remark	total
Unit 1：Introduction	3				3
Unit 2：Operating System Structure	3				3
Unit 3：Process	3				3
Unit 4：CPU Scheduling	6				6
Unit 5：Process Synchronization	6				6
Unit 6：Deadlocks	6				6
Unit 7:Memory Management	6				6
Unit 8：Virtual Memory	3				3
Unit 9：File System	3				3
Total	**39**				**39**

3. Experiment period distribution

Unit	Content	Lab site	Hours	No.per Group	Experiment type	Compulsory
Experiment 1	process management	Software Lab. of Information College	3	1	comprehensive	Yes
Experiment 2	process communications	Software Lab. of Information College	3	1	comprehensive	Yes
Experiment 3	memory management	Software Lab. of Information College	3	1	comprehensive	Yes

3. Teaching methods

In this course, teacher should lecture in English at around 70 percentages. It is better to use case teaching. Teacher should adopt "question –result" mode to give students strong impression.

Teacher should use multimedia classroom to teach this course. The amount of time should be left to discussion for all students. The discussion class can be grouped. One group may quest; the other may answer the question.

VI Assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
the rates of attendance	full	10	1,2,3
homework	10	20	1,2,3
experiment	report	20	2,3
Final Exam	close-book	50	1,2,3

VII Textbooks and References

1.Textbook

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne ：《Operating System Concepts》 , Higher Education Press John Wiley & Sons, Inc

2.References

[1]Abraham Silberschatz, Peter Baer Galvin, Greg Gagne ，郑扣根 译：《操作系统概念》（Sixth Edition）.

[2]汤子瀛、哲风屏、汤小丹编著：《计算机操作系统》（第二版），西安电子科技大学出版社，2001 年出版。

[3]Web Resource：http://www.linux.org/

Written by：Lican Huang **Reviewed by: Qiuhong Tian**

Date：2021.05.20

Syllabus of Software Engineering A

Course Name/Title：Software Engineering A **Course code**：**62947**

Course Type：(General Course，Basic Course ，**Specialized Course**)(**Compulsory Course**，Optional Course)

Total Teaching Hours： 39 (Classroom Hours: 56 Laboratory Hours or Tutorial Hours：9)

Course Credit：3

I Course Introduction

Software engineering is an engineering discipline that studies the general principles and techniques of software development and maintenance. Now it has become an important branch of computer science and technology, an extremely active research field. Strictly following the methodology of software engineering can greatly improve the success rate of software development and significantly reduce the problems in software development and maintenance. The course explains the basic concepts, principles and methods of software engineering from the perspective of software life cycle, covering four parts of relevant basic knowledge, including requirement modeling, design and implementation, test maintenance and development management. It emphasizes the fundamental and permanent principles of software engineering, such as testability, software architecture, modularity and reusability, and focuses on the complexity of software system and iterative development method, the ability of analyzing and solving problems and software engineering practice training, aims at cultivating students' good engineering development habits. Through the study of this course, students can master the process, methods and tools of software engineering, understand the design and application of software in large and complex systems from the perspective of engineering, and have the professional ability required to engage in software development as a software engineer.

II Course Objective

Course Objective 1：By explaining the current situation of China's software industry and the experience of being stuck, we can guide students' patriotism, social responsibility, cultural confidence and humanistic spirit; Form a team to complete a software analysis and design work, cultivate students' spirit of cooperation, work with others in the project team, and jointly complete the software project.

Course Objective 2：Master the basic concepts, principles and methods of software engineering, understand common software process models, understand agile software process, master basic software measurement and estimation methods, and be able to select feasible design schemes according to specific needs. By explaining the current situation of China's software industry and the experience of being stuck, we can guide students' patriotism, social responsibility, cultural confidence and humanistic spirit.

Course Objective 3： Be able to use RUP unified process to carry out iterative software development, master the basic process of requirement acquisition, requirement analysis, software architecture design, human-computer interaction design, detailed design, testing and maintenance, and understand the basic software verification methods in each stage. The software process model and life development planning are compared to guide students to think about their own learning direction and career planning according to their own needs.

Course Objective 4： Can accurately and effectively build target software analysis and design model and write design documents by using use case diagram, class diagram, sequence diagram and other graphical mechanisms and object-oriented analysis and design methods. Understand the data flow oriented method for structured software system analysis and design. Master at least one object-oriented analysis and design tool, software testing tool and configuration management tool, and be able to draw UML diagram, test software and manage version.

Course Objective 5： Be able to understand the important role of customers in requirements acquisition, and be able to use requirements engineering method to acquire and analyze software requirements from customers. Able to adapt to software architecture, form a project team with team members with multi-disciplinary background, communicate and communicate on the basis of object-oriented design model, and cooperate to complete the design of a software system. Form a team to complete a software analysis and design work, cultivate students' spirit of cooperation, work with others in the project team, and jointly complete the software project.

Course Objective 6： Understand the basic principles and process model of software project management, be able to make software project plan, master continuous integration methods and tools, so as to improve software development efficiency and target software product quality. In order to make students understand the laws and regulations of the software industry and professional ethics, we should select information security crime cases to carry out professional ethics education, and form correct professional concept and moral evaluation ability through inspiration and reflection.

III Correlations between Course Objectives and Graduation Requirements

Table 1 Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 2：Problem analysis	2-1 Can identify computer complex engineering problems, and through research, clearly express the needs and key processes of computer complex	Course Objective 1,3

	engineering problems.	
Graduation Requirement 3：Design/develop solutions	3-1 Master the whole cycle development process, basic design / development methods and technologies of computer system and software products, and understand various factors affecting design objectives and technical solutions.	Course Objective 2
Graduation Requirement 5：Using modern tools	5-3 Be able to use modern software engineering tools to manage the implementation process of software projects and understand their limitations.	Course Objective 4
Graduation Requirement 9：Individuals and teams	9-2 Be able to manage the team as a team leader, coordinate and organize the work of team members.	Course Objective 6
Graduation Requirement 10：Communications	10-2 Have the ability of listening, speaking, reading and writing in English, have an understanding of globalization and multiculturalism, and be able to carry out basic communication and exchange on computer professional issues under the cross-cultural background.	Course Objective 5

IV Correlations between Course Content and Course Objectives

（二）Theory contents（39 Periods）

Table 2 Theory Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 Introduction to Software engineering** (1) The concept of software crisis and software engineering.	(1) The paper expounds the causes of software crisis and defines software engineering; (2) The paper describes the common software development process model, and selects the process model reasonably according	6	Classroom teaching、Online courses、Discussion	Course Objective 1、2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(2) Software process model includes traditional software development model, software evolution model, object-oriented process model and agile model. **(3)** Software development principles.	to the situation; (3) The principles of agile development and the process of agile software development are described; (4) Through the explanation of the current situation of software industry and the experience of being stuck in the neck, the students' patriotic feelings, social responsibility, cultural confidence and humanistic spirit are guided. (5) It has been a lot of twists and turns to guide students to think about the development history of our country. Under the guidance of many heroes and advanced characters, the people have been exploring and striving constantly, and then the prosperity of today can be formed.			
2	**Chapter 2 Requirement analysis** (1) The concept and classification of software requirements; (2)The process model of software requirement acquisition and analysis; (3) Basic methods of demand survey; (4) Object oriented use case model and analysis model are established;	(1) Define the software requirements and specify the categories of software requirements; (2) Explain the important role of customers and their business knowledge in requirement acquisition; (3) The common methods of requirement acquisition are listed; (1) The object-oriented method is used to analyze complex system, and UML is used to model software requirements. (2) Using structured method, analyzing simple system, using data flow diagram for software requirements modeling. (3) The software process model and life development planning are compared to guide students to think about their own	10	Classroom teaching、Online courses、Discussion	Course Objective 1、3、4、5

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		learning direction and career planning according to their own needs.			
3	**Chapter 3 Software design** (1) The basic concepts of software design, including software design tasks, basic principles, component design and design documents; (2) Software architecture, user interface, detailed design process and method; (3) The representation of software architecture; (4) Architecture design pattern; (5) The representation of user interface design model.	(1) Define software design and describe the software design process model; (2) Detailed basic principles of software design (3) Based on the common mode, the software architecture is selected and the object-oriented method is used to model; (4) (4) Four kinds of elements of user interface are marked out and the object-oriented method is used to model; (5) The process and method of detailed design are described; (6) The paper compares five views in software architecture design with the reality to analyze problems from different perspectives, trains the dialectical view of things, and extracts the framework from details and simplifications of complex problems。	10	Classroom teaching、Online courses、Discussion	Course Objective 1、 3、4
4	**Chapter 4 Software testing** (1) The process and method of software debugging; (2) The basic concept of testing; (3) Black box test method and white box test method; (4) Software testing implementation strategy.	(1) Detailed debugging process, using a variety of methods for software debugging; (2) Detailed software testing process; (3) Define black box test and white box test; (4) Black box test and white black test are used to design test cases; (5) Mixed use of a variety of strategies to carry out software testing.	10	Classroom teaching、Online courses、Discussion	Course Objective 3、4
5	**Chapter 5 Software**	(1) know about maintenance categories,	3	Classroom	Course

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	maintenance (1) maintenance categories; (2) major causes of maintenance problems, reverse engineering and refactoring; (3) software evolution, and organizational and managerial issues.	understand major causes of maintenance problem; (2) know about reverse engineering and refactoring, master software evolution, and understand organizational and managerial issues.；		teaching、Online courses、Discussion	Objective 3、4、6

(二) Experimental contents（9 Periods）

Table 3 Correlations between experimental Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Experiment 1 Boat Renting Management system** (1) Use Visio, UML to draw use case diagram, class diagram and activity diagram; (2) According to the target system, analyzing requirement, design solutions and implement the system.	(1) Familiar with the use of UML drawing tools such as vision; (2) Familiar with the drawing of use case diagram, class diagram and activity diagram; (3) Can select appropriate elements to draw use case diagram, class diagram and activity diagram according to specific requirements. (4) Can design the solution and implement the solutions.	3	Classroom teaching、experiments	Course Objective 3、4、5
2	**Experiment 2 Home safety management system** (1) Use Visio, UML to draw use case diagram, class diagram and activity diagram; (2) According to the target system,	(1) Familiar with the use of UML drawing tools such as vision; (2) Familiar with the drawing of use case diagram, class diagram and activity diagram; (3) Can select appropriate elements to	3	Classroom teaching、experiments	Course Objective 3、4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
3	analyzing requirement, design solutions and implement the system.	draw use case diagram, class diagram and activity diagram according to specific requirements. (4) Can design the solution and implement the solutions.			
3	**Experiment 3 Software Testing** (1) Learn the use of JUnit library; (2) Draw the flow chart of program and design white box test cases with different coverage; (3) The black box test case is designed by using the method of equivalent test and boundary value analysis.	(1) Familiar with JUnit library; (2) Be able to select appropriate white box test method according to specific requirements; (3) Be able to select appropriate black box test method according to specific requirements; (4) Can draw program flow diagram and design test cases with different coverage.	3	Classroom teaching、experiments	Course Objective 3、4、5

V Period Distribution and Teaching Modes

（一）Period distribution of theory courses

Table 4 Period distribution of theory course

Teaching / Period / Course content	Theory teaching	Exercise course	Discussion		All
Chapter 1 Introduction to Software	5		1		6
Chapter 2 Requirement analysis	9	1			10
Chapter 3 Software design	9	1			10
Chapter 4 Software testing	8	1	1		10
Chapter 5 Software maintenance	3		1		3
All	44	6	6		56

（二）Period distribution of experiments

Table 5 Period distribution of experimental course

Name	Contents	Environments	Period	Group size	Attributes（Basic /Comprehensive/ Design/ Creative Study）	Requirements (Compulsory /Optional)
Experiment 1 Boat Renting Management System	Use Visio, UML to draw use case diagram, class diagram and activity diagram; According to the target system, analyzing requirement, design solutions and implement the system.	PC	3	1	Comprehensive	Compulsory
Experiment 2 Home safety management system	Use Visio, UML to draw use case diagram, class diagram and activity diagram; According to the target system, analyzing requirement, design solutions and implement the system.	PC	3	1	Comprehensive	Compulsory
Experiment 3 Software Testing	Learn how to use JUnit library; For a sentence, draw the program flow diagram and design test cases with different coverage; The black box test case is designed by using equivalence test and boundary value analysis.	PC	3	1	Basic	Compulsory

（三）Teaching Mode

There are many theories and methods in software engineering. To understand these theories and methods, we must adhere to the principle of attaching equal importance to theory and practice. On the basis of clarifying the basic knowledge, we should pay special attention to the practical training of students. In the classroom, we need to choose the case which is moderate difficulty and close to life, and explain the analysis and design work of each stage of software engineering in combination with the progress of theory course; After class, students can be asked to form groups of 3-5 and choose their own topics for analysis and design. Arrange the discussion and display to guide, revise and improve the students' design scheme.

In order to make students fully understand the key and difficult contents of this course, students are required to use UML drawing tools, rapid prototype tools and unit testing tools to analyze, design and test software system on the basis of mastering relevant theoretical knowledge of software engineering, so that students can better grasp the concept and methods of software engineering, and improve students' analysis of problems The ability to solve problems lays a good foundation for the follow-up study of professional courses.

In order to make students give full play to their learning initiative, MOOC courses are provided for students to preview. Some discussion questions are arranged on the teaching network platform for students to think and submit answers online.

VI Assessment

Table 6 Course assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Usual performances	Attendance Other performances (answering questions, discussion)	10%	Course Objective 1、2、3、4
Experiments	According to classroom performance, graphic correctness, program correctness, test coverage and experimental report writing quality evaluation.	50%	Course Objective 2、3、4、5
Final exam (closed-book exam)	According to the evaluation of the correctness of the answers to the test paper, the reasonable drawing and the correct thinking of solving the questions are given as appropriate.	40%	Course Objective 1、2、3

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

Textbook

（一）Textbook

[2] Roger S. Pressman, Bruce R. Maxim.《Software Engineering A Practitioner’s Approach》(Eighth Eidtion)， McGraw-Hill Education, 2014

（二）References

[4] Roger S.Pressman 等著，郑人杰等译，《软件工程-实践者的研究方法》（本科教学版）（第 8 版），机械工业出版社，2017 年 1 月

[5] 张海藩等编著，《软件工程导论》（第 6 版），清华大学出版社，2013 年 8 月

[6] Hassan Gomaa 等编著：《软件建模与设计：UML、用例、模式和软件体系结构》（第 1 版），机械工业出版社，2014 年出版。

（三）Resources

[1] https://mooc1.chaoxing.com/course/214799093.html

[2] https://www.xuetangx.com/course/THU08091000367/5883555

Written bv：Zuohua Ding **Reviewed by：Mingyue Jiang**

Date：2021.5.20

Syllabus of Curriculum Design for Operating system

Course Name/Title: Curriculum Design for Operating system* **Course code:** 62908

Course Type: Specialized Course, Compulsory Course

Total Teaching Hours: 20 (Laboratory Hours 20)

Course Credit: 1

I Course Introduction

Operating system is a basic Specialized course of computer science and technology major and a main course of computer major. It studies the control mechanism of effective cooperation of multiple resources in program execution. This course focuses on the basic knowledge, basic theory and basic methods , and provides the basic training of skills for design conception and operation .

II Course Objective

Course Objective 1: By explaining the advantages and disadvantages of our country in the field of operating system, we can promote patriotism and establish a correct outlook on life and values. Through the description of the design methods and ideas of the file system, students' practical ability and innovative consciousness are cultivated.

Course Objective 2: Through the research and analysis of Linux synchronization mechanism and file system, students design their own application scenarios, simulate the implementation of producer consumer problems and a small file system, deepen the understanding of the role, status and characteristics of the operating system in the computer system, and lay a solid theoretical foundation for the independent analysis and solution of practical problems in the future. By explaining the advantages and disadvantages of our country in the field of operating system, we can promote patriotism and establish a correct outlook on life and values.

Course Objective 3: It can comprehensively use these theories and methods to analyze and solve some application problems in engineering practice. And students' practical ability and innovative consciousness are cultivated.

Course Objective 4: Through the writing of curriculum design report and defensing their experimental demo , students' document writing ability and communication ability can be improved, and effective summary can be made.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 4.1	It can reflect the research consciousness in the process of solving complex computer engineering.	Course Objective 1,2,3
Graduation Requirement 6.2	Be able to analyze and evaluate the impact of professional engineering practice and complex engineering problem solutions on society, health, safety, law and culture, and understand the responsibilities.	Course Objective 1,2,3
Graduation Requirement 10.1	Communicate effectively on complex engineering activities with the engineering community and with society, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions	Course Objective 1,3,4

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Simulation of "producer consumer problem"	Master Java programming methods and common synchronization mechanism functions	6	Lab	Course Objective 1,2,3
2	Simulation of file system	Master Java programming methods and the functions of common file system commands	10	Lab	Course Objective 1,2,3
3	Writing curriculum design report	According to the standard requirements of software development, the text description and chart required in each stage are given, and the source code is attached, which is written according to the chapter directory provided in the task book.	3	Lab	Course Objective 3

V Period Distribution and Teaching Modes

1. Period Distribution

Unit	Content	Lab site	Hours	No.per Group	Experiment type	Compulsory

Unit 1	Simulate the Linux synchronization mechanism and file system	Software Laboratory of Information College	4	1	basic	Yes
Unit 2	Design and implementation of "producer consumer" and file system	Software Laboratory of Information College	8	1	design	Yes
Unit 3	Write a report on course design	Software Laboratory of Information College	2	1	Basic	Yes
Unit 4	Demonstration and inspection	Software Laboratory of Information College	2	1	comprehensive	Yes

2. Teaching methods

This experiment is mainly based on the computer. The teacher needs to explain to the students tasks, requirements of the course, as well as the schedule of the course, the contents to be assessed, assessment methods, experimental rules and laboratory safety system.

Each student has a computer, and the students must complete the designated experimental project. In order to cultivate students' practical operation ability and teamwork spirit, students can ask two people to complete the topic selection, preview, experiment and experiment report writing.

This course can be previewed and reviewed in spare time through the teaching courseware provided by teachers;

Download various supplementary materials through relevant websites to expand the breadth and depth of learning;

The selection of extracurricular exercises should focus on the teaching requirements, emphasize the training of basic exercises, reduce the requirements of skilled exercises, and try to select some assignments that can not only cultivate students' ability to analyze and solve problems, consolidate the learned knowledge, but also close to the practical application and stimulate students' interest in learning.

VI Assessment

This course adopts two combined assessment methods: defense and report.

1. Final assessment

The experimental report was graded according to five grades: excellent, good, medium, pass and fail.

The defense assessment is graded according to five levels: excellent, good, medium, pass and fail.

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
defence	Check the quality of completion, whether you can express your work in an orderly way, and whether student can answer the teacher's questions clearly and fluently	40	1,2
report	Write the report according to the standard	60	3

VII Textbooks and References

1.Textbook

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne ：《Operating System Concepts》，Higher Education Press John Wiley & Sons, Inc

2.References

[1]Abraham Silberschatz, Peter Baer Galvin, Greg Gagne ，郑扣根 译：《操作系统概念》（Sixth Edition）.

[2]汤子瀛、哲风屏、汤小丹编著：《计算机操作系统》（第二版），西安电子科技大 学出版社，2001 年出版。

[3]Web Resource：http://www.linux.org/

Written by：Lican Huang **Reviewed by: Qiuhong Tian**

Date：2021.05.20

Syllabus of Software Design Patterns

Course Name/Title： Software Design Patterns **Course code：62938**

Course Type： Basic Course (Optional Course)

Total Teaching Hours： 32 (Classroom Hours: 32 Laboratory Hours or Tutorial Hours : 0)

Course Credit：2

I Course Introduction

Software Design patterns is one of the optional basic courses open for the computer science and technology. In this course, the students will extend their knowledge of object-oriented analysis design by learning how to apply design patterns to addressing design issues. We help the students to further establish object-oriented way of thinking, and to make the developed software more reusable, flexible and maintainable. In addition, design patterns help to create a design vocabulary which makes it easier for developers to communicate.

II Course Objective

Course Objective 1: Demonstrate how to use design patterns to address user interface design issues. We encourage students to study for the rise of China with patriotism of learning enthusiasm, introducing the programming language evolution and the developing of design patterns which are widely applied to science and technology area such as national defense and military industry.

Course Objective 2: Identify the most suitable design pattern to address a given application design problem. Apply design principles (e.g., open-closed, dependency inversion, least knowledge).

Course Objective 3: Critique code by identifying and refactoring anti-patterns. We cultivate the practical ability and innovation consciousness of the students. Apply the model-view-controller architectural pattern.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 3	3-1 Master the full-cycle development process, basic design/development methods and techniques of computer systems and software products, and understand the factors that affect design objectives and technical solutions.	Course Objective 2
Graduation Requirement 3	3-3 Able to follow the software engineering specifications to achieve the requirements of the computer system, according to the design solution as well as fully consideration of cost-effective.	Course Objective 1
Graduation Requirement 5	5-2 Be able to understand and master modern engineering tools and information technology tools, as well as understand their technical advantages and limitations.	Course Objective 3

(Note: Basic courses and specialized courses must correlate with the graduation requirements as specified in the Program outline. The correlated graduation requirement index point must be put before the descriptive phrases or sentences. General courses are exempted from this rule.)

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 Introduction** (1) Object-Oriented Design: Review key points learned in Object-Oriented programming, including the categories of objects in design and the major design principles of object oriented programs (e.g. abstraction, encapsulation, decomposition, generalization). (2) Software Design Patterns Overview: Overview the design patterns, which provides practical proven solutions to recurring design problems.	(6) Identify different categories of objects: entity objects, boundary objects and control objects. (7) Master major principles of object-oriented programming: abstraction, encapsulation, decomposition and generalization. (8) Understand that organizing software using different objects and object-oriented design principles could make the code more flexible, reusable and maintainable. (9) List the examples of design patterns applied to software for military industry, and inspire students' patriotism of learning enthusiasm. Computer science technology is vital for national development especially for national defense. Mastering relevant knowledge speeds to arm our country with technologies.	6	Lecture	Course Objective 1
2	**Chapter 2 Creational & Structural Patterns** (5) Creational Patterns: Deal with the creation or cloning new objects. (6) Structural Patterns: Describe how objects are connected to each other. Review UML Diagrams.	(4) Master creational patterns such as Factory Method Pattern. (5) Master structural patterns which relate to the design principles of decomposition and generalization principles, such as Façade Pattern , Adapter Pattern and Composite Pattern. Examine UML diagrams for design patterns illustration.	9	Lecture	Course Objective 2、3
3	**Chapter 3 Behavioural Design Patterns**	(7) Describe the template method pattern. (8) Describe the chain of responsibility	9	Lecture	Course Objective

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(6) Behavioural design patterns focus on ways that individual objects collaborate to achieve a common goal. (7) Explain what a behavioural design pattern is and describe several popular behavioural design patterns such as template method pattern and the chain of responsibility pattern. Examine UML diagrams for example patterns.	pattern. (9) Describe the state pattern. (10)Describe the command pattern. (11)Describe the observer pattern. Use the story of how expert developers summarized those twenty-three design patterns to encourage students thinking and exploring their own ideas or methods during study. The way of solving problems could be used anywhere to make our country prosperous and thriving.			2、3
4	**Chapter 4 Working with Design Patterns & Anti-patterns** (5) We cover the Model View Controller (MVC) design pattern which builds upon design patterns presented in previous two chapters (Ch2 and Ch3). (6) Explain several general design principles that underlie design patterns. (7) Analyze and critique code for bad design. (8) Explain antipatterns (also known as code smells). Redesign applications with code smells using design patterns.	(6) Describe the Model View Controller (MVC) (7) Understand general design principles: a. the open/close principle b. the dependency inversion principle c. the composing object principle d. the interface segregation principle e. the principle of least knowledge (8) Identify code smells: a. duplicate code b. long method c. large class d. divergent class e. shotgun surgery f. long parameter list g. feature envy h. etc. Identify problematic software designs by referencing a catalogue of code smells and redesign applications using design patterns.	8	Lecture	Course Objective 2、3

V Period Distribution and Teaching Modes

（1）Period Distribution

Table 3 Period Distribution

Teaching Mode / Period / Course Content	Lecture	Tutorial	Discussion	Note	Hours
Chapter 1 Introduction	6				6
Chapter 2 Creational & Structural Patterns	8	1			9
Chapter 3 Behavioural Patterns	8	1			9
Chapter 4 Working with Design Patterns & Anti-patterns	6	1	1		8
Total	**28**	**3**	**1**		**32**

（2）Teaching Modes

Design issues in applications can be resolved through design patterns commonly applied by experts. This course extends your knowledge of object-oriented analysis and design by covering software design patterns used in interactive applications. We make most parts of this course as lectures together with proper amount of tutorials and discussion as well.

We examine UML diagrams for design patterns illustration which make students understand software design patterns better. The tutorial after each module not only summarizes key points and but also provides sufficient practice, improving students' ability of solving real design problems. The discussion of particular topics is also valuable. Besides, we also take the course feedback as first place thus we do quiz and investigation on the class.

VI Assessment

(Assessment content and methods must correlate with course objectives.)

(a) Assessment Method

The assessment of this course take both usual performance (quiz & assignment) and final exam into account. Total score is computed as: Quiz * 20% + Assignment * 20% + Final exam * 60%.

Table 4 Assessment Method

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Quiz	Score the quiz according to the answer correctness evaluation, also consider the steps correctness	20%	Course Objectives 1、2、3

	and the idea of solving problems for credits.		
Assignment	Score the assignments according to the answer correctness evaluation, also consider the steps correctness and the idea of solving problems for credits.	20%	Course Objectives 1、2、3
Final exam	Score the exam according to the answer correctness evaluation, also consider the steps correctness and the idea of solving problems for credits.	60%	Course Objectives 1、2、3

(b) Evaluation Criterion

Usual performance depends on class quiz scores.

Table 5 Usual performance evolution criterion

Usual Performance	Evaluation Criterion	Equation
Quiz	4 quizzes in total, and Ai represents the score (out of 100) of the i-th quiz, then we compute the average score of quizzes as A.	A=(A1+A2+A3+A4)/4
Assignment	4 assignments in total, and Bi represents the score (out of 100) of the i-th assignment, then we compute the average score of assignments as B.	B=(B1+B2+B3+B4)/4
Total	**Usual Performance = A * 20% + B * 20%**	

Table 6 Final exam evaluation criterion

Problem Type	Point	Criterion	Prop.	Objectives
Cloze/Multiple choice/Indefinite multiple choice	Investigate the fundamental definitions and concepts of design patterns and principles; the ability of identifying the most suitable design pattern to address a given application design problem; the ability of identifying code smells and redesign or refactoring the problematic applications.	Scoring according to the answer correctness evaluation.	60%	Course Objectives 1、2、3
Comprehensive design	Investigate the ability of applying design principles and design patterns when solving design problems; the ability of critique	Scoring according to the answer correctness	40%	Course Objectives 1、2、3

	code.	evaluation, also consider the steps correctness and the idea of solving problems for credits.		

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

（1）Textbooks

[3] Written by Erich Gamma and etc.，Translated by Yingjun Li and etc.: Design Patterns: Elements of Reusable Object-Oriented Software (English-Chinese Bilingual Edition) , China Machine Press, 2007.

（2）References

[7] Written by Freeman E. and etc.: Head First Design Patterns, China Electric Power Press, 2007.

[8] Written by Erich Gamma and etc.: Design Patterns: Elements of Reusable Object-Oriented Software, China Machine Press, 2019.

（3）Web resources

[3] https://refactoringguru.cn/design-patterns/

Written by：Zhiyi Luo **Reviewed by: Feng Fu**

Date：2021.05.20

Syllabus of Comprehensive Research on Intelligent Software

Course Name/Title： Comprehensive Research on Intelligent Software **Course code：** 62930

Course Type： Specialized Compulsory Course

Total Teaching Hours： 16 (Classroom Hours: 16 Hours)

Course Credit： 1

I Course Introduction

Comprehensive Research on Intelligent Software is a compulsory professional course for computer majors. This course combines basic theories with professional technologies. By studying this course, students can learn basic concepts, principles and models of intelligent software. Moreover, they grasp the development trend of intelligent softwares, and master the general development process of intelligent software. In fact, it is conducive to the cultivation of students' creativity and practical ability, and strengthens their knowledge so as to solve complex engineering problems.

II Course Objective

Course Objective 1: Master the basic concepts, principles and models of intelligent software. Distinguish the difference between intelligent softwares and traditional softwares. Learn the development trend of intelligent software. By analyzing the research on intelligent software in our country, it stimulates students' innovative spirit and craftsman spirit.

Course Objective 2: Master some intelligent algorithms and development environments of intelligent software design and developments. To recognize the bottleneck in the development technology of intelligent softwares so as to guide students to establish a national mission for the motherland and serves the people.

Course Objective 3: Master the basic development process of intelligent software through practical project cases. On the one hand, it cultivates students' creativity and innovation. On the other hand, it cultivates students' dialectical perspectives based on the demand analysis of practical problems, and develop dialectical system thinking.

III Correlations between Course Objectives and Graduation Requirements

Table 1 Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 4	4-1 Be able to analyze and study critical algorithms and modules in computer science, and also can verify and test them empirically.	Course Objective 2
Graduation Requirement 6	6-2 Be able to analyze the impact of engineering practices and their complex problem solutions on	Course Objective 3

	society, health, safety, law, and culture. Be able to evaluate the impact of constraints on project implementation, and understand some undertaken responsibilities.	
Graduation Requirement 12	12-2 Be able to meet the needs of personal professional developments, pay attention to the frontiers and trends in the computer field, learn new technologies independently, and adapt to the changement of environments.	Course Objective 1

(Note: Basic courses and specialized courses must correlate with the graduation requirements as specified in the Program outline. The correlated graduation requirement index point must be put before the descriptive phrases or sentences. General courses are exempted from this rule.)

IV Correlations between Course Content and Course Objectives

Table 2 Correlations between Theoretical Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 Overview of Smart Software** (1) Intelligent software concept and characteristics (2) The difference between smart software and traditional software (3) Basic principles and models of intelligent software (4) The development history of smart software	(1) Explain the concepts of smart software; (2) List the advantages of smart software over traditional software; (3) Explain the principles and mathematical models of intelligent software; (4) List the researches of our country on intelligent softwares so as to inspire students' innovative spirit and craftsmanship.	2	Classroom teaching; case studies.	Course Objective 1
2	**Chapter 2 Intelligent Software and Intelligent Algorithms** (1) Intelligent software and machine learning algorithms; (2) Intelligent software and deep learning algorithms; (3) Intelligent software and neural network; (4) Intelligent software and reinforcement learning.	(1) Explain the classification and clustering algorithm (2) Analyze Markov decision process (3) List DNN, CNN, RNN and GAN algorithms	8	Classroom teaching; case studies.	Course Objective 2
3	**Chapter 3 Intelligent Software Development Technology** (1) Intelligent software	(1) Build development environments for intelligent software; (2) Use smart software tools to	2	Classroom teaching; case	Course Objective 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
4	development environment, such as Python language and cloud platform; (2) Intelligent algorithm tools, such as Tensorflow, Pychar, Keras, etc.	develop smart modules. (3) Recognize the bottleneck in software development technology, and guide students to establish a national sentiment for the country and the people.		studies	
4	**Chapter 4 Intelligent Software Engineering** (1) Intelligent software life cycle; (2) Intelligent software product development and documentation; (3) Intelligent software testing; (4) Some cases of intelligent software engineering.	(1) Illustrate the process model of intelligent software; (2) Explain the development activities and document specifications of intelligent software; (3) Analyze intelligent software testing technology. (4) Design the general development process of intelligent software.	4	Classroom teaching; case studies; discussion	Course Objective 3

V Period Distribution and Teaching Modes

(1) Period distribution of theory courses

Table 3 Period distribution of theory course

Teaching Mode / Period / Course content	Theory teaching	Exercise course	Discussion		All
Chapter 1 Overview of Smart Software	2	0	0		2
Chapter 2 Intelligent Software and Intelligent Algorithms	8	0	0		8
Chapter 3 Intelligent Software Development Technology	2	0	0		2
Chapter 4 Intelligent Software Engineering	3	0	1		4
All	**15**	**0**	**1**		**16**

(2) Teaching Mode

The comprehensive research of intelligent software is a theoretical and research-oriented course with a corresponding training course. In theory course teaching, classroom teaching is the main method. In terms of teaching methods, theories can be taught through classroom discussion, heuristic teaching and case analysis so as to deepen the understanding and consolidation of theoretical knowledge. By assigning a certain amount of homework, it is helpful for students to consolidate the understanding of teaching contents.

In order to make students fully understand the key points and difficult contents of this course, students are required to analyze some existing engineering cases based on intelligent algorithms. By grasping the basic design process of intelligent softwares, it can improve students' ability to analyze and solve problems, and lay a good foundation for the follow-up study of professional courses.

VI Assessment

Table 4 Course assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Classroom performance	Attendance Other performances (answering questions, discussion)	20%	Course Objectives 1、2、3
Homework	Submit homework on time	20%	Course Objectives 1、2、3
Final performances (Report paper)	Theoretical basis, research methods, analysis and summary, and writing	60%	Course Objectives 1、2、3

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

[1] 雷明等编著：《机器学习与应用》，清华大学出版社，2018 年出版。

[2] 赫德林 德 庞特维斯著：AI 速成课(从 AI 编程到构建智能软件)，机械工业出版社，2020 年出版。

Written by：Dongming XIANG **Reviewed by: Wang LIN**

Date: 2021.5.20

Syllabus of Artificial Intelligence and Pattern Recognition

Course Name/Title： Artificial Intelligence and Pattern Recognition

Course code： 62700

Course Type： Specialized Course + Optional Course

Total Teaching Hours： 32 (Classroom Hours: 32)

Course Credit： 2

I Course Introduction

Artificial intelligence is a professional elective course for computer science and technology majors. It is a data processing and information analysis course that combines basic theory with professional skills and hands-on ability. Through the study of this course, students can fully understand the basic theories of artificial intelligence and train students in practical skills for data analysis and processing. Especially for the requirements of computer students, artificial intelligence courses can easily expand their knowledge and cultivate their ability to analyze actual complex engineering problems and deal with these problems.

II Course Objective

Course objective 1: By explaining the leading advantages and disadvantages of our country in the field of artificial intelligence, we will promote patriotism and establish a correct outlook on life and values. Through telling the design and application cases of artificial intelligence algorithm model, cultivate students' practical ability and innovative consciousness.

Course objective 2: Understand the knowledge framework and basic concepts of artificial intelligence, master the knowledge graph, swarm intelligence algorithm, machine learning algorithm logic, and its applicable scenarios and advantages and disadvantages; be familiar with artificial intelligence in computer vision, natural language processing, and speech processing And other aspects. Be able to apply literature research/scheme reasoning and other methods to understand the basic design methods of classic algorithm models; be able to read artificial intelligence program codes written by others, and be able to add or modify the content; design machine learning programs that implement specified tasks and functions by yourself, And debug, verify, and obtain basic training in experimental skills. Analyze application problems, design machine learning models, obtain data, process data, model debugging, model evaluation, and compile test documentation skills. Through actual hands-on programming and modeling training, cultivate students' hands-on practical ability and innovative consciousness.

Course objective 3: Through the introduction of several types of artificial intelligence algorithms such as swarm intelligence algorithms and deep learning algorithms, understand the latest development trends of artificial intelligence and the applications of artificial intelligence algorithms in various fields. Based on the constraints of the application of artificial intelligence algorithms, it can analyze the impact of the application of artificial intelligence algorithms from multi-dimensional factors such as society, health, safety, law, and culture, and the responsibilities that users need to bear. Learn about the outstanding contributions of some Chinese in the field of machine learning, promote national pride, and enhance the motivation of students to learn.

Course objective 4: Can read Chinese and English documents of artificial intelligence, can be used to track the frontiers of technical development in the professional field, and be able to conduct cross-cultural communication. Improve the ability to analyze and solve practical problems. It will lay a solid theoretical foundation for students to independently analyze and solve practical problems in the computer field in the future, and cultivate their hardworking spirit and craftsmanship.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 4	4-2 Based on scientific principles and scientific methods, students can formulate experimental plans, build experimental systems, and conduct experiments for the overall realization of computer complex systems.	Course Objective 1,2,
Graduation Requirement 6	Students can analyze and evaluate the impact of professional engineering practices and complex engineering problem solutions on society, health, safety, law, and culture, and understand their responsibilities.	Course Objective 1,3
Graduation Requirement 12	Students can meet the needs of personal professional development, pay attention to the forefront and trends of the computer field, independently learn new technologies, and adapt to the development of the times and changes in the environment.	Course Objective 1,4

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Chapter 1 Introduction (1) The origin and definition of	(1) Explain the origin and definition of artificial intelligence.	2	Classroom lectures,	Course Objective

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	artificial intelligence (2) The genre of artificial intelligence (3) Progress and development trend of artificial intelligence	(2) Explain the genre of artificial intelligence. (3) Explain the progress and development trend of artificial intelligence. (4) Explain my country's achievements in the field of artificial intelligence, promote patriotism, and establish a correct outlook on life and values.		online courses, classroom discussions	1、2、3
2	Chapter 2 Conceptual Representation and Knowledge Representation (1) Classical concept theory (2) Mathematical logic (3) Set theory (4) Modern representation theory of concepts (5) The concept of knowledge and knowledge representation (6) Production notation (7) Frame notation (8) State space representation	(1) Introduce the mathematical foundation of classical concept theory, mathematical logic, set theory, etc. (2) Introduction the modern representation theory of concepts. (3) Introduce the concept of knowledge and knowledge representation. (4) Explain the centralized methods of knowledge representation: production representation, frame representation, and state space representation.	2	Classroom lectures, online courses, classroom discussions	Course Objective 1、2、3
3	Chapter 3 Knowledge Graph (1) Ontology knowledge representation (2) World Wide Web Knowledge Representation (3) Briefly describe the current situation and development of the knowledge graph (4) Application examples of knowledge graph	(1) Explain the core of the knowledge graph, that is, the knowledge representation of the ontology. (2) Introduction of knowledge representation on the World Wide Web. (3) Briefly describe the current situation and development of the knowledge map. (4) Introduce the application examples of the knowledge graph.	2	Classroom lectures, online courses, classroom discussions	Course Objective 1、2、3
4	Chapter 4 Search Technology (1) Graph search strategy (2) Blind search (3) Heuristic search (4) Game search	(1) Detailed map search strategy. (2) Introduce the algorithm models of several search technologies, such as blind search, heuristic search, and game search. (3) Introduce the applicable scenarios, advantages, and disadvantages of several search technologies.	2	Classroom lectures, online courses, classroom discussions	Course Objective 1、2、3
5	Chapter 5 Swarm Intelligence Algorithm	(1) Briefly describe the background of the group intelligence algorithm.	4	Classroom lectures,	Course Objective

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(1) The background of the group intelligence algorithm (2) Genetic algorithm (3) Particle swarm optimization algorithm and its application (4) Ant colony algorithm	(2) Explain the algorithm logic of genetic algorithm, particle swarm algorithm, and ant colony algorithm, and application cases. (3) Through practical programming and modeling training, cultivate students' practical ability and innovative consciousness.		online courses, classroom discussions	1、2、3
6	Chapter 6 Machine Learning (1) The development of machine learning (2) Supervised learning (3) Unsupervised learning (4) Weakly supervised learning	(1) Introduce the development and current situation of machine learning. (2) Explain the classification of machine learning. (3) Introduce the classification and representative algorithms of supervised learning. (4) Introduce representative algorithms for unsupervised learning. (5) Introduce representative algorithms for weakly supervised learning. (6) Through actual hands-on programming and modeling training, cultivate students' hands-on practical ability and innovative consciousness.	6	Classroom lectures, online courses, classroom discussions	Course Objective 1、2、3、4
7	Chapter 7 Artificial Neural Networks and Deep Learning (1) The development history of neural networks (2) Neurons and neural networks (3) BP neural network and its learning algorithm (4) Convolutional neural network, recurrent neural network (5) Generate a confrontation network (6) Application of deep learning	(1) Briefly describe the development history of neural networks. (2) Detailed neuron structure and neural network architecture. (3) Explain the BP neural network and its learning algorithm logic. (4) Explain the algorithm logic of convolutional neural network and cyclic neural network (5) Explain the algorit.hm logic of generating a confrontation network. (6) Introduce application cases of deep learning. (7) Through practical programming and modeling training, cultivate students' practical ability and innovative consciousness.	6	Classroom lectures, online courses, classroom discussions	Course Objective 1、2、3、4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
8	Chapter 8 Expert System (1) Overview of expert system (2) Reasoning method (3) A simple expert system (4) Non-deterministic reasoning (5) Expert system tools (6) Application of expert system (7) Limitations of the expert system	(1) Introduce expert system and simple reasoning method. (2) Give examples of typical expert systems. (3) Explain the deterministic reasoning and non-deterministic reasoning used in the expert system. (4) Explain the expert system tools, the application of the expert system, and the limitations of the expert system.	2	Classroom lectures, online courses, classroom discussions	Course Objective 1、2、3、4
9	Chapter 9 Applications of Artificial Intelligence (1) Computer vision model and key technologies (2) Face recognition technology (3) Overview of natural language processing (4) Machine translation (5) Natural language human-computer interaction: intelligent question answering (6) Speech recognition, speech synthesis (7) Voice enhancement, voice conversion (8) Robots and multi-robots	(1) Introduce the application of artificial intelligence algorithms in computer vision and the corresponding algorithms. (2) Introduce the application of artificial intelligence algorithms in natural language processing and their algorithms. (3) Introduce the application of artificial intelligence algorithm in speech processing and its algorithm. (4) Introduce the application of artificial intelligence algorithms in the field of robot control and multi-robot collaboration. (5) Understand the outstanding contributions of some Chinese in the field of machine learning, promote national pride, and enhance the motivation of students to learn.	4	Classroom lectures, online courses, classroom discussions	Course Objective 1、2、3、4
10	Chapter 10 The Ethics and Future of Artificial Intelligence (1) Machine ethics (2) Shallow artificial intelligence (3) General intelligence (4) Discussion	(1) Explain the machine ethics involved in the development of artificial intelligence. (2) Introduce the feasibility of shallow artificial intelligence and general artificial intelligence.	2	Classroom lectures, online courses, classroom discussions	Course Objective 1、2、3、4

V Period Distribution and Teaching Modes

1. Period Distribution

Content	Lecture	exercise	Discussion	Others	Total
Introduction	2				2

Conceptual Representation and Knowledge Representation	2				2
Knowledge Graph	2				2
Search Technology	2				2
Swarm Intelligence Algorithm	4	1			4
Machine Learning	6	2			6
Artificial Neural Network and Deep Learning	6	2			6
Expert System	2				2
Application of Artificial Intelligence	4	1			4
The Ethics and Future of Artificial Intelligence	2				2
Total	26	6			32

2. Teaching Modes

This course is a highly theoretical and practical course with complex and cumbersome content. In the teaching method, attention should be paid to the comparison and explanation of algorithms, and certain discussions and exchanges should be arranged to help students deepen their understanding of the content of classroom teaching.

To enable students to fully understand the key and difficult content of this course, multimedia can be used for in-depth teaching during the teaching process, so that the two can be combined and complement each other, to teach intuitively and vividly, and improve teaching efficiency. Appropriate arrangements can be made for extracurricular computer practice to assist students in developing modeling skills.

VI Assessment and criterion

1. Assessment

Assessment Methods or Approaches		**Assessment Requirements**	**Assessment Weighting**	**Evaluation of Course Objectives**
Usual performance	homework	It is evaluated based on the number of times the homework is turned in, the accuracy rate, and the quality of writing.	60%	Course Objective 1、2、3、4
	Classroom test	According to the evaluation of the correctness of the test paper, the correctness of the answering		

		steps and the correctness of the problem-solving ideas are given points as appropriate.		
	Classroom performance	Based on attendance rate, classroom performance, and small class exercises.		
Final exam (open book)		According to the evaluation of the correctness of the test paper, the correctness of the answering steps and the correctness of the problem-solving ideas are given points as appropriate.	40%	Course Objective 1、2、3、4

2. Assessment criterion

The assessment of the usual performance is shown in Table 5,6.

Table5 The Evaluation of the usual performance

Usual performance	Evaluation criterion	Calculation formula
Classroom performance A	① According to the number of sign in, full attendance is 10 points. 0.5 point will be deducted for one time of being late and 1 point will be deducted for one time of absenteeism; The score is A1. ② Classroom performance, full score 10. 1 point will be deducted if the question is not answered; The score is A2.。 ③ Small exercises in class, with a full score of 10. 1 point will be deducted if no answer is given in each exercise; The score is A3.	A=(A1+A2+A3)/30*100
Homework B	① According to the number of submissions, 50 points for all submissions. 10 points will be deducted for one missed payment. The score is recorded as B1. ② Literature review assignments, according to the format, summary situation and length of reference documents as appropriate to give points. Excellent 40 points, good 40 points, medium 35 points, passing 30 points, and failing 30 points or less; the score is recorded as B2.	B=B1+B2
Experiments C	Test 3 times, with a total score of 100 points each time. According to the assessment of the correctness of the test paper, the correctness of the answering steps and the correctness of the problem-solving ideas will be given points as appropriate; the scores are recorded as C1, C2 and C3.	C=(C1+C2+C3)/3
Total	**Usual performance=A*30%+B*30%+C*40%**	

Table 6 Evaluation Criteria for Final Exam Scores

Question type	Focus of investigation	evaluation standard	Assessment Weighting	Evaluation of Course

				Objectives
Fill-in-the-blank/multiple choice/simple calculation (optional)	Learn the basic concepts and principles of artificial intelligence and be able to use basic knowledge of artificial intelligence.	According to the reference answer evaluation of the test paper, the correct answering steps and problem-solving ideas can be scored as appropriate.	60%	Course Objective 1、2、3、4
Comprehensive calculation and analysis questions	Master gradient descent algorithm, swarm intelligence algorithm principle, forward feedback thought in machine learning algorithm, etc., and use case design to achieve the purpose of proficient application of artificial intelligence algorithm.	According to the reference answer evaluation of the test paper, the correct answering steps and problem-solving ideas can be scored as appropriate.	40%	Course Objective 1、2、3、4

VII、Achievement index setting

The recommended weight of each assessment method on the evaluation of the achievement of curriculum objectives is shown in Table 7.

Table 7 Assessment weight setting and target achievement analysis

Evaluation W_{iik} / Course Objective	**Usual performance R_1 =60%**			**Final Exam R_3 =40%**
	Classroom performance S_{11} =0.3	**Homework S_{12} =0.30**	**Test S_{13} =0.4**	**Final Exam S_{31} =1**
Course Objective 1	0.60	0.40	0.50	0.50
Course Objective 2	0.20	0.30	0.40	0.40
Course Objective 3	0.20	0.20	0.10	0.10

VII Textbooks and References

1. Textbooks

[1] Stuart R, Norvig P., 《Artificial Intelligence : A Modern Approach, 3Rd Edition》,Upper Saddle River, New Jeysey, 2010.

2. References

[1] Mehrotra D. 《Basics of Artificial Intelligence & Machine Learning》, 2020.

3. Internet resources

[1] https://www.icourse163.org/course/ZJU-1003377027?from=searchPage

[2] https://www.icourse163.org/course/ZJUT-1002694018?from=searchPage

[3] https://www.icourse163.org/course/PKU-1002188003?from=searchPage

Written by：Liming Nie **Reviewed by: Tingting Wu**

Date：2021.05.20

Syllabus of Human-Computer Interaction

Course Name/Title： Human-Computer Interaction* **Course code：** 62941

Course Type：(General Course，Basic Course) (Compulsory Course)

Total Teaching Hours： (Classroom Hours: 32 Laboratory Hours or Tutorial Hours)

Course Credit： 2.0

I Course Introduction

This course is designed to introduce graduate students in computer science, psychology, educational psychology, and others topics to principles of and research methods in human-computer interaction (HCI), an interdisciplinary area concerned with the study of the interaction between humans and interactive computing systems. Research in HCI looks at major cognitive and social phenomena surrounding human use of computers with the goal of understanding their impact and creating guidelines for the design and evaluation of software and physical products and services in industry. The course consists of three modules: (1) principles of and literature in HCI through a set of readings, class presentations, and discussions, (2) empirical methods for exploratory and experimental human-subjects research in online lectures and tutorials and weekly assignments, and (3) a group project in which student teams will practice these principles and research methods in an application domain.

While the course is designed primarily for graduate students in computer science, psychology, and educational psychology, advanced undergraduates in these programs and graduate students from other programs may take the course with the permission of the instructor. Specifically, these students should meet with the instructor after the first class of the semester or during the first office hours of the semester to discuss their enrollment. No prerequisites are required to take the course.

II Course Objective

1. Ideological and political goals

Put Xi Jinping’s new era of socialism with Chinese characteristics throughout the curriculum, use cases to guide students to establish the great ideal of a community with a shared future for mankind, cultivate noble sentiments and patriotism, cultivate craftsmanship and scientific spirit, and continuously forge innovative ways of thinking in learning, and learn to be easy to use Good human-computer interaction courses serve the society.

2. Knowledge goal

Cognitive psychology and ergonomics are the theoretical basis of human-computer interaction technology, while multimedia technology and virtual reality technology and human-computer interaction technology intersect and penetrate each other. The research content of human-computer interaction is very extensive, covering modeling, design, evaluation and other theories and methods, as well as application research and development in Web, mobile computing, and virtual reality.

3. Abilities and personal qualities goals

The teaching purpose of this course is to broaden students' knowledge, so that students can understand and master the basic theories and principles of human-computer interaction, and can use relevant knowledge to solve problems according to actual needs.

4. Higher abilities goals

This course arranges 8 hours of practical courses, through practical learning and mastering the ability to analyze and solve problems using engineering tools.

5. Horizon broaden

According to students' interests and social employment needs, appropriate introductions to the development of cutting-edge technologies that expand students' horizons, key technologies in social or research fields, and special lectures that have a forward-inspiring effect on the future.

III Correlations between Course Objectives and Graduation Requirements

1. Course Structure

The course is designed to follow three modules — (1) principles of HCI, (2) human subjects research methods, and (3) project — and each hour of the three-hour class will focus on each module in this order. While there will be exceptions, e.g., project work spilling over to the second hour or the third component ending early, we will follow this format.

- Module 1: Principles of HCI

Principles of and literature in HCI is reviewed through a set of readings. Below is a list of topics that will are covered in this module. A comprehensive list of readings on these topics will be posted on the course website.

1. History and Foundations of HCI
2. Research Frameworks in HCI
3. Modeling Social and Emotional Processes
4. Computer-Mediated Communication
5. Social and Embodied Interfaces I
6. Social and Embodied Interfaces II

7. Computer-Supported Collaborative Work

8. Speech Interfaces

9. Games

10. Crowdsourcing

11. Information Visualization

12. Ubiquitous Computing

13. Assistive and Accessible Interfaces

14. Future of HCI

The first hour of class will include a brief introduction to that week's topic by the instructor, followed by an extended discussion led by students assigned serve as moderators. As a preparation for the lecture and discussion, students will be asked to complete an average of three readings for each topic and to write a 250-word essay that discusses one or several of the readings the beginning of class each week (i.e., a total of 14 essays throughout the semester).

The instructor will make moderator assignments a week in advance and post the names of the moderators along with the list of readings. The essays are due by midnight the day before class (Monday midnight) and should be submitted to Canvas.

- Module 2: Human Subjects Research Methods

A set of human-subjects research methods and procedures commonly used in HCI will be covered through lectures, tutorials, and weekly assignments. Lectures will be through online videos that students will be required to watch before class time, and class time will be used to discuss the lecture, run tutorials, work on assignments, and so on. Below is a list of the research methods and procedures that will be covered.

1. Introduction: Research methods in HCI

2. Introduction: What are elements of a research project in HCI?

3. Introduction: How to choose research designs?

4. Introduction: Methodological fit

5. Qualitative research: data collection

6. Qualitative research: data analysis

7. Quantitative research: Experimental design principles

8. Quantitative research: Step-by-step experimental design

9. Quantitative research: Measurement, Part I

10. Quantitative research: Measurement, Part II

11. Quantitative research: Measurement, Part III

12. Quantitative research: Scale construction

13. Quantitative research: Statistics, Part I

14. Quantitative research: Statistics, Part II

15. Quantitative research: Statistics, Part III

Students will practice a subset of these methods through six weekly assignments that include writing one-page reports of their process and findings. Students will submit their reports along with supplementary material to Canvas.

- Module 3: Project

Students will conduct a semester-long team project to explore HCI research in critical and emerging domains of computing from envisioning novel uses of wearable computing to studying how people

use and trust micro-finance practices. The goals of the project are:

Completing the required human-subjects research training program and an Institutional Review Board (IRB) application for the project,

Gaining a theoretical and empirical understanding of the application domain,

Applying exploratory and experimental research methods in HCI,

Prototyping user interfaces,

Designing exploratory and experimental studies,

Gaining experience in recruiting participants and conducting studies with human subjects,

Creating generalizable knowledge on how computing can improve aspects of human life.

The third hour of the class will be devoted to interim and final project presentations, class discussion, project group work sessions, and feedback on student work. Project teams will consist of three students. Students will develop project topics and form tentative project teams in the first week of classes through a class-wide brainstorming session and through discussions with the instructor.

Project teams and topics will be finalized by the second week of classes.

Project teams will informally present their progress at the milestones listed below (roughly every 2–3 weeks). Class times at other weeks will be used as work sessions. Below is the expected week-to-week timeline of the project, although students should expect changes in the timeline.

1. **Topic development** – Project topics are discussed as a whole class, and tentative project teams are formed.

2. **Team formulation** – Project teams are finalized and project topics are more formally defined. Project goals and timeline are discussed.

3. **Work session –** Teams work and seek feedback from the instructor during class time.

4. **Preliminary presentation** – Project teams will present their findings from a survey of

previous work in the problem area. The goal of this stage is to identify significant but unexplored phenomena in the problem area that teams will focus on for their exploration.

5. **Work session** – Teams work and seek feedback from the instructor during class time.

6. **Exploratory study design** – The teams will present their plans for an exploratory study that they will conduct to gain a deeper understanding of the identiLied phenomena.

7. **Work session –** Teams work and seek feedback from the instructor during class time.

8. **Exploratory study findings** – Project teams will present the findings from their exploratory study and the implications of these findings for a follow-up experiment that will further reLine their understanding of the explored phenomena.

9. **Work session –** Teams work and seek feedback from the instructor during class time.

10. **Experimental study design** – Students will present the design of a follow-up experiment informed by the findings from their exploratory study.

11. **Work session –** Teams work and seek feedback from the instructor during class time.

12. **Work session –** Teams work and seek feedback from the instructor during class time.

13. **Experimental study findings –** Teams will present the findings from their experimental study and discuss these findings in the light of the findings from their exploratory study.

14. **Work session –** Teams work and seek feedback from the instructor during class time.

15. **Final** – Project teams will present their overall project process and discuss the implications of their findings from the exploratory and experimental studies for HCI research and practice. Teams will also prepare their presentations in the form of a research poster and present their work to the department in a poster session that will be organized by the instructor.

Interim milestones will involve informal team presentations and classroom discussion. The final presentation will be formal and considered as the final exam of the class. The final presentation will be scheduled either on the last class of the semester or during the campus-designated final exam slot.

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
4.2	Module 1 and Module 2	Course Objective 1,2,3
6.2	Module 1 and 2 and Module 3	Course Objective 2,3,4
12.2	Module 2 and Module 3	Course Objective 3,4,5

(Note: Basic courses and specialized courses must correlate with the graduation requirements as specified in the Program outline. The correlated graduation requirement index point must be put before the descriptive phrases or sentences. General courses are exempted from this rule.)

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Lecture: Course Introduction Project: Introduction and Brainstorming		2	Lecture Q&A	1,2
2	Reading and Discussion: History and Foundations of HCI		2	Lecture Q&A	1,2
3	Project: Ideation & Team-Formation Online Lecture + Reading: Introduction to HCI Methods Project: Team Formation & Topic Selection		2	Lecture Q&A	1,2,3
4	Assignment 1: Human Subjects Research Training		2	Experiment	2,3
5	Reading and Discussion: Research Frameworks in HCI		2	Reading & Discussion	2,3
6	Discussion: Ubiquitous Computing Q&A + HoA: Methodology Matters Online Lecture + Reading: Methodology Matters Project: Worksession		2	Lecture Q&A	2,3
7	Reading and Discussion: Modeling Social and Emotional Processes		2	Reading & Discussion	2,3,4
8	Discussion: Assistive and Accessible Interfaces Q&A + HoA: Ethnography Online Lecture + Reading: Ethnography Online Lecture + Reading: Methodological Fit (Optional) Project: Interim Presentation, Research Question		2	Reading & Discussion Q&A	2,34
9	Reading and Discussion: Social and Embodied Interfaces		2	Reading & Discussion	3,4
10	Discussion: Crowdsourcing Q&A + HoA: Qualitative Data Analysis Online Lecture + Reading:		2	Discussion Q&A	3.4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	Qualitative Data Analysis Project: Worksession				
11	Reading and Discussion: Computer-Mediated Communication		2	Reading & Discussion	3,4
12	Discussion: Physiological Computing Q&A + HoA: Experimental Design Basics Online Lecture + Reading: Experimental Design Basics Project: Interim Presentation, Study Design		2	Reading & Discussion Q&A	3,4
13	Assignment 2: Qualitative Research		2	Experiment	4.5
14	Reading and Discussion: Computer-Supported Collaborative Work		2	Reading & Discussion	4,5
15	Discussion: Modeling Social and Emotional Processes Q&A + HoA: Step-by-step Experimental Design Online Lecture + Reading: Step-by-step Experimental Design Project: Worksession		2	Reading & Discussion	4,5
16	Discussion: The Future of HCI Project: Final Presentation		2	Discussion and Q&A	5

V Period Distribution and Teaching Modes

Communication: All class material will be available on the class Canvas page. Assignment handouts, readings, supplemental materials, and pointers to other resources will be posted on the course website. Announcements will be made through Canvas announcements. Students should set up their notification settings in Canvas in the way that supports receiving announcements on time. All class-related communication must be done using the Canvas messaging system. Email should not be used for class-related communication.

Office Hours: Office hours are the best time to get feedback from the instructor on assignments and projects. Other questions, concerns, individual issues, and team communication problems can be discussed by appointment. The instructor will also be available for questions via email anytime and phone during work hours.

Classroom recording: Because many of the pictures and videos we present in class are copyrightprotected, no student may record or tape any classroom activity without the express written consent of the instructor. If a student has a disability that requires him/her to record or tape classroom activities, he/she should contact the University Student Center to arrange an appropriate accommodation.

VI Assessment

(Assessment content and methods must correlate with course objectives.)

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Be in Class	Exemption for asking	20%	Absent less than 4
Performance in class	Yes	30%	
Experimental & Report	Yes	50%	

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

1. **<u>Textbooks</u>**

Helen Sharp, Yvonne Rogers and Jennifer Preece主编(或编著):《INTERACTION DESIGN beyond human-computer interaction》， John Wiley & Sons, Inc.出版社，2019年出版。ISBN: 978-1-119-54725-9

2. **<u>References</u>**

吴亚东，张晓蓉，王赋攀主编（或编著）：《人机交互技术及应用》，机工出版社，2020年出版。ISBN：9787111655268

3. **<u>Web Resources</u>**

[1] http://sonify.psych.gatech.edu/~walkerb/classes/ms-hci/index.html

[2] https://canvas.cornell.edu/courses/20084/assignments/syllabus

Written by：Yanfie Liu **Reviewed by: Dongming Xiang**

Date：2021.05.20

《Python程序设计高阶》教学大纲

课程中文名称：Python 程序设计高阶　　　　　　　　**课程代码：**62812

课程英文名称：Advanced Python Programming

课程类别与性质：专业课、选修

总学时：32 学时（其中讲课 22 学时，实验、上机或课外实践 10 学时）

学　分：2

先修课程：C 语言程序设计，面向对象程序设计

面向对象：计算机科学与技术专业

开课系(室)：计算机科学与技术系

一、课程简介

本课程是面向计算机大类的通识课程，主要讲授 Python 语言及其应用。目前由于 Python 语言的简洁性、易读性以及可扩展性，已经在 Web、机器学习等领域内得到了广泛的应用，特别是因为 Python 免费，扩展性好的特点，已经有取代 Matlab 而成为工程计算首选的趋势。本课程在讲授 Python 语言的基础上，通过实际应用案例的讲授和实践，让学生能够掌握使用 Python 语言来解决具有一定复杂程度的理论、工程和社会问题。

二、课程教学目标

课程目标 1：树立正确的职业荣誉感与使命感，具备认真严谨的学习和工作态度，为其今后从事相关的专业工作打下正确的思想基础。

课程目标 2：掌握 Python 语言的语法，了解 Python 语言的特点，掌握 Python 语言包括列表、元组等在内的常用数据类型，能够应用这些数据类型，引入常用的包，构建函数、类等模块，进行模块化或面向对象编程。

课程目标 3：能够 Python 程序编程过程中，利用搜索引擎和其他信息检索工具，获取已有的实现特定功能的程序和包，通过 pip、anaconda 等工具，在 Python 编程环境中进行安装，并进行认知和分析，以确定相对较优的参考解决方案。

课程目标 4：在编程过程中，能够阅读已有的 Python 程序和包的文档，来了解这些程序和包的使用方法和过程，并能够在自己的程序中，对这些程序或包进行使用、模仿、修改和改进。

课程目标 5：能够以组为单位，针对具有一定复杂性的 Python 编程问题，能够探讨与组内的成员分工，在明确各自的工作基础上，共同分析问题，完成分担的工作，并针对随机选取的成员的答辩进行共同准备。

课程目标 6：能够以组为单位，对共同完成的基于 Python 编程的解决方案，使用性能

分析（如使用 time 包）、测试结果（如使用 matpoltlib 包进行可视化分析）等方式和方法，分析该解决方案是否达到问题的需求，并给出分析的过程和结果。

三、课程教学目标与毕业要求的对应关系

表 1 课程目标与毕业要求的对应关系

<table>
<tr><th>毕业要求</th><th>毕业要求指标点</th><th>课程目标</th><th>支撑分析说明</th></tr>
<tr><td rowspan="3">2.问题分析：能够应用数学、自然科和工程的基本原理，识别表达并通过文献研究复杂以获得有效结论。</td><td rowspan="2">2.2 能够通过文献调研/方案推理等方法寻求计算机应用领域复杂工程问题的多种解决方案，并能确定适合具体问题的方案。</td><td>课程目标 2</td><td>Python 语言是本课程解决问题的工具，因此，在寻求并确定适合具体问题的解决该方案时，必须具备对 Python 语言的认知和使用 Python 语言编程的能力，才能做到这一点。</td></tr>
<tr><td>课程目标 3</td><td>Python 程序设计过程中，使用搜索引擎、GitHub 工具来获得已有的或相近的程序实例，以此参考或基础，进行分析，来构建适合的解决方法，是 Python 程序设计中常用的做法。</td></tr>
<tr><td>2.3 能够运用数学、自然科及工程的原理计算机专业方法和技术，分析和评估具体的计算机应用复杂工程问题的解决方案，并能给出具体指标和有效结论</td><td>课程目标 6</td><td>对基于 Python 程序的解决方案，有多种方法可以分析、评价程序的执行情况和结果，如可以通过使用 time 包来确定程序的执行时间，从而推断程序执行的效率，在此基础上，可以通过使用 matplotlib 进行可视化分析等方法，或其他的方式来分析整体解决方案是否达到需求的要求。</td></tr>
<tr><td>9. 沟通交流：能够就复杂计算机工程问题与业界同行及社会公众进行有效沟通和交流，包括设计文稿、撰写报告、陈述发言、清晰表达或回应指令。并具备一定的国际视野，能够在跨文化背景下进行沟通和交流。</td><td>9.2 能够就复杂工程问题进行有效的书面和口头表述，并与他人进行有效沟通，包括撰写报告和设计文档、陈述发言、清晰表达或回应指令。</td><td>课程目标 5</td><td>要求以组为单位完成具有一定复杂性的基于 Python 编程的问题，在小组内进行分工协作，沟通交流，最终共同形成整体的解决方案，由于答辩人在小组内随机选定，所以要求每人都必须了解整个小组成员的工作情况。为了达到这一要求，小组内交流和答辩准备都是必要的。</td></tr>
</table>

12. 终身学习：具有自主学习和终身学习的意识，有不断学习和适应发展的能力。	12.1 理解终身学习的重要性，形成终身学习的意识，适应持续的职业发展。	课程目标 4	获取已有 Python 程序和包的，并在自己的程序中进行使用、模仿、修改和改进，是一个循序渐进的过程，本身就一个学习、积累的过程，能够培养和形成学习的意识。
		课程目标 1	终身学习思想建立过程本身就是建立正确的学习和工作态度的过程，能够为今后相关工作打下正确的思想基础。

四、教学内容与课程目标的关系

（一）理论教学内容与课程目标的关系（22 学时）

表 2 理论教学内容与课程目标的关系

序号	教学内容	教学要求	学时	教学方式	对应课程目标
1	1. 概述和基本程序设计 （1）Python 历史； （2）Python 语言环境安装; （3）基本程序设计；	了解 Python 语言，掌握 Python 语言开发环境的安装，掌握 Python 语言常用的数据类型，运算符和运算符的优先级，表达式，语句;能够写简单的 Python 程序进行屏幕输出	2	课堂讲授、在线课程、课堂讨论、课后查找资料	课程目标 2、3
2	2. 分支和循环 （1）分支结构 （2）循环结构	掌握 if-else 语句的使用，掌握循环语句的使用	2	课堂讲授、在线课程、程序案例分析	课程目标 2、3
3	3. 序列类型 （1）列表 （2）枚举 （3）字典 （4）集合	掌握 Python 列表等序列类型，能够使用序列类型解决一些实际问题	4	课堂讲授、在线课程、提问讨论	课程目标 3、4
4	4. 函数	掌握函数定义、调用方法，能够编写达到目的的函数	2	课堂讲授、在线课程、程序案例分析、课堂讨论	课程目标 2、3、4
5	5. OOP 编程 （1）类和对象	掌握 Python 语言类和对象的概念，掌握继承和多态，能够使用面向对象思想	4	课堂讲授、在线课程、	课程目标 3、4

序号	教学内容	教学要求	学时	教学方式	对应课程目标
	（2）多态	进行编程		程序案例分析、课堂讨论	
6	6.文件和异常	掌握 Python 语言文本文件和二进制文件的建立和读写，目录遍历，异常的分类，处理异常的方法	2	课堂讲授、在线课程、课堂讨论、课后查阅资料	课程目标 3、4
7	7.爬虫 （1）http 请求，请求类型 （2）爬虫	掌握 Http 请求的常用类型，使用工具 Postman 发起请求，掌握 Request 包的安装，编程使用 Request 包发起请求，并处理响应数据	4	课堂讲授、在线课程、课堂讨论、课后查阅资料	课程目标 3、4、5
8	8.Web 服务	掌握 Tornado 包的安装和引入，使用 Tornado 包和 OOP 编程方法实现一个基础的 Web 服务框架	2	课堂讲授、在线课程、课堂讨论、课后查阅资料	课程目标 3、4、5

（二）实验教学内容与课程目标的关系

表 3 实验教学内容与课程目标的关系

序号	教学内容	教学要求	学时	教学方式	对应课程目标
1	实验 1 Python 语言环境安装，基本程序设计，选择和循环，调试	掌握 Python 语言开发环境的安装，掌握 Python 语言常用的数据类型，运算符和运算符的优先级，表达式，语句，选择和循环编程，pdb 的使用； 能够自己建立 Jupyter 开发环境，编写九九乘法表，并使用级数来求 π 的近似值	2	课堂讲授、实验	课程目标 2、3

序号	教学内容	教学要求	学时	教学方式	对应课程目标
2	实验 2 有序类型和函数 列表、元组、字典、集合和函数	掌握列表、元组、字典、集合和函数的使用，学会结合数据类型和程序控制结构完成特定功能模块的设计实现	2	课堂讲授、实验	课程目标 2、3、4
3	实验 3 正则和 OOP 字符串，正则表达式和 OOP 编程	掌握字符串的使用，使用正则表达式匹配字符串的方法，并使用面向对象编程来实现模块化程序设计； OOP 编程使用 re 包去掉 html 文本中的 html 标签	2	课堂讲授、实验	课程目标 2、3、4
4	实验 4 爬虫和 Web 服务	掌握 Http 请求和后端服务的概念，并理解如何使用请求获取数据，和响应请求返回数据； 以小组为单位，在参考相关实例的基础上，使用 Request 和 Tornado 包实现爬虫和 Web 服务程序	2	课堂讲授、实验	课程目标 3、4、5、6
5	实验 5 综合实验	以 Python 程序设计为基础，以组为单位，通过知识和方法检索，使用各种工具来协作给出复杂问题的解决方案，并能够对方案进行评估； 通过团队分工和协作，给出解决方案，并完成答辩	2	课堂讲授、实验	课程目标 1、3、4、5、6

五、课程学时分配及教学方法

（一）课程学时分配

表 4 理论教学内容课时分配表

教学方式 教学时数 课程内容	理论讲授	习题课	讨论课	备注	小计

概述和基本程序设计	1		1		2
选择、循环和调试	2				2
列表、元组、字典和集合	2		2		4
函数	2				2
OOP 编程	3		1		4
文件和异常	2				2
爬虫	4				4
Web 服务	2				2
总　　计	18		4		22

表 5 实验教学内容课时分配表

实验项目名称	内容提要	所用主要设备或实验环境	实验学时	每组人数	实验属性（基本/综合/设计/研究创新）	开出要求（必做/选做）
基本程序设计	环境安装，基本程序设计，选择和循环，调试	Python3，Jupyter notebook	2	1	基本	必做
有序类型和函数	列表、元组、字典集合和函数	Python3，Jupyter notebook	2	1	基本	必做
正则和 OOP	字符串，正则表达式和 OOP 编程	Python3，Jupyter notebook	2	1	综合	必做
爬虫和 Web 服务	爬虫和 Web 服务	Python3，Jupyter notebook	2	3-5	综合	必做
综合实验	复杂特定工程问题	Python3，Pycharm 或其他自选工作	2（课内）	3-5	综合	必做

（二）教学方法

整个课程都是讲授 Python 语言的编程，因此，动手实践必须有非常高的比重。目前课程安排的是 50%的时间作为实践环节，但要熟练掌握一门语言，还是不够的。在教学过程中，必须安排一定量的课后编程实践，同时对实验报告的检查也必须跟上。课程考核可以采用大作业或上机考试的方式，教师可以根据实际情况进行选择。

六、课程考核方式与评价标准

（一）考核方式

表7 最终成绩评定依据

考核方式或途径	考核要求	考核权重	评估级别
课堂表现	出勤	10%	随机点名，未被点到 1 次扣 1 分
	课堂提问、讨论		加分制，每一次加 1 分，最高 10 分
平时作业和实验报告	4-5 次	30%	根据程序质量、文本质量进行 5 级给分，按 A+=100，A=90，B+=85，B=80，C+=75，C=70，D+=65，D=60，E=50，未交则以 0 的等级给分
期末考试	综合实验报告+答辩	60%	根据报告和答辩的质量给分

表8 实验报告和作业的评估依据

项目	比例	优秀（A）	良好（B）	中等（C）	及格（D）	不及格（E）
程序质量	50%	代码清晰，注释规范，结果达到实验目标，体现创新性意识	代码较清晰，有重要注释，结果达到实验目标	代码格式比较规范，能够达到实验目标，个别条件欠考虑	基本达到实验目标	未达到实验目标
文本质量	20%	文本格式规范，条理清晰，逻辑性强	格式有不规范的地方，方案有条理和一定的逻辑性	字体、段落等格式不规范处较多，有些内容描述不清楚	内容的流水账	缺少必要内容
方案质量	30%	检索资料充分，对问题和已有的解决方法有合理的分析，方法体现创新意识，结果分析合理	检索资料充分，对问题和已有的解决方法有分析，方法是已有方法的改进	能找到参考资料，在此基础上修改给出解决方案，分析合理性不足	达到目标的方案	方案有严重错误

表9 综合实验报告的评估依据

项目	比例	优秀（A）	良好（B）	中等（C）	及格（D）	不及格（E）
程序和方案质量	50%	代码清晰，注释规范，结果达到实验目标，检索资料充分，对问题和已有的解决	代码较清晰，有重要注释，检索资料充分，对问题和已有的解决方法有分析，结	代码格式比较规范，能找到参考资料，在此基础上修改给出解决方案，分析合理	基本能够给出方案和结果，实验目标，资料、	方案完全不可行

		方法有合理的分析，方法体现创新意识，结果分析合理	果分析有合理性	性不足	分析等一般	
文本质量	10%	文本格式规范，条理清晰，逻辑性强	格式有不规范的地方，方案有条理和一定的逻辑性	字体、段落等格式不规范处较多，有些内容描述不清楚	内容的流水账	缺少必要内容
答辩质量	30%	对工作熟悉，阐述清晰，逻辑性强，正确回答问题，体现创新意识和能力	对工作熟悉程度较好，方案阐述清晰，逻辑性较好，回答问题正确	工作熟悉程度一般，回答存在瑕疵，但能够描述清楚所做的工作	对工作能够复述，回答问题的内容仅限于做的内容，对某些内容表述不清楚	对工作不了解，无法回答问题
其他	10%	着装正式，仪态大方自如，语言有感染力，时间把握正好	着装得体，仪态较大方，语言流畅，时间把握较好	着装尚可，仪态有点拘束，语言平庸，对时间把握不到位	着装随意，仪态紧张，语言干涩，没有时间概念	有重大言论错误
协作性	乘数：0.5-1，表明协作性从低到高，上述四项得分之和*协作性*小组人数为小组总分，各成员根据参与度，在小组内分配总分					

七、推荐教材及参考资料

（一）教材：

[1] ［美］梁勇（Liang y. d.）编著，李娜译：《Python 语言程序设计》，机械工业出版社，2015 年 4 月出版。

（二）参考书：

[1] (美)Mark Summerfield 编著，王弘博，孙传庆译，《Python 3 程序开发指南（第 2 版）》，机械工业出版社，2011 年 10 月出版。

[2] 华校专编著，《Python 大战机器学习：数据科学家的第一个小目标》，人民邮电出版社，2015 年 2 月出版。

[3] Eric Matthes, 袁国忠译，《Python 编程：从入门到实践》，人民邮电出版社，2016 年 7 月出版

（三）网络资源：

[1] http://www.codecombat.com
[2] http://www.runoob.com/python/python-tutorial.html

执笔：沈炜　　　　审稿：张涵翠　　　　　　审定：信息学院教学委员会

制（修）订时间：2021 年 5 月 20 日

Syllabus of Comprehensive Research and Training of Intelligent Software

Course Name/Title：Comprehensive Research and Training of Intelligent Software
Course Code： 62712
Course Type： (Specialized Course)(Compulsory Course)
Total Teaching Hours：60 (Classroom Hours: 0 Laboratory Hours or Tutorial Hours 60)
Course Credit： 3

I Course Introduction

Comprehensive research and training of intelligent software is a professional compulsory course for computer science and technology majors. It is a basic course that cultivates students' basic skills in intelligent software technology, and carries out follow-up professional courses and graduation design. Through the study of this course, students are required to deepen their understanding of the basic concepts, basic theories and analysis, and design methods of intelligent software technology, and be able to independently complete the analysis, design and implementation of intelligent systems, and summarize and write design reports.

II Course Objective

Course objective 1: Through the analysis of the research results of intelligent software in our country, we can stimulate students' hard-working innovation and craftsmanship; recognize the bottleneck of our country's intelligent software development technology, and guide students to establish a national sentiment and national mission for the country and the people; Practice project cases, cultivate students' perspectives on things dialectically, analyze problems from different angles, and cultivate dialectical system thinking.

Course objective 2: Understand the basic concepts, models, analysis and design methods of intelligent software, master intelligent software technology and related standards, and have intelligent software design capabilities. Recognize our country's bottlenecks in intelligent software technology, enhance the sense of responsibility, and promote patriotism.

Course objective 3: Aiming at problems in specific application areas, analyze actual application requirements, design plans, and use methods such as performance analysis and cost-benefit evaluation to demonstrate the feasibility and rationality of the design plans. Improve the ability to independently analyze and solve practical problems.

Course objective 4: Understand intelligent software application scenarios and technologies, and master the application of intelligent software technologies in the industry. Analyze typical cases to cultivate students' sense of service and social responsibility, and enhance their innovation in the process of exercising practical hands-on ability.

Course objective 5: Take the group as a unit, jointly analyze problems, design schemes and solve problems in practice, write design reports and feed them back to the design of complex projects; with guide of practical application problems, train students to simplify think about complex problems, cultivate practical skills and teamwork spirit.

Course objective 6: Understand the advanced technology of intelligent software, have the literacy of combining software engineering technology and machine learning method to carry out intelligent software application innovation; Understand the full cycle process of intelligent software development, take into account the constraints in the development process, and have a software project management awareness.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation requirement 3: Design/Develop Solutions	3-2 Consider economic, social, health, safety, legal, cultural and environmental constraints, and demonstrate the feasibility and innovation of the design plan	Course Objectives 6
Graduation requirement 6: Engineering and Society	6-1 Familiar with computer-related technical standards, intellectual property rights, information security regulations, industrial policies, laws and regulations, and understand the responsibilities that should be undertaken	Course Objectives 2
Graduation requirement 8: professional norms	8-3 Understand the social responsibility of computer engineers for the safety, health and well-being of the public, and environmental protection, and be able to consciously abide by and perform corresponding responsibilities in the practice of computer engineering	Course Objective 1、4
Graduation requirement 9: Individuals and teams	9-1 Able to work in cooperation with members of other disciplines, capable of fulfilling the roles of individuals and team members and assuming corresponding responsibilities	Course Objective 5

Graduation requirement 11: project management	11-2 Use cost-benefit evaluation methods to conduct cost-benefit analysis of engineering schemes	Course Objectives 3

(Note: Basic courses and specialized courses must correlate with the graduation requirements as specified in the Program outline. The correlated graduation requirement index point must be put before the descriptive phrases or sentences. General courses are exempted from this rule.)

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Design and implementation of machine learning system (1) Introduce how to extract machine learning problems from actual problems, and choose suitable machine learning models for modeling, learning and optimization (2) Introduce some classic instances of machine learning systems	Master the basic theories and algorithms of machine learning, so as to have the ability to design machine learning systems Be able to correctly use machine learning methods in actual project research and track cutting-edge machine learning algorithms, ideas, applications, etc.	20	experiment	Course Objectives 1、2、3、4、5
2	Design and implementation of intelligent components based on cloud platform (1) Introduction of mainstream artificial intelligence cloud platforms; (2) Design of intelligent component based on cloud platform; (3) Intelligent technology based on cloud platform.	(1) Master the design methods and technologies of machine learning systems based on cloud platforms; (2) Ability to independently build a machine learning system for application scenarios with the help of cloud platforms; (3) Understand the advantages and disadvantages of mainstream artificial intelligence cloud platforms, and enhance the sense of innovation and craftsmanship.	20	experiment	Course Objectives 1、2、3、4、5
3	Comprehensive development and realization of intelligent software system (1) Use database technology and intelligent software technology to design and implement	Be able to use the knowledge related to intelligent software to independently analyze and solve problems, and complete the analysis, design and development of a certain scale of intelligent systems; (2) The course design report should be completed at the end of the course design process, able to explain the details of system analysis and design	20	experiment	Course Objectives 1、2、3、4、5、6

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	intelligent software systems; (2) Summarize the design content, sort out relevant experimental data, and write a design report.	implementation, and submit and display the corresponding system (3) Train and improve the ability to discover, analyze, and solve problems in application practice, and cultivate the spirit of craftsmen who work hard.			

V Period Distribution and Teaching Modes

V.1 Period Distribution

Name	Contents	Environments	Period	Group size	Attributes（Basic /Comprehensive/ Design/ Creative Study）	Requirements(Compulsory /Optional)
Course design	Design and implementation of machine learning system	computer	20	1	Comprehensive	Compulsory
Course design	Design and implementation of intelligent components based on cloud platform	computer	20	1	Comprehensive	Compulsory
Course design	Comprehensive development and realization of intelligent software system	computer	20	1	Comprehensive	Compulsory
Total			60			

V.2 Teaching Mode

This experiment is mainly based on laboratory study. The teacher needs to explain to the students the nature, tasks, requirements, course arrangement and progress, contents to be assessed, experimental rules and laboratory safety system of the course.

Each student has a computer, and the students must complete the designated experimental project. In order to cultivate students' practical operation ability and teamwork spirit, students can ask two people to complete the topic selection, preview, experiment and experiment report writing.

VI Assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives

Classroom performance		Attendance Other performances (answering questions, discussion)	20%	Course Objectives 1、2、3、4
Operational performance		Solve problem and complete the project on time.	30%	Course Objectives 1、2、3
Final grade	Final report	Write an experimental report based on the project.	50%	Course Objectives 2、3、4、5、6
	Oral presentation	Prepare a ppt based on system design, implementation and solutions, and do a presentation in English.		Course Objectives 2、3、4、5、6
	Course Project	Provide an intelligent software system with good performance		Course Objectives 2、3、4、5、6

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

[2] Lei Ming et al. Machine Learning and Application. Tsinghua University Press, 2018.

[3] Hadelin de Ponteves. AI Crash Course. China Machine Press, 2020

Written by：Wang Lin **Reviewed by: Dongming Xiang**

Date：2021.05.20

Syllabus of Embedded System Principles and Design

Course Name/Title：Computer Network **Course code：62933**

Course Type： Specialized Course, Optional Course

Total Teaching Hours：32 (Classroom Hours: 32)

Course Credit：2

I Course Introduction

This course introduces the various building blocks and underlying scientific and engineering principles behind embedded real-time systems. The course covers the integrated hardware and software aspects of embedded processor architectures, along with advanced topics such as real-time, resource/device and memory management.

Students can expect to learn how to program with the embedded architecture that is ubiquitous in smartphones, portable gaming devices, robots, etc. Students will then go on to learn and apply real-time principles that are used to drive critical embedded systems like automobiles, avionics, medical equipment, etc. Topics covered include embedded architectures (building up to modern 16/32/64-bit embedded processors); interaction with devices (buses, memory architectures, memory management, device drivers); concurrency (software and hardware interrupts, timers); real-time principles (multi-tasking, scheduling, synchronization); implementation trade-offs, profiling and code optimization (for performance and memory); embedded software (exception handling, loading, mode-switching, programming embedded systems).

Although this is a theoretical course, we still set up enough extracurricular hands-on practice. In addition, we set up an independent practical, hands-on course *embedded system project practice,* which systematically helps students acquire skills in the design/implementation/debugging of core embedded real-time functionality. Welcome to select the course at same semester.

II Course Objective

Course Objective 1: With the characteristics of highly customized and real-time embedded system as the breakthrough point, cultivating interdisciplinary engineering and technical talents in multiple dimensions such as ideological and political, time and space in an all-round way. During theories teaching and hand-on practice, the ideological and political links of "stimulating interest", "cultivating self-confidence" and "tapping potential" are embedded.

Course Objective 2: Master the various building blocks and underlying scientific and engineering principles behind embedded real-time systems, and consciously apply them to engineering practice.

Course Objective 3: Learn how to program with the embedded architecture, including exception handling, loading, mode-switching. Implementation trade-offs, profiling and code optimization (for performance and memory).

Course Objective 4: Describe embedded architecture; Describe and evaluate the interaction with different devices; Concurrent (software and hardware interrupt, timer) processing; Applying real-time principle (multi task, scheduling, and synchronization) to the key equipment. Explains how to make engineering trade-offs of power use and performance; Investigates practical issues concerning software reliability, aging, rejuvenation, security, and power management.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
GR1 Knowledge	1-4 Be able to use engineering basic knowledge, professional knowledge and mathematical model for scheme comparison and analysis of computer complex engineering problems.	Course Objective 1,2
GR4 Investigation	4-1 Be able to research and verify the key algorithms and modules related to computer science	Course Objective 4
GR5 Modern tool usage	5-2 Be able to use integrated development tools, open source and third-party resources to develop, debug and test computer systems, and understand their limitations	Course Objective 3

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Unit 1: Introduction** (1) Basic concepts of embedded system (2) Hardware unit of embedded system (3) Software design of embedded system (4) Real time system	(1) Define the embedded system. (2) List various building blocks and underlying scientific and engineering principles behind embedded real-time systems. (3) Define a real-time system. (4) Give further examples, such as large and complex systems which need strict reliability and real-time response, such as civil aviation	4	Lecture	1,2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		passenger aircraft, high-speed rail and spacecraft. It shows China's great achievements in the great power heavy equipment, stimulates patriotic enthusiasm and "cultivates self-confidence".			
2	**Unit 2: ISA & CPU** (1) Computer architecture (2) msp-exp430 microprocessor architecture (3) Msp-exp430 instruction system.	(1)Describe the architecture of embedded processor and its hardware and software cooperation according to the evolution of processor type. (2) Describe the relationship between instruction set, architecture and CPU. (3) Mater the programming technology of embedded system bottom instruction system. (4) Analyze why the chip field can be "strangle", grasp the importance of core technology, "tap potential", and tell the story of brave breakthrough.	6	Lecture group discussion	2 3,4
3	**Unit 3: embedded system based on bus** (1) Input / output programming (2) Memory and storage system mechanism (3) Development and debugging	(1) The data input and output can be understood by connecting all parts of the system with bus (2) The concept of concurrency, processor system level scheduling, interrupt handling (3) Storage system, and can understand the software and hardware implementation process. (4) One belt, one road linking China to the world, plays a role of the bus. "Stimulating interest" refers to how to rely on the existing bilateral and multilateral mechanisms between China and relevant countries to build a community of interests, destiny and responsibility with mutual political trust, economic integration and cultural tolerance.	6	Lecture group discussion	2 3,4
4	**Unit 4: The embedded software architecture** (1)Embedded software architecture (2)Real-time operating system.	(1) Embedded software architecture, the impact on real-time processing. (2) In the real-time operating system environment, further improve the performance of the system.	6	Lecture group discussion	2 3,4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
5	**Unit 5: Embedded operating system services** (1) Message queue, mailbox and pipeline, timer function, event (2) RTOS storage management, interrupt program in RTOS environment.	(1) Understand the communication, timing service, event management, storage management, interrupt rules and other system services provided by RTOS (2) To improve students' ability of embedded design in RTOS environment.	6	Lecture	2 3,4
6	**Unit 6: Multi processor system** Software implementation process modeling, multiprocessor system modeling	Understand the main concepts and methods of program modeling before software implementation, and understand the modeling process of multiprocessor system.	4	Lecture	3,4

V Period Distribution and Teaching Modes

6. Theory Lecture period distribution

Teaching mode / hours / contents	lecture	exercise	Discussion	remark	S total
Unit 1: Introduction	4				4
Unit 2: ISA & CPU	6				6
Unit 3: Embedded system based on bus	6				6
Unit 4: The embedded software	6				6
Unit 5: Embedded operating system	6				6
Unit 6: Multi processor system	4				4
Total	**32**				**32**

1. Teaching methods

In this course, teacher should lecture in English at around 70 percentages. It is better to use case teaching. Teacher should adopt “question –result” mode to give students strong impression.

2. Teaching Ways

Teacher should use multimedia classroom to teach this course. The amount of time should be left to discussion for all students. The discussion class can be grouped. One group may quest; the other may answer the question.

VI Assessment

(Assessment content and methods must correlate with course objectives.)

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
the rates of attendance	Evaluate according to class attendance, participation and practice.	10%	Course Objectives 1
homework	3-4 times, Evaluate according to the hand in, accuracy, writing quality of homework.	30%	Course Objectives 2、3、4
Final Exam (close-book)	Assess the correctness, the steps and the ideas of solving the questions.	60%	Course Objectives 2、3、4

Note: 1. Assessment methods or approaches mainly include classroom performance, conventional assignments, unit tests, mid-term exam, final exam, big assignments, course paper, project design and works, etc.

2. Assessment requirements include frequencies of assignments, assessment methods (open-book, close-book), and project design requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

1. Textbook

[1] 贾宇波，夏劲松，李俊松.《嵌入式系统原理与设计》(自编教材). 浙江理工大学，2021

2. References

[1] 王宜怀，《嵌入式技术基础与实践》(第 5 版)，清华大学出版社，2019.4。

[2] 卡莫尔，《嵌入式系统:体系结构、编程与设计》(第 2 版)，清华大学出版社，2010。

[3] 西蒙，《嵌入式系统软件教程》，机械工业出版社 2005.。

Written by：Jinsong Xia **Reviewed by: Junsong Li**

Date：2021.05.20

Syllabus of Trust and Reputation for Electronic Service-oriented Environments

Course Name/Title： Trust and Reputation for Electronic Service-oriented Environments

Course code：62955

Course Type： (Specialized Course) (Optional Course)

Total Teaching Hours： 32 (Classroom Hours: 30, Exam Hours: 2)

Course Credit：2

I Course Introduction

With the development of modern service industry, trust, as an important reference material in modern society, has been paid attention to by many organizations and individuals. The modernization of trust service has become an urgent problem to be solved. In modern society, the development of the Internet has made a great progress, which has had a profound impact on the social development and rapidly permeate into a variety of industries. The networking of trust industry, becomes an important way and means of modernization of trust service, trust service also may become a part of modern service industry, which can be easily found、searched and used. In the field of network economy environment and information technology, as a knowledge system, trust and reputation is causing extensive concern.

<Trust and Reputation for Electronic Service-oriented Environments> is an optional specialized course. This course provides an opportunity for advanced undergraduates of Computer Science and Technology major and Information and Computing Science major to fully understand the concepts and related topics in trust and reputation for electronic service-oriented environments. Through the study of this course, students can have a comprehensive understanding of the basic concepts and implementation mechanisms of trust, manage to understand current typical trust and reputation model, and can propose some design ideas of some trust and reputation model.

By introducing the characteristics of trust, Sesame Credit and Huabei applications, we should tell students to be honest and establish a correct outlook on life and values. By telling the story of the first transaction of Alipay which marks the emergence of an era of trust, we need to cultivate students' consciousness of innovation. We need to stimulate students' sense of pride and mission by introducing the world's leading position in customer-based innovation achieved by Chinese electronic service companies, including their amazing capabilities in the areas of user profiles、targeted marketing and promotions.

II Course Objective

Course objective 1: By introducing the characteristics of trust, Sesame Credit and Huabei applications, we should tell students to be honest and establish a correct outlook on life and values. By telling the story of the first transaction of Alipay which marks the emergence of an era of trust, we need to cultivate students' consciousness of innovation. We need to stimulate students' sense of pride and mission by introducing the world's leading position in customer-based innovation achieved by Chinese electronic service companies, including their amazing capabilities in the areas of user profiles、targeted marketing and promotions.

Course objective 2: master the core concepts of trust、trust relationship、trustworthiness、reputation、trust ontology、reputation、trust and reputation modeling. Master the characteristics of trust and reputation: dynamic characteristic and dependency on context and time.

Course objective 3: based on real examples, using the principles of mathematics, natural science and engineering science as well as the methods and technologies of computer science, analyze and evaluate all kinds of trustworthiness system and reputation system from two different perspectives of business and technology, and can verify and evaluate these systems therefore provide some specific indicators and effective conclusions.

Course objective 4: To have the ability of active learning. For open questions, manage to utilize information and bibliographies tools to learn knowledge autonomously and write relevant course papers. To cultivate students' ability of collecting information and innovation consciousness.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1	Able to use the principles of mathematics, natural science and engineering science as well as the methods and techniques of computer science major to analyze and evaluate solutions of complex engineering problems in specific computer applications, and to provide specific indicators and effective conclusions (verification and evaluation).	Course Objective 1,2,3
Graduation Requirement 2	Have the ability to learn actively, manage to use information and bibliographic tools, learn knowledge autonomously.	Course Objective 4

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**1 Trust and security in service-oriented environments** (1) Why trust is important to e-business、trust and security、trust in security context、trust in business context; (2) service-oriented environment、 agents in service-oriented environments、infrastructure in service-oriented environments、technology in service-oriented environments 、 trust in service-oriented environments	(1) Explain why trust is important to e-business; (2) State the difference between trust and security; (3) Define what is a service-oriented network environment; (4) Illustrate issues in service-oriented environments; (5) Illustrate the components in service-oriented environments; (6) Introduce the technology in service-oriented environments; (7) Illustrate the infrastructure in service-oriented environments; (8) Define what are agents, buyers, sellers, customers, users, website, products and services; (9) Define what is Quality of Service (QoS) and quality of products (QoP) ? (10) The service-oriented architecture web service as a technology.	2	in-class teaching、Online resource platform、in-class test	Course Objective 2
2	**2 Trust concepts and trust model** (1) trust Environments、 trust definitions in literature、advanced trust concepts; (2) trust relationships、trust relationship diagram、trust attributes and methods、initiation of the relationship、the trust model;	(1) States trust may exist in virtual environment and physical environment; (2) advanced trust concepts, illustrate trust relationships and trust relationship diagram; (3) Introduce the initiation of the relationship, briefly illustrates trust model and pave the way for the subsequent chapters; (4) By telling the story of the first transaction of Alipay which marks the emergence of an era of trust, we need to cultivate students' consciousness of innovation.	4	in-class teaching、Online resource platform、in-class test	Course Objective 2, 3
3	**3 Trustworthiness** (1) Trustworthiness in literature、advanced trustworthiness definition; (2) Seven levels of the	(1) States that trust and trustworthiness are two distinct concepts; (2) Illustrates seven levels of the trustworthiness; (3) Illustrates semantics representation	2	in-class teaching、Online resource platform、	Course objective 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	trustworthiness; (3) Semantics representation and postulates for trustworthiness levels; (4) Trustworthiness measure and prediction 、 challenges in trustworthiness measure and prediction.	and postulates for trustworthiness levels; (4) Illustrates the definition of trustworthiness measure and prediction, demonstrate the challenges in trustworthiness measure and prediction.		in-class test	
4	**4 Trust Ontology for Service-Oriented Environment** (1) Ontology、hierarchy of trust concepts、hierarchy of agents, service and product concepts; (2) Hierarchy of context and association with quality assessment criteria; (3) Agent trust ontology、service trust ontology、product trust ontology; (4) Trust databases.	(1) Introduces the concept of ontology、hierarchy of trust concepts、hierarchy of agents, service and product concepts; (2) Illustrates the hierarchy of context and association with quality assessment criteria; (3) States agent trust ontology、service trust ontology、product trust ontology; (4) Introduce trust databases.	4	in-class teaching、Online resource platform、in-class test	Course objective 2
5	**5 The Fuzzy and Dynamic Nature of Trust** (1) Fuzzy and dynamic characteristics of trust; (2) Endogenous and exogenous characteristics of agents; (3) Reasoning the fuzziness and dynamism; (4) Managing the Fuzziness of Trust.	(1) States the fuzzy and dynamic characteristics of trust; (2) States the endogenous and exogenous characteristics of agents; (3) Illustrates the reasoning of fuzziness and dynamism; (4) Illustrates how to manage the fuzziness of trust.	2	in-class teaching、Online resource platform、in-class test	Course objective 2
6	**6 Trustworthiness Systems** (1) Amazon's trustworthiness systems; (2) Yahoo's trustworthiness systems; (3) Epinions.com's trustworthiness systems; (4) eBay.com's trustworthiness	(1) The trustworthiness systems; (2) Trustworthiness measure for rating agents,sellers,reviews,reviewers and products; (3) The weakness of existing measures. (4) We need to stimulate students' sense	2	in-class teaching、Online resource platform、in-class test	Course objective 3, 4

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	systems; (5) BizRate.com's trustworthiness systems; (6) CNet.com's trustworthiness systems; (7) Summary of trustworthiness systems.	of pride and mission by introducing the world's leading position in customer-based innovation achieved by Chinese electronic service companies, including their amazing capabilities in the areas of user profiles、targeted marketing and promotions.			
7	**7 Reputation Concepts and the Reputation Model** (1) Reputation in literature、advanced reputation concepts; (2) Reputation relationship、recommendation trust relationship、third-party trust relationship、 reputation query relationship; (3) Trustworthiness of third-party recommendation agents、trustworthiness of the opinion; (4) Reputation model and reputation relationship diagrams.	(1) Introduce reputation concept in literature and advanced reputation concepts; (2) Illustrate reputation relationship、recommendation trust relationship、third-party trust relationship and reputation query relationship; (3) Illustrate trustworthiness of third-party recommendation agents and trustworthiness of the opinion; (4) Introduce reputation model and reputation relationship diagrams.	4	in-class teaching、Online resource platform、in-class test	Course objective 2, 3
8	**8 Reputation Ontology** (1) Reputation ontology、basic and advanced reputation ontology; (2) Trustworthiness of opinion ontology; (3) Ontology for reputation of an agent、ontology for reputation of service、ontology for reputation of a product;	(1) Presents reputation ontology、basic and advanced reputation ontology; (2) Presents trustworthiness of opinion ontology; (3) Introduces ontology for reputation of an agent、ontology for reputation of service and ontology for reputation of a product; (4) Illustrates reputation databases; (5) Denotes levels of reputation	2	in-class teaching、online resource platform、in-class test	Course objective 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(4) Reputation databases; (5) Levels of reputation measurement; (6) The fuzzy nature of reputation、the dynamic nature of reputation.	measurement; (6) States the fuzzy nature of reputation and the dynamic nature of reputation.			
9	**9 Reputation Systems** (1) Trust and reputation systems versus recommendation systems; (2) BizRate.com; (3) Elance.com; (4) Alibris.com; (5) MoneyControl.com; (6) Yahoo.com; (7) Epinions.com; (8) eBay.com; (9) CNET.com; (10) MovieLens recommendation system.	(1) States the differences between reputation and recommendation systems; (2) Illustrates the reputation system of BizRate.com; (3) Illustrates the reputation system of Elance.com; (4) Illustrates the reputation system of Alibris.com; (5) Illustrates the reputation system of MoneyControl.com; (6) Illustrates the reputation system of Yahoo.com; (7) Illustrates the reputation system of Epinions.com; (8) Illustrates the reputation system of eBay.com; (9) Illustrates the reputation system of CNET.com; (10) Illustrates MovieLens recommendation system.	2	in-class teaching、Online resource platform、in-class test	Course objective 3,4
10	**10 Trust and Reputation Modeling** (1) Significance of pictorial modeling; (2) Notation systems;	(1) Illustrates the significance of pictorial modeling; (2) Introduces the notation systems; (3) Illustrates the trust relationship diagrams;	4	in-class teaching、Online resource platform、in-class	Course objective 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	(3) Trust relationship diagrams; (4) Trust case diagrams; (5) Trust class diagrams; (6) Trust transition diagrams;	(4) Illustrates the trust case diagrams; (5) Illustrates the trust class diagrams; (6) Illustrates the trust transition diagrams;		test	
11	**11 The Vision of Trust and Reputation Technology** (1) Business intelligence; (2) Traditional IT and new-age digital ecosystems and technology; (3) Trust and reputation-an example of digital ecosystem and technology; (4) Future research and development.	(1) Introduce the concept of business intelligence; (2) Presents traditional IT and new-age digital ecosystems and technology; (3) States an example of Trust and reputation - a digital ecosystem and technology; (4) Presents future research and development.	2	in-class teaching、Online resource platform、in-class test	Course objective 3, 4

V Period Distribution and Teaching Modes

Period distribution of theoretically teaching content

Teaching method / Teaching period / Course content	Theoretical Teaching	Exam class	Notes	Summary
Chapter 1 Trust and security in service-oriented environments	2			2
Chapter 2 Trust concepts and trust model	4			4
Chapter 3 Trustworthiness	2			2
Chapter 4 Trust Ontology for Service-Oriented Environment	4			4
Chapter 5 The Fuzzy and Dynamic Nature of Trust	2			2
Chapter 6 Trustworthiness Systems	2			2

Chapter 7 Reputation Concepts and the Reputation Model	4			4
Chapter 8 Reputation Ontology	2			2
Chapter 9 Reputation Systems	2			2
Chapter 10 Trust and Reputation Modeling	4			4
Chapter 11 The Vision of Trust and Reputation Technology	2			2
Exam Class		2		
Summary	30	2		32

In theoretical teaching, in-class teaching is the main method. At the same time, theories are taught through classroom discussion and questions asking, which deepens the understanding and consolidation of theoretical knowledge、 important and difficult points. By assigning a certain amount of homework, which helps students to further consolidate the teaching content in classroom.

In order to make students to learn initiatively, we provide corresponding online learning platform for students to learn. Some votes、questionnaires and discussion questions are provided to students to accomplish on the online platform. We understand the learning status of students in time, then we can answer students' questions immediately and adjust the teaching methods accordingly.

VI Assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Performance in class	Evaluations based on attendance and in-class tests	10	Course objective 1, 2
Casual homework	Evaluations based on the hand-in frequencies、accuracy and writing quality of homework.	20	Course objective 1, 2 , 3
Online learning	Evaluations based on watching frequencies of teaching PPTs、participation in votes、 answers to questionnaires.	10	Course objective 1
Final exam	Open-book exam, score according to the scoring criteria	60	Course objective 1, 2

VII Textbooks and References

1. Textbook:

[1] Elizabeth Chang et al. <Trust and Reputation for Service-Oriented Environments: Technologies for Building Business Intelligence and Consumer Confidence>. John Wiley & Sons, Ltd, 2006.

2. Reference book:

[1] Zaki Malik. <Trust Management for Service-Oriented Environments>. Springer. 2009.

3. Online resources:

[1]https://www.mosoteach.cn/web/index.php?c=interaction&m=index&clazz_course_id=5470C90C-414A-4CC4-8B61-1D988E191A5E

Written by：Yu Zhang **Reviewed by: Jing Bian**

Date：2021.05.20

Syllabus of Mobile Application Development

Course Name/Title： Mobile Application Development
Course code： 62578
Course category and nature： Specialized Course、Optional Course
Total Teaching Hours： 32 hours (theory: 24 hours, Laboratory: 8 hours)
Course Credit: 2
Prerequisite course: Java programming
Object-oriented: Computer Science and Technology
Department of Courses (room): Department of Computer Science and Technology

I Course Introduction

With the popularization of mobile terminal applications and the popularization of the Android system, there has been a large demand for Android application software. This course aims to cultivate students’ small and medium-sized Android application software development capabilities through comprehensive learning of Android application development technology. The social transportation urgently needs talents. This course focuses on the cultivation of students' engineering ability, innovative spirit and career development, and aims to enable students to master Android development technology through comprehensive learning of Android application development technology, and cultivate students' ability to develop small and medium-sized Android application software. This course organizes a discussion in the classroom on the comparative analysis between the domestic Harmony OS and the Android system, and strengthens students' sense of mission and urgency for independent research and development of core technologies. Cultivate students' good logical thinking ability through specific project development, develop good programming habits, cultivate the ability of self-study, reading materials and use materials, encourage students to analyze and solve problems independently in the software design process in a team, so as to equip students with theoretical knowledge And practical skills are developed together.

II Course Objective

"Mobile Application Development" is a professional elective course for undergraduates majoring in computer science and technology. It is a teaching practice course that links mobile application development and professional technical knowledge. Its main purpose is to enable students to fully understand and master the basic concepts, basic processes and basic methods of mobile application development through the study of this course, and to develop mobile platform applications based on Java language and Android mobile phone operating system; master preliminary experiments Skills, cultivate students' scientific thinking ability, scientific induction ability, analytical calculation ability, experimental research ability and problem-solving ability, and prepare the necessary knowledge for the follow-up courses.

Course Objective I: Master the design philosophy of good user experience, Android programming ideas and performance optimization principles. Understand the historical development of the Harmony OS and the Android system, clarify the importance of developing independent research and development of core technology, and promote patriotism.

Course Objective II: Master the related technologies of Android development, complete the basic experiments required by this course, and enhance the ability to analyze and solve problems. Understand the application of mobile applications in the fields of mobile Internet and Internet of Things, and cultivate students' innovative ways of thinking.

Course Objective III: Students can complete the conversion of real business logic to code, can analyze and solve technical problems independently, and enhance the cognitive ability of independent learning and lifelong learning. Cultivate students' practical ability and innovative consciousness.

Course Objective IV: Be able to write technical documents in accordance with the specifications, cultivate teamwork and communication skills, and be able to cooperate with team members to complete medium-sized mobile application development tasks.

III Correspondence between curriculum teaching objectives and graduation requirements

Graduation requirements	Index points for graduation requirements	Course objectives
Graduation requirements 3	3-1 Master the full-cycle development process, basic design/development methods and technologies of computer systems and software products, and understand various factors that affect design goals and technical solutions.	Course objectives 1,2,3
	3-2 Considering economic, social, health, safety, legal, cultural, and environmental constraints and other practical constraints to demonstrate the feasibility and innovation of the design plan.	
Graduation requirements 5	5-2 Be able to use integrated development tools, open source and third-party resources to develop, debug and test computer systems, and understand their limitations.	Course objectives 2,3
Graduation requirements 9	9-1 Be able to work in cooperation with members of other disciplines, capable of fulfilling the roles of individuals and team members and assuming corresponding responsibilities.	Course objective 3
Graduation requirements 10	10-1 Be able to make effective written and oral presentations on complex engineering issues, and be able to communicate effectively with others, including writing reports and design documents, making statements, expressing clearly or responding to instructions.	Course objective 4
Graduation requirements 12	12-1 Be able to recognize the necessity of continuous exploration and learning, have the awareness of autonomous learning and lifelong learning, master the methods of autonomous learning, and understand the ability and ways to expand knowledge.	Course objective 5

IV The relationship between teaching content and course objectives

(1) The relationship between theoretical teaching content and course objectives (24 hours)

The relationship between theoretical teaching content and course objectives

Serial number	Teaching content	Teaching requirements	Class hours	The way of teaching	The corresponding course target
1	**Chapter 1 Overview of Mobile Application Development** (1) summary of Android (2) the Android program (3)Android program debugging	(10)Explore the history and architecture of Android, compare and analyze Harmony OS with Android, understand the importance of independent research and development of core technologies, and promote patriotism; (2)Explain JDK installation and configuration methods, Android development environment download and install and Android simulator establishment and configuration in detail; (3) Explain the steps to create an Android application using Android Studio and run it on a phone or emulator; Demonstrate the use of LogCat, Android breakpoint debugging process.	4	Classroom lectures, online courses, class discussions	Course Objective 1
2	**Chapter 2 Android Views and Components**Activity (7) Overview (8) Activity layout (9) Android basic components (10)Android advanced components (11)Android adapter	(6) Explain the implicit jump and explicit jump method between Activity; (7) Explain the principle and usage of data transfer between Bundle and Activity; (8) Explain the data saving and recovery mechanism of Activity exit; (9) Demonstrate the methods and application scenarios of absolute layout, relative layout, linear layout and restrictive layout (10)List the usage methods of basic components such as TextView, Button, EditText, and ProgressBar. (11)List the use of intermediate	4	Classroom lectures, online courses, class discussions	Course objectives 1, 2, 3

Serial number	Teaching content	Teaching requirements	Class hours	The way of teaching	The corresponding course target
3		components such as CheckBox, Switch, RadioButton, ImageViewer. (12)List the use of Spinner and ListView adapters.			
3	**Chapter 3 Fragment Components** (1) How to use Fragment controls; (2)How to use the ViewPager control.	(12)Explain principles and static and dynamic loading methods of Fragment controls in detail; (13)Explain the methods of data interaction between Fragment and Activity, Fragment in detail; (14)Explain the principle of ViewPager control loading Fragment and adapter usage; (15)The method of data interaction between RadioGroup and ViewPager.	2	Classroom lectures, online courses, classroom discussions	Course objectives 1, 2, 3
4	**Chapter 4 Broadcast** (1) How Android broadcast works (2) Custom broadcast (3)Orderly broadcast	(9) Explain the method of broadcasting to ensure security; (10)Explain system broadcasting and development of mobile phone battery monitoring program in detail; (11)Explain the principles of broadcast reception and transmission in detail; (12)Explain the sending and receiving of custom broadcasts in detail; (13)Explain the receiving process of orderly broadcast in detail; (14)By discussing whether the Harmony OS can replace the Android system, it reflects the importance of domestic technology, mobilizes students' enthusiasm for independent innovation, and stimulates students' sense of mission.	4	Classroom lectures, online courses, classroom discussions	Course objectives 1, 2, 3

Serial number	Teaching content	Teaching requirements	Class hours	The way of teaching	The corresponding course target
5	**Chapter 5 Services** (1) Android system service (2) Android custom service (3) Multithreading (4) Multimedia services Data storage service	(3) Explain the classification and function of Android services; (4) Explain the application occasions and functions of Android system services and custom services; (5) Explain the use of binding to realize the data interaction between the custom service and the Activity; (6) Explain description of the asynchronous task start process of IntentService and AsyncTask in detail. (7) Demonstrate of the launch of the album service and permission access service (8) Explain how to use audio and video service interfaces (9) Demonstrate the storage process of the Android file system; (10)Explain the common interface of database operation, and the operation of adding, deleting, modifying and checking database.	6	Classroom lectures, online courses, classroom discussions	Course objectives 1, 2, 3
6	**Chapter 6 System Providers** (1) Principle of ContentProvider (2)ContentProvider application	(3) Explain the principle and implementation method of ContentProvider; (4) Explain the grammatical structure and realization method of URL and URI; (5) Explain the common structure of ContentProvider and the structure and usage of system database (6) The ContentProvider interface implements functions such as adding,	2	Classroom lectures, online courses, classroom discussions	Course objectives 1, 2, 3

Serial number	Teaching content	Teaching requirements	Class hours	The way of teaching	The corresponding course target
		deleting, modifying, and checking the address book in detail.			
7	**Chapter 7 Network** (1) Network framework service (2) Network protocol (3)Volley network framework	(3) Explain the method of WebView control to access web pages; (4) Explain the principle and implementation method of the network callback interface; (5) Explain the JSON network protocol and the method of using the network framework to quickly realize JSON data analysis; (6) Explain that Volley framework realizes JSON data analysis. (7) Discuss the role of firewall in network communication, improve students' awareness of data security in the process of technology development, and stimulate students' security sensitivity.	2	Classroom lectures, online courses, classroom discussions	Course objectives 1, 2, 3

(11) The relationship between experimental teaching content and course objectives

Table 3 The relationship between experimental teaching content and course objectives

Serial number	Teaching content	Teaching requirements	Class hours	The way of teaching	Course objectives
1	**Experiment 1: Development environment installation experiment** (1)Installation and update of Android Studio; (2)Familiar with the working environment of Android Studio, and set the manifestation of Android	(1) Familiar with the installation and update of Android Studio, (2) Understand and master the working environment of Android Studio, Android SDK configuration and Gradle configuration; (3) Understand the Android project structure, understand the meaning and composition of various directories and files;	2	Classroom lectures, experiments	Course objectives 2, 3

Serial number	Teaching content	Teaching requirements	Class hours	The way of teaching	Course objectives
	Studio according to personal preferences; (3)Generate the first Android program and analyze the structure of the Android program; (4)Analyze the life cycle process of Activity through logs.	(4) Grasp the log application and breakpoint debugging methods, and understand the debugging in the Android Studio environment.			
2	**Experiment 2: Basic control experiment** (1)In the layout design of Android Studio, the interface layout is realized by drag and drop, and the interface layout and the attributes of the basic controls are set through the visual window; (2)Design interface and basic controls through code; (3)Use code binding to controls to realize common operations of basic controls by code operation.	(1) Familiar with the visual interface design of Android Studio; (2) Master the XML method of designing interface layouts and basic controls, and understand the meaning and setting methods of common attributes of layouts and controls; (3) Grasp the basic methods of code binding controls, and master the basic methods of code manipulation of controls.	2	Classroom lectures, experiments	Course objectives 2, 3
3	**Experiment 3: Data Transmission Experiment** (1)Realize data transmission between different Activities and between Activity and Fragment; (2)Broadcast data transmission and reception; (3)Multi-threaded data interaction; Data transfer between the application and external web pages.	(1) Master the principle of data transmission between different Activities and between Activity and Fragment; (2) System broadcast data sending and receiving; (3) Use WebView control and Http protocol to access web pages.	2	Classroom lectures, experiments	Course objectives 2, 3
4	**Experiment 4: Independent project design and development** (1)According to life scenarios and	(1) Understand the basic methods of Android project development; (2) Master the ability to develop software	2	Classroom lectures,	Course objectives 2, 3.4

Serial number	Teaching content	Teaching requirements	Class hours	The way of teaching	Course objectives
	applications, design a comprehensive project by yourself, complete project design documents and software development. (2)Prepare project summary report.	codes based on actual needs to achieve application capabilities; (3) Master the ability to write design documents and summarize reports.		experiments	

V Course hours allocation and teaching methods

(1) Course hours allocation (total hours are allocated in order (knowledge unit or item) list, reasonable allocation of hours required for teaching, experiment, exercises, discussion, tutoring, etc.)

Course hours distribution of theoretical teaching content

Teaching Methods / Teaching hours / Course content	Theory Teaching	Exercises Course	Discuss Course	Remarks	Subtotal
Chapter 1 Overview of mobile application development	2				2
Chapter 2 Android views and components	4				4
Chapter 3 Fragment component	4				4
Chapter 4 Broadcasting	4				4
Chapter 5 Services	6				6
Chapter 6 System provider	2				2
Chapter 7 Network	2				2
Total	**24**				**24**

Course hours distribution of experimental teaching content

Experimental project	Content Abstract	Main equipment or experimental environment used	Experimental hours	Number of people in each group	Experimental attributes (basic/comprehensive /design/research innovation)	Request (required/ optional)
Experiment 1	(1) Familiar	Eclipse、	2	1	Basic	Must to do

development environment installation experiment (1) Installation and update of Android studio; (2) Familiar with the working environment of Android studio, according to personal preferences to set the presentation of Android studio; (3) Generate the first Android program and analyze the structure of Android program; (4) Analyze the life cycle process of activity through log	with Android studio installation and update; (2) Understand and master the working environment of Android studio, Android SDK configuration and gradle configuration; (3) Understand the engineering structure of Android, understand the meaning and composition of each directory and file; (4) Master the application of log and breakpoint debugging method, and understand the debugging of Android studio environment.	Android platform				
Experiment 2 basic control experiment (1) In the	(1) Familiar with visual interface design of Android	Eclipse、Android platform	2	1	Basic	Must to do

layout design of Android studio, the interface layout is realized by drag and drop, and the interface layout and the properties of basic controls are set through the visual window; (2) The interface and basic controls are designed by code; (3) Through code binding to the control, the common operations of code operation basic control are realized.	studio; (2) Master XML design interface layout and basic control, understand the meaning and setting method of common attributes of layout and control; (3) Master the basic method of code binding control, master the basic method of code operation control.					
Experiment 3 Data transmission experiment (1) Data transmission is realized between different activities and	(1) Master the principle of data transmission between different activities and between activities and fragments;	Eclipse、Android platform	2	1	Basic	Must to do

between activities and fragments; (2) Broadcast data transmission and reception; (3) Multi thread data interaction; (4) Data transfer between application and external web pages.	(2) Data transmission and reception of system broadcast; (3) Use WebView control and HTTP protocol to access web pages.					
Experiment 4 independent project design and development (1) According to the life scene and application, design a comprehensive project, complete the project design documents and software development. (2) Prepare project summary report.	(1) Understand the basic method of Android project development; (2) Master the ability to develop software code according to the actual needs; (3) Master the ability to write design documents and summary reports.	Eclipse、Android platform and Android phone	2	Less than three	comprehensive	Must to do

(2)Teaching methods

Explain how to implement teaching methods including classroom lectures, online learning, group seminars, etc. Pay attention to the application of modern educational technology.

VI Course assessment

This course adopts the comprehensive evaluation method, including the usual results and experimental results, as shown in Table 6. The total score of the course examination (hundred point system) = usual score * 40% + experimental score * 60%.

Table 6 course assessment methods

<table>
<tr><th colspan="2">Assessment methods or approaches</th><th>Assessment requirements</th><th>Assessment weight</th><th>Corresponding course objectives</th></tr>
<tr><td rowspan="3">Usual performance</td><td>Classroom performance</td><td>According to the attendance rate, classroom performance, classroom exercises.</td><td rowspan="3">40%</td><td rowspan="3">Course objectives 1、2、3</td></tr>
<tr><td>Classroom test (open book)</td><td>According to the assessment of the correctness of the examination paper, the correctness of the steps and the ideas of solving the questions are given points as appropriate.</td></tr>
<tr><td>Online learning</td><td>According to the number of watching teaching videos, participation in discussions, posts, and questionnaire answers.</td></tr>
<tr><td colspan="2">Experimental results</td><td>According to the classroom performance, experimental design, experimental operation; The experimental results run normally, the function is completed reasonably, and the defense and experimental report are completed.</td><td>60%</td><td>Course objectives 2、3、4</td></tr>
</table>

VIII Recommended Teaching Material and Reference Books

1. Textbook

[1] Bill Phillips etc.:”Android programming: the big nerd ranch guide”（The first version）, The People's Posts and Telecommunications Press, 2014.

2. References

[1] 郭霖著：《第一行代码 Android》（第二版），人民邮电出版社，2016 年出版。

[2] 巅峰卓越等著：《Android 从入门到精通》（第一版），人民邮电出版社，2016 年出版。

[3] Google：《Android 开发者指南》，https://developer.android.google.cn/guide?hl=zh-cn，2021。

3. Network Resources

[4] Superstar platform：https://mooc1-1.chaoxing.com/course/217283160.html

Written by：Shuyun Luo **Reviewed by：Licheng Xu**

Date: 2021.05.20

Syllabus of Software Testing Project Practice

Course Name/Title：Software Testing Project Practice

Course Code： 62935

Course Type： (Specialized Course)(Compulsory Course)

Total Teaching Hours： 32 (Classroom Hours: 20, Laboratory Hours or Tutorial Hours: 12)

Course Credit： 2

I Course Introduction

Software testing technology and quality management are professional and compulsory courses for computer science and technology majors. The orientation of this course lies in the study of the basic principles of software testing and testing techniques. The focus of the course is to enable students to master how to develop high-quality software, how to ensure software quality, and how to avoid or reduce software testing risks. Through the theoretical study and practice of the course, students will master the basic concepts and principles of software testing, learn the methods of software testing, the use of software testing tools, the management of software testing processes, and understand the important way of basic knowledge and technology to design of test cases, test models, test processes, and testing tool usage. Lays a good foundation of software testing theory, technology and engineering for subsequent software testing project practice and graduation design courses.

II Course Objective

Course objective 1: Guided by the scientific concept of development, adhere to the combination of knowledge transfer and value guidance, use the themes and content that cultivate college students' political beliefs, values, and spiritual pursuits, fully integrate into the core socialist values, and treat scientific theories with philosophical thinking; broadly understand socialism build great achievements, and be able to conduct professional analysis of related achievements in combination with professional knowledge.

Course objective 2: Learn the basic concepts and principles of software testing, and be familiar with the basic concepts and implementation process of software quality management. Use the basic knowledge and theory of testing to initially have the ability to analyze and design test plans. Guided by the scientific concept of development, adhere to the combination of knowledge transfer and value guidance, use the themes and content that cultivate the political beliefs, values, and spiritual pursuits of college students, fully integrate into the core socialist values, and treat scientific theories with philosophical thinking.

Course objective 3: According to the specific test requirements of users, analyze the test environment, construct test conditions, organize and implement test activities, and cultivate students' good software engineering qualities. Have a broad understanding of the great achievements of socialist construction, and be able to conduct professional analysis of related achievements in combination with professional knowledge.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation requirement 4: Research	4-2 Formulate experimental solutions, build experimental systems, and conduct experiments based on scientific principles and methods for the overall implementation of complex computer systems.	Course Objectives 1, 2
Graduation requirement 5: Using modern tools	5-2 Use integrated development tools, open source and third-party resources to develop, debug and test computer systems, and understand their limitations	Course Objectives 1, 3

IV Correlations between Course Content and Course Objectives

IV.1 Correlations between Theoretical Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Chapter 1 An Introduction to Software Testing** (1) Definition of software testing (2) Importance and purpose of software testing (3) Basic classification of software testing	(1) Get an overview of the development process of software testing (2) Know why software testing is important; to know the misunderstanding of software testing (3) Know the objectives and principles of software testing (4) Understand the classification of software testing methodologies and contents (5) Through the introduction of software quality accident cases, students are guided to find that quality problems are not only related to human life and property, but are further	2	Lectures and discussion	Course Objectives 1

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		improved to the point that without high-quality software, there will be no modern socialist country, and students will be promoted to build socialism. A sense of mission to enhance students' awareness of the importance of course learning.			
2	**Chapter 2 Graph Coverage** (1) An introduction to graph coverage (2) Graph coverage criteria (3) Graph coverage for source code	(1) Understand the concepts of graph and graph coverage criteria (2) Master structural coverage criteria and data flow criteria (3) Master the skill of analyzing graph coverage for programs	3	Lectures, discussion and experiment	Course Objectives 1, 2
3	**Chapter 3 Logic Coverage** (1) An introduction to logic predicates and clauses (2) Logic expression coverage criteria (3) Structural logic coverage of programs	(1) Understand the concepts related to logic coverage criteria (2) Master structural logic criteria (3) Master the skill of analyzing logic coverage for programs	3	Lectures, discussion and experiment	Course Objectives 1, 2
4	**Chapter 4 Input Space Partitioning** (1) The background of input space partitioning (2) Input domain modeling (3) Input domain partitioning strategies	(1) Understand the concepts related to input space and input space partitioning (2) Know input domain modeling methods (3) Master some input space partitioning strategies (4) Master the skill of applying some input space partitioning strategies to programs	3	Lectures, and discussion	Course Objectives 1, 2
5	**Chapter 5 Practical Software Testing** (1) Integration testing (2) System testing (3) The basic process of integration and system testing	(1) Know the concept and significance of integration testing (2) Understand environment and strategy of integration testing (3) Master the testing process	3	Lectures, discussion and experiment	Course Objectives 1, 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
6	**Chapter 6 Engineering Criteria for Technologies and Building Testing Tools** (1) Testing object-oriented software (2) Testing the functionality and performance of web applications	(1) Know the characteristics of object-oriented software testing (2) Understand the way of testing web applications (3) Use functional and performance testing tools and platforms	3	Lectures, discussion and experiment	Course Objectives 1, 2
7	**Chapter 7 Software Quality Assurance** (1) Definitions of software quality and software quality assurance (2) Methods of assuring software quality (3) Software quality standard (4) Implementation procedures of software quality assurance	(1) Get an overview of software quality (2) Understand the prevention and monitoring of software quality (3) Master the standard of software quality (4) CMM (capability maturity model) (5) Understand the software quality assurance implementation procedures	3	Lectures, and discussion	Course Objectives 1

IV.2 Correlations between Experimental Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Experiment 1 White-box Testing** (1) Design test cases by using white box testing technology (2) Conduct test using Junit	(1) Able to design test cases using statement coverage, decision coverage, condition coverage and path coverage methods (2) Familiar with Junit testing tool to write test code for testing	2	Lectures, and experiment	Course Objectives 1, 2
2	**Experiment 2 Black-box Testing** (1) Design test cases by using black box	(1) Able to divide equivalence classes and design test cases according to the equivalence class division method; (2) Familiar with Junit testing tool to write	2	Lectures, and experiment	Course Objectives 1, 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	testing technology (2) Conduct test using Junit	test code for testing, and perform defect analysis			
3	**Experiment 3 Unit Testing** (1) Design test cases according to requirements of unit testing (2) Conduct test using Junit	(1) Able to use multiple methods to design and design test cases; (2) Familiar with Junit testing tool to write test code for testing and perform defect analysis.	2	Lectures, and experiment	Course Objectives 1, 2
4	**Experiment 4 Functional Testing** (1) Explain the test mode and process of Selenium test tool (2) Use Selenium to record test scripts, execute and analyze test scripts	(1) Familiar with using Selenium to perform functional tests on business websites and record the complete business operation process; (2) Set inspection points, execute test scripts and analyze the results	2	Lectures, and experiment	Course Objectives 1, 2
5	**Experiment 5 Performance Testing** (1) Introduction to performance testing and related metrics (2) Load test using Jmeter performance test tool	(1) Familiar with using Jmeter to load test business websites; (2) Familiar with monitoring performance, analyze test results	2	Lectures, and experiment	Course Objectives 1, 2
6	**Experiment 6 Web Testing** (1) Explain the related concepts of automated testing and web testing (2) Use Selenium function test, Jmeter performance test	(1) Familiar with the development of test plans and test schemes for Web systems; (2) Familiar with designing test cases for Web system testing; (3) Familiar with the use of Selenium and Jmeter5 business websites for function and performance testing, and analysis of test results	2	Lectures, and experiment	Course Objectives 1, 2

V Period Distribution and Teaching Modes

V.1 Period Distribution of Theoretical Course Content

Course content / Teaching hours / Teaching methods	Theoretical Teaching	Assignment	Discussion	Note	Total
Chapter 1 An Introduction to Software	2				2
Chapter 2 Graph Coverage	2		1		3
Chapter 3 Logic Coverage	3				3
Chapter 4 Input Space Partitioning	3				3
Chapter 5 Practical Consideration	3				3
Chapter 6 Engineering Criteria for	3				3
Chapter 7 Software Quality Assurance	2		1		3
Total	18		2		20

V.2 Period Distribution of Experimental Course Content

Name	Contents	Environments	Period	Group size	Attributes（Basic /Comprehensive/ Design/ Creative Study）	Requirements (Compulsory /Optional)
Experiment 1 White-box Testing	Take specific programs as an example to conduct white-box testing and master basic logic coverage methods	Junit	2	1	Basic	Compulsory
Experiment 2 Black-box Testing	Take specific procedures as an example to conduct black-box testing, and master methods such as equivalent classification and boundary value analysis.	Junit	2	1	Basic	Compulsory
Experiment 3 Unit Testing	Familiar with the application of the unit testing tool Junit for Java program unit modules.	Junit	2	1	Basic	Compulsory
Experiment 4 Functional Testing	Use Selenium to perform functional testing on Mercury Tours website	Selenium	2	1	Basic	Compulsory
Experi	Use Jmeter to test the	Jmeter	2	1	Basic	Compulsory

ment 5 Performance Testing	concurrency of an application system					
Experiment 6 Web Testing	Use Web testing tools to perform functional testing and performance testing on the Web system.	Selenium, Jmeter	2	1	Comprehensive	Compulsory
Total			12			

V.3 Teaching Mode

This course is a course that emphasizes both theory and practice. The course group should take "cultivating advanced application-oriented talents" as the starting point, guided by modern teaching concepts, and carefully conduct teaching design, embodying thought of "teacher as the leading role and students as the main body". On the basis of analyzing the characteristics of the curriculum, it is determined that the structure of the curriculum system is composed of classroom teaching, experimental teaching, auxiliary teaching activities and innovative practice projects, and each part is organically combined to form an organic whole. The curriculum focuses on cultivating students' practical ability and innovative consciousness, exploring the teaching mode that combines multimedia classroom teaching and case-driven practical teaching, advocating heuristic teaching and application skills training, stimulating students' interest and potential, and improving students' software testing skills and skills. Engineering literacy enhances students' sense of collaboration and teamwork.

VI Assessment

The comprehensive assessment of this course includes classroom performance, online assignment, experiment and exercise. Assessment score (Hundred Marking System) = classroom performance * 10% + online assignment * 30% + experiment and exercise * 60%.

The final grade is assessed by Five-Levels Marking System: A-Excellent (90 - 100), B-Good (80 - 89), C-Medium (70 - 79), D-Passing (60 - 69), E-Failure (Below 60).

Assessment Methods or Approaches		Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Usual grade	Classroom performance	Attendance, class interaction, class exercise	10%	Course Objectives 1, 2, 3
	Online Assignment	Online assignment submit and scores	30%	
Experiment and exercise		Classroom performance, experimental solution design, experimental operation, and experimental report	60%	Course Objectives 1, 2, 3

Note: 1. Assessment methods or approaches mainly include classroom performance, online assignment, and experiments, etc.

2. Assessment requirements include times of assignment and experiment requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score

VII Textbooks and References

3. Textbooks

[4] P. Ammann. Introduction to Software Testing. Cambridge University Press, 2008.

4. References

[4] L. Gu, J. Shi. Introduction to Software Testing Technology. Tsinghua University press, August 2011.

[5] R. Patton. Software Testing (The second edition). China Machine press, July 2005.

[6] G. G. Schulmeyer. Handbook of Software Quality Assurance. Artech House, 2007.

Written by：Tingting Wu **Reviewed by: Qi Sun**

Date： 2021.05.20

《工程伦理与项目管理》教学大纲

课程中文名称：工程伦理与项目管理 **课程代码：** 60401

课程英文名称：Engineering Ethics and Project Management

课程类别与性质：通识教育课

总学时：32 学时（其中讲课 32 学时）

学 分：2

先修课程：面向对象程序设计、软件工程

适用专业：计算机科学与技术

开课系(室)：计算机

一、课程简介

工程伦理与项目管理是计算机专业一门通识教育课，是一门基础理论结合实际应用的重要课程。通过本课程的学习，培养学生对工程的生态环境、公众健康、社会安全、人文关怀等深刻的认识，建立高度社会责任感、正确的价值观和强烈的伦理道德意识。工程伦理与项目管理是研究以工程化的理论、方法和技术为指导，管理较大规模软件研发项目的学科，在软件工程领域占有十分重要的地位。为了理论结合实践，本课程采用大量分散案例来解释和验证软件项目管理领域的基本概念、基本原理及基本方法，同时采用综合案例将整个知识内容有机地结合在一起。

二、课程教学目标

课程目标 1：通过讲解工程伦理以及项目管理在实践中的应用，培养学生刻苦钻研的精神和工匠精神。

课程目标 2：掌握工程伦理、项目和项目管理的基本概念与知识，总结描述项目管理领域的最新问题与标准，了解项目管理相关职业的发展现状，培养相关从业者的工程伦理意识。通过学习项目管理的历史发展过程, 弘扬学生的爱国情怀，增强他们的民族自豪感。

课程目标 3：掌握项目产品的全周期开发流程、基本设计与开发方法，了解影响设计项目开发的各种因素。

课程目标 4: 掌握十个项目管理知识领域的详细管理过程，包括项目综合、范围、时间、成本、质量、人力资源、沟通、风险、采购、干系人管理等，能够利用管理过程的相关工具与技术，以及理解不同工具和技术的局限性。

课程目标 5： 通过系统学习，增强学生的项目管理意识和创新意识，掌握具体工程领域的伦理规范要求，增强学生竞争意识和开拓创新能力，提高工程伦理的决策能力，提高学生分析问题和解决问题的能力，能够解决工程实践中的复杂伦理问题。

课程目标 6：掌握软件项目管理方法，将软件项目管理知识和方法运用于具体的案例分析和实践中，完成项目的时间与成本效益分析。利用网络和文献数据库资源，获取新的项目管理成功案例信息以及前沿的项目管理方法

三、课程教学目标与毕业要求的对应关系

毕业要求	毕业要求指标点	课程目标
毕业要求 1	1-3 能够将工程基础知识、专业知识和数学模型用于推演工程问题，并分析其中的影响因素；	课程目标 2
毕业要求 3	3-1 掌握计算机系统及软件产品全周期开发流程、基本设计/开发方法和技术，了解影响设计目标和技术方案的各种因素；	课程目标 3
毕业要求 5	5-3 能够使用现代软件工程工具，对软件项目的实施过程进行管理，并理解其局限性。	课程目标 4
毕业要求 6	6-2 能够分析评价计算机专业的工程实践和复杂工程问题解决方案对社会、健康、安全、法律以及文化的影响，以及这些制约因素对项目实施的影响，并理解应承担的责任。	课程目标 1，5
毕业要求 11	11-2 运用成本效益评估方法，进行工程方案的成本效益分析。	课程目标 6

四、教学内容与课程目标的关系

课程教学内容与课程目标的关系

序号	教学内容	教学要求	学时	教学方式	对应课程目标
1	**第 1 章 工程与伦理** （1）如何理解工程，包括定义、过程和维度； （2）如何理解伦理，包括伦理困境与伦理选择； （3）工程中的伦理问题及特点、辨识、处理的基本原则和基本思路。	（1）阐述工程伦理的基本概念； （2）阐述工程与技术的关系、工程的基本内涵、工程的基本环节、工程实践的特点； （3）道德与伦理的区别与联系的辨析。	2	课堂讲授	课程目标 2、5
2	**第 2 章 工程中的风险、安全与责任** （1）工程风险的来源、可接受性、防范与安全； （2）工程风险的伦理评估原则、评估途径和评估方法； （3）工程风险伦理责任定义、责任的主题和责任的类型。	（1）了解工程风险的来源，掌握防范工程风险的措施； （2）熟悉工程风险的伦理评估原则、途径与方法； （3）理解工程中“伦理责任”的含义、主体与类型，增强学生的责任感意识。	2	课堂讲授	课程目标 2、5

序号	教学内容	教学要求	学时	教学方式	对应课程目标
3	**第 3 章 工程中的价值、利益与公正** （1）工程的价值导向性、工程价值的多元性、工程价值的综合性； （2）工程所服务的对象和可及性； （3）工程实践中的邻避效应、社会成本承担； （4）基本公正原则、利益补偿机制和利益协调机制。	（1）理解工程的价值及其特点； （2）理解工程服务的可及性； （3）从社会成本和利益攸关方的角度理解工程实践中的公正问题； （4）工程实践中的基本公正原则，以及实现工程公正的机制和途径。	2	课堂讲授	课程目标 2、5
4	**第 4 章 工程师的职业伦理** （1）工程职业的地位、性质与作业；工程职业制度； （2）工程师职业伦理章程和工程职业伦理的实践指向； （3）工程师的职业伦理规范，包括首要责任原则、权利和责任、职业美德和应对职业行为中的伦理冲突。	（1）了解、掌握工程职业的地位、性质与作用，并加强对 工程职业伦理标准的认识； （2）工程师职业伦理规范有整体性认识，能清楚理解工程师在职业活动中的权利与责任，准确认知工程职业活动中的主要伦理问题， 并具备分析具体工程伦理问题的能力； （3）培养学生的工程职业精神，使学生初步具有面对较为复杂的工程伦理困境时的伦理意志力和解决问题的方案与能力。	2	课堂讲授	课程目标 2、5
5	**第 5 章 项目管理和 IT 背景** （1）基本概念，包括项目、项目管理、项目群和项目组合管理、项目管理的系统观点； （2）项目经理的作用； （3）组织的框架、结构和文化； （4）项目阶段和项目生命周期。	（1）掌握项目、项目管理、项目群和项目组合管理的概念；了解项目经理的工作、作用和应掌握的技能； （2）熟悉组织的 4 个框架、3 种结构，以及组织文化的含义；掌握项目阶段和管理评审的重要性； （3）了解 IT 项目的环境和影响 IT 项目管理的最新趋势 （4）通过介绍两弹一星、长城建造过程中的项目管理方法，弘扬学生的爱国情怀，增强他们的民族自豪感。	4	课堂讲授	课程目标 2、3
6	**第 6 章 项目管理过程组**	（1）描述 5 个项目管理的过程组、每个	2	课堂	课程目标

序号	教学内容	教学要求	学时	教学方式	对应课程目标
	（1）启动、计划、执行、监控和收尾等 5 个项目管理过程组； （2）把项目管理过程组映射到 10 个知识领域。	过程组的典型活动层次以及它们之间的相互作用； （2）理解项目管理过程组与项目管理知识领域的关系； （3）讨论为满足需要，组织如何发展 IT 项目管理的方法学。		讲授	2、3
7	**第 7 章 项目管理知识领域** （1)10 个项目管理知识领域(包括综合管理、范围管理、时间管理、成本管理、质量管理、风险管理、沟通管理、采购管理、人力资源管理和干系人管理）的基本概念； （2）每个项目管理知识领域的关键理论和重要性； （3）每个项目管理知识领域包含的具体管理过程及相关的方法和技术。	（1）描述 10 个项目管理知识领域的整体框架；理解每个项目管理知识领域的重要性；掌握每个项目管理知识领域的具体管理过程； （2）理解每个知识领域的关键理论，区分不同理论的差异及特点。 （3）掌握每个项目管理知识领域的管理过程相关的输入、输出和工具与技术，理解不同工具和技术的局限性。 （4）针对具体案例应用项目管理知识和方法；获取新的项目管理成功案例信息以及前沿的项目管理方法。 （5）结合实际案例，谈及项目时间管理的重要性，传授项目时间管理方法，让学生意识到时间的重要性，借助科学的方法，合理高效地利用时间。	18	课堂讲授	课程目标 1、4、5、6

五、课程学时分配及教学方法

（一）课程学时分配

理论性课程教学内容课时分配表：

教学方式 教学时数 课程内容	理论讲授	习题课	讨论课	备注	小计
第 1 章 工程与伦理	2	0	0		2
第 2 章 工程中的风险、安全与责任	2	0	0		2
第 3 章 工程中的价值、利益与公正	2	0	0		2
第 4 章 工程师的职业伦理	2	0	0		2
第 5 章 项目管理和 IT 背景	4	0	0		4

第 6 章 项目管理过程组	2	0	0		2
第 7 章 项目管理知识领域	18	0	0		18
总 计	32		0		32

（二）教学方法

工程伦理与项目管理是一门理论性和实践性都很强的课程。在理论课教学中，以重点知识讲授为基础，以案例教学为特点，以 IT 项目管理与职业伦理教育为重心。可采用课堂讲授与案例研讨等多种方式相结合。同时可结合 MOOC、头脑风暴等多种方式进行教学。

为使学生充分发挥学习主动性，配套在线教学视频供学生预习，在教学网络平台上布置一些作业题供学生思考并网上提交回答。通过发放问卷，及时了解学生学习情况，并做出教学方式方法的调整，及时答疑解惑。

六、课程考核

本课程采用综合评价的考核方式，包括平时成绩与论文报告成绩，如表 6 所示。课程考核总成绩（百分制）= 平时成绩*40%+项目报告成绩*60%。

考核方式或途径		考核要求	考核权重	对应课程目标
平时成绩	课堂表现	根据出勤率、课堂表现、课堂小练习情况评定。	20%	课程目标 2、3、4
	平时作业	根据作业上交次数、正确率、书写质量评定。	20%	
期末成绩 (项目报告)		项目背景与研究现状、理论基础、方法学习、分析总结	60%	课程目标 1、4、5、6

七、推荐教材及参考资料

（一）教材：

[1] Kathy Schwalbe 编著：《IT 项目管理》（原书第 7 版），机械工业出版社，2015 年出版。

（二）参考书：(列出书名、作者、出版社等)

[1] 徐海涛编著：《工程伦理》，电子工业出版社，2020 年出版。

[2] 查尔斯 E.哈里斯等编著：《工程伦理：概念与案例》（第五版），浙江大学出版社，2018 年出版。

[3] 顾剑等编著：《工程伦理学》，同济大学出版社，2015 年出版。

[4] 项勇编著：《工程项目管理》（第 4 版），机械工业出版社，2017 年出版

（三）网络资源:

[1] https://www.icourse163.org/course/XMU-1002608004?from=searchPage

执笔：相东明　　　审稿：林望　　　　　　审定：信息学院教学委员会

制（修）订时间：2021 年 5 月 20 日

Syllabus of Embedded System Project Practice

Course Name/Title： Embedded system project practice **Course code：62928**

Course Type： Specialized Course，Optional Course

Total Teaching Hours： 40 (2 week)

Course Credit： 2

I Course Introduction

Embedded system project practice is a project practice course related to *embedded system principle and design*.

This practical, hands-on course introduces the various building blocks and underlying scientific and engineering principles behind embedded real-time systems. The course covers the integrated hardware and software aspects of embedded processor architectures, along with advanced topics such as real-time, resource/device and memory management. Students can expect to learn how to program with the embedded architecture that is ubiquitous in smartphones, portable gaming devices, robots, etc. Students will then go on to learn and apply real-time principles that are used to drive critical embedded systems like automobiles, avionics, medical equipment, etc. Topics covered include embedded architectures (building up to modern 16/32/64-bit embedded processors); interaction with devices (buses, memory architectures, memory management, device drivers); concurrency (software and hardware interrupts, timers); real-time principles (multi-tasking, scheduling, synchronization); implementation trade-offs, profiling and code optimization (for performance and memory); embedded software (exception handling, loading, mode-switching, programming embedded systems). Through a series of laboratory exercises with state-of-the-art embedded processors, sensors, actuators and industry-strength development tools, students will acquire skills in the design/implementation/debugging of core embedded real-time functionality.

II Course Objective

Course objective 1: Determine the project considering both the benefit from the technology and society, conduct multi-dimensional analysis, participate in the whole process of the project with the spirit of craftsman and professionalism, and reflect the national feelings of engineers.

Course objective 2: Apply the design, configuration, implementation, testing and engineering of embedded system (single chip microcomputer) by comprehensively using the acquired professional ability and background knowledge.

Course objective 3: Design the system under the constraints of real-time, low power consumption, reliability and hardware.

Course objective 4: Transmit the data between devices, or between embedded system and PC. Present data results clearly. Build the reactive control system properly.

Course objective 5: Design embedded system software by applying the principle of hierarchical design, which includes the hardware abstraction layer, business logic layer and data display layer, and skillfully use various resource to build the embedded system software. Be able to use state diagram to model the system, and have the ability to implementation the model into real system.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
GR 1	1-4 Be able to use engineering basic knowledge, professional knowledge and mathematical model for scheme comparison and analysis of computer complex engineering problems.	Course Objective 1,2
GR 2	2-1 Can identify computer complex engineering problems, and through research, clearly express the needs and key processes of computer complex engineering problems.	Course Objective 2,3
GR3	3-3 Able to follow the software engineering specifications to achieve the requirements of the computer system, according to the design solution as well as fully consideration of cost-effective.	Course Objective 3,4
GR5	5-1 Be able to use software and hardware simulation tools to verify the computer related theory, simulate and analyze the system design scheme, and understand its limitations	Course Objective 5

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	Detail the purpose, content and requirements of curriculum design, clear assessment methods	(1) The content of curriculum design must have certain engineering complexity, clear task objectives and detailed function description; (2) The task complexity is graded from low to high; (3) Through the demonstration of the past excellent works, stimulate students to challenge their professional spirit	4	Lecture	1,2
2	Analyze the system requirements, explain the components and key factors of embedded system	Help students to clarify the task objectives, understand the role of each component in their own design system, as well as the key factors of the design	4	Lecture	2
3	sensors and actuators design	Selection, accuracy, cost and requirement analysis of sensors and actuators	8	Experiment /group discussion/ presentation	2
4	Communication design	Requirement analysis and design of communication	8	Experiment /group discussion/ presentation	3
5	Real time, low power and reliability design	Real time, low power and reliability design	8	Experiment /group discussion/ presentation	4
6	Project replay	Students demonstrate and explain the results of the project. Teachers should ask at least three questions for students to answer	4	presentation	2,5
7	Final written project document	Final written project document should at least include analysis, design, implementation and summary of four parts, illustrated, logical clear, standard format	4	Lecture	2,5

V Period Distribution and Teaching Modes

7. Practice period distribution

Course Content	Content	Device & lab enviroment	Hours	No.per Group	Experiment type	Compulsory
topic selection and task interpretation	Detail the purpose, content and requirements of curriculum design, clear assessment methods	PCB, development tools	4	2-3	Research and innovation	Compulsory
System solution and design	分析各个模块的功能，设计部署的方案	PCB, development tools	4	2-3	comprehensive	Compulsory
sensors and actuators design	分析传感器和执行器的特点，完成相关设计	PCB, development tools	8	2-3	design	Compulsory
Communication design	选择并确定通信设计方案	PCB, development tools	8	2-3	design	Compulsory
Real time, low power and reliability design	选择并确定实时性，低功耗和可靠性设计方案	PCB, development tools	8	2-3	design	Compulsory
Project replay	项目答辩	PCB, development tools	4	2-3	comprehensive	Compulsory
Final written document	Final written document , including task content, purpose, requirements, indicators, business process design, function design, module design, code and operation combination and analysis	PCB, development tools	4	comprehensive	comprehensive	Compulsory
总计			40			

2. Teaching methods

1. This course is mainly based on design and hands-on experiments. The teacher needs

to explain to the students the nature, tasks, requirements, course arrangement and progress, assessment contents, experimental rules and laboratory safety system of the course.

2. Students must complete the whole process of hardware platform and software application design and engineering of the selected project, with high overall complexity. Students are encouraged to challenge themselves.

3. Encourage the collaborative work of different majors and backgrounds. Focus on developing the following non-technical abilities:

- Decomposing, structuring, and formulating solutions to unstructured problems.
- Assessing what can be done and delivering a product on time.
- Interdisciplinary problem solving: Data collection, analysis, and synthesis, formulation and evaluation of policy recommendations.
- Developing professional oral and written communication skills through participation in oral presentations, and preparation of the final written project document.
- Developing the ability to function on multidisciplinary teams.

VI Assessment

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Final Document	the rationality, integrity and format standardization of the project design	30	Course objects 2,3,4,5
Function completion	According to the hierarchical scoring mechanism in the project assignment	40	Course objects 2,3,4,5
Project Replay	Score according to the answers to the teacher's questions	20	Course objects 2,3,4,5
class attendance	Score according to the class attendance	10	Course objects 1

VII Textbooks and References

3. Textbook

[1] 贾宇波，夏劲松，李俊松.《嵌入式系统原理与设计》(自编教材)．浙江理工大学，2021

4. References

[1] 王宜怀，《嵌入式技术基础与实践》(第 5 版)，清华大学出版社，2019.4。

[2] 卡莫尔，《嵌入式系统:体系结构、编程与设计》(第 2 版)，清华大学出版社，2010。

[3] 西蒙，《嵌入式系统软件教程》，机械工业出版社 2005.。

Written by：Jinsong Xia **Reviewed by: Junsong Li**

Date：2021.05.20

Syllabus of Mobile Application Project Practice

Course Name/Title： Mobile Application Project Practice **Course code：** 62915

Course Type： Specialized Course,Optional Course

Total Teaching Hours： 40 (Laboratory Hours or Tutorial Hours 40)

Course Credit： 2.0

I Course Introduction

"Mobile Application Project Practice" is optional course in computer science and technology and related majors. The mobile project cases are the main tool to learn the contents. It is required the students to familiar with installation and configuration in the development environment, and gradually understand and grasp the Android system development by basic methods, master the important components in Android applications, such as Activity, Intent, Services Broadcast, UI, learn simple design, grasp the common controls, data storage, file storage, networking, multi thread programming, custom view, audio file control and GPS positioning applications. Students should have the ability to analyze and solve problems for a given design goal. In the implementation of the project, we should focus on cultivating students' team spirit.

II Course Objective

Course Objective1: To understand and master the basic structure and development methods of Android system, skillfully build the development environment of Android system, learn the basic methods of programming, debugging.

Course Objective2: To master the development of important components in Android system, to grasp the common controls, data storage, network communication, graphics rendering, audio and video playback, multi-thread programming; To master application design, programming and debugging methods.

Course Objective3: To improve the ability to analyze problems, develop good programming habits; To develop communication skills and team spirit.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation Requirement 1 Engineering knowledge	1-4 Be able to use engineering basic knowledge, professional knowledge and mathematical model for scheme comparison and analysis of computer complex engineering problems.	Course Objective 1,2

Graduation Requirement 2 Problem analysis	2-1 Be able to identify complex computer engineering problems, and clearly express the requirements and key processes of complex computer engineering problems through research;	Course Objective 1,2
Graduation Requirement 3 Design/ development solutions	3-1 Master the whole cycle development process, basic design /development methods and technologies of computer system and software products, and understand various factors influencing design objectives and technical solutions;	Course Objective 1,2,3
Graduation Requirement 4 Research	4-1 Be able to research and verify the key algorithms and modules related to computer science.	Course Objective 1,2,3
Graduation Requirement 9 Individuals and teams	9-1 Be able to work with other discipline members, be competent for the role of individual and team members and take corresponding responsibilities.	Course Objective 1,2,3

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Experiment 1：Build Android Development Environment.** (1) Android overview, Android characteristics and advantages; (2) Development components of Android SDK software; (3) Build the Android development environment in Window system; (4) APP program editing, compiling and running methods; (5) Android system architecture.	(1) To master the development methods and steps of Android in Windows system and to complete the preparation of simple procedures for testing; (2) Focus on the development and use of Android development platform.	4	Classroom teaching and experiment	Course objectives 1, 2,3
2	**Experiment 2：Use of Basic Interface Elements.** (1) Android development, the	(1) To master the layout methods of interface elements; (2) To master the use of basic	4	Classroom teaching and experiment	Course objectives 1, 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	layout methods of interface elements; (2) The use of basic interface elements: TextView and EditView, Botton and ImageButton.	interface elements:TextView and EditView, Botton and ImageButton; (3) To understand the basic methods of Android platform UI design.			
3	**Experiment 3：Activity, Service, Intents, and Life cycle.** (1) The basic components of applications, introduce to Activity, Service, BroadCast Receiver, Content Provider, Intents; (2) The application life cycle.	(1) To master the programming method of Activity, Service, BroadCast Receiver, Content Provider, Intents; (2) To understand the application life cycle.	4	Classroom teaching and experiment	Course objectives 1, 2
4	**Experiment 4：Menu Design.** Menu settings, layout and programming methods.	To master the commonly used interface elements of the menu.	2	Classroom teaching and experiment	Course objectives 1, 2
5	**Experiment 5 : Widget Components.** The use of Widget components.	(1) To master the use of Widget components; (2) Understand the basic programming methods of Widget components.	2	Classroom teaching and experiment	Course objectives 1, 2
6	**Experiment 6：Interface Control Design.** List (ListView), dialog box (Dialog), interface layout design, interface events.	To master the use of list (ListView), dialog box (Dialog), the programming method of interface events.	2	Classroom teaching and experiment	Course objectives 1, 2
7	**Experiment 7 : Drawing Two-dimensional Graphics.** (1) The process and method of	(1) To master the Android system on the two-dimensional graphics rendering method, according to the	2	Classroom teaching and experiment	Course objectives 1, 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	drawing two-dimensional graphics; (2) The completion of the two-dimensional graphics rendering.	application scenarios and requirements to draw a simple two-dimensional map; (2) To understand the two-dimensional graphics rendering on Android system.			
8	**Experiment 8：Music Player Software.** Insert the background music in the game.	(1) To master the programming method of audio playback on Android system; (2) To understand the basic process of audio programming on Android system.	2	Classroom teaching and experiment	Course objectives 1, 2
9	**Experiment 9：Video Player Software.** Design and development of a simple video player.	(1) To master the basic methods of video programming on Android system; (2) To understand the basic process of video programming on Android system.	2	Classroom teaching and experiment	Course objectives 1, 2
10	**Experiment 10：Game Design.** Knowledge and methods to write a small game. Write a game design report first, and then realize the programming development.	(1) To improve the ability of comprehensive application of knowledge and methods; (2) To understand the basic development process of Android system.	2	Classroom teaching and experiment	Course objectives 1, 2,3
11	**Experiment 11：Two Cubes Creation.** Create two cubes by OpenGL in Android system.	(1) To draw the basic three-dimensional graphics and master OpenGL programming methods; (2) To understand the basic principles of three-dimensional graphics and OpenGL programming process.	2	Classroom teaching and experiment	Course objectives 1, 2

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
12	**Experiment 12 : Collision Detection of Two Cubes.** Create two cubes by OpenGL, and programming the collision detection of two cubes.	(1) To master OpenGL programming methods by drawing the basic three-dimensional graphics and controlling the location of graphics; (2) To further understand the principles of three-dimensional graphics and OpenGL programming methods.	2	Classroom teaching and experiment	Course objectives 1, 2
13	**Experiment 13 : Data Storage,SQLite Database.** (1) Overview of Android data storage; (2) The basic method of data storage and file storage, SQLite database programming. Design a small database as required.	(1) To Master data storage methods on Android system and SQLite database programming; (2) To understand the basic methods of data storage, the use of SQLite database.	2	Classroom teaching and experiment	Course objectives 1, 2,3
14	**Experiment 14: Address Book of Mobile Phone.** Address Book of Mobile Phone. Imitation address book software of mobile phones, design and programming to achieve a simple phone address book.	(1) To master the basic methods of data storage; (2) To understand the basic methods of data storage on mobile phone, to complete the design and development of mobile phone address book by the learned knowledge.	4	Classroom teaching and experiment	Course objectives 1, 2,3
15	**Experiment 15：Comprehensive Experiment of Mobile Application Project.** Design a mobile application for life, study, work and other purposes.	(1) Learn to analyze requirements according to application scenarios; (2) Master the design method of mobile application software.	4	Classroom teaching and experiment	Course objectives 1, 2,3

V Period Distribution and Teaching Modes

1. Period Distribution

Experiment name	Content	Main equipment or experimental environment used	Period	Number of people in each group	Experimental attributes (Basic / Comprehensive / Design / Research innovation)	Request (Required / Optional)
Experiment 1: Build Android Development Environment.	(1) Android overview, Android characteristics and advantages; (2) Development components of Android SDK software; (3) Build the Android development environment in Window system; (4) APP program editing, compiling and running methods; (5) Android system architecture.	Computer	4	1	Basic	Required
Experiment 2: Use of Basic Interface Elements.	(1) Android development, the layout methods of interface elements; (2) The use of basic interface elements: TextView and EditView, Botton and ImageButton.	Computer	4	1	Basic	Required
Experiment 3: Activity,	(1) The basic components of	Computer	4	1	Basic	Required

Service,Intents,and Life cycle.	applications, introduce to Activity, Service, BroadCast Receiver, Content Provider, Intents; (2) The application life cycle.					
Experiment 4 : Menu Design.	Menu settings, layout and programming methods.	Computer	2	1	Basic	Required
Experiment 5 : Widget Components.	The use of Widget components.	Computer	2	1	Basic	Required
Experiment 6 : Interface Control Design.	List (ListView), dialog box (Dialog), interface layout design, interface events.	Computer	2	1	Basic	Required
Experiment 7 : Drawing Two-dimensional Graphics.	(1) The process and method of drawing two-dimensional graphics; (2) The completion of the two-dimensional graphics rendering.	Computer	2	1	Basic	Required
Experiment 8 : Music Player Software.	Insert the background music in the game.	Computer	2	1	Basic	Required
Experiment 9 : Video Player Software.	Design and development of a simple video player.	Computer	2	1	Basic	Required

Experiment 10: Game Design.	Knowledge and methods to write a small game. Write a game design report first, and then realize the programming development.	Computer	2	1	Comprehensive	Required
Experiment 11: Two Cubes Creation.	Create two cubes by OpenGL in Android system.	Computer	2	1	Basic	Required
Experiment 12: Collision Detection of Two Cubes.	Create two cubes by OpenGL, and programming the collision detection of two cubes.	Computer	2	1	Comprehensive	Required
Experiment 13: Data Storage, SQLite Database.	(1) Overview of Android data storage; (2) The basic method of data storage and file storage, SQLite database programming. Design a small database as required.	Computer	2	1	Comprehensive	Required
Experiment 14: Address Book of Mobile Phone.	Address Book of Mobile Phone. Imitation address book software of mobile phones, design and programming to achieve a simple phone address book.	Computer	4	1	Comprehensive	Required
Experiment 15: Comprehensive Experiment of Mobile	Design a mobile application for life, study, work and other purposes.	Computer	4	1	Comprehensive	Required

Application Project.						

2. Teaching Modes

6) Teaching modes include classroom teaching, experiments, online courses, classroom discussions, comprehensive demonstration, etc.
7) In the process of the experiment, students should be familiar with the programming environment and master the use of debugging tools.
8) Special stress should be put on the introduction of modern educational technology with an optimal integration of various teaching media.
9) Make reasonable use of presentation instruments, inquiring instruments, interaction instruments and design instruments to effectively improve the teaching quality.

VI Assessment

(Assessment content and methods must correlate with course objectives.)

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Classroom Performance	Asking for leave in advance when you are unable to attend the course; Answering teacher's questions properly.	10%	Course objectives 1, 2
Experiment	8 to 10 experiments are required to complete the experiment report. The program can run normally and has friendly UI design.	30%	Course objectives 1, 2, 3
Comprehensive Experiment	Normal experimental results and reasonable functions; Independently writing experimental report.	60%	Course objectives 1, 2, 3

VII Textbooks and References

9. Textbook

[1] Zhang Simin, Android application design, Tsinghua University press, 2013.

10. References

[1] Burnett (Ed Burnette), Android basic tutorial (Fourth Edition), People's Posts and Telecommunications Press, 2016.

[2] Black horse programmers,Android mobile development basic case tutorial, People's Posts and Telecommunications Press, 2017.

Written by：Xu Licheng **Reviewed by: Luo Shuyun**

Date：2021.05.20

Syllabus of Software Testing Project Practice

Course Name/Title：Software Testing Project Practice

Course Code： 62935

Course Type：(Specialized Practical Course)(Optional Course)

Total Teaching Hours：40 (Classroom Hours: 4, Laboratory Hours or Tutorial Hours: 36)

Course Credit： 2

I Course Introduction

Software testing project practice is a professional practice optional course for computer science and technology majors. It is a comprehensive testing practice for the course of software testing and quality assurance. It is one of the professional practice courses that trains students' basic software testing skills and carry out follow-up professional course study and graduation design. Through the study of this course, students are required to master the basic concepts and basic theories of software testing, and be able to formulate test plans, design test case sets, build test environments, complete unit testing, functional testing, performance testing, security testing, achieve test results, summarize and write defect reports and comprehensive test reports based on test tasks to achieve the course objectives of this course.

II Course Objective

Course objective 1: Through understanding the relevant science and technology culture and technological development trends in the application field of software testing, enhance the sense of responsibility for a “powerful country in science and technology”; Recognize the application and embodiment of informatization, intelligence, and automation in the development of software testing technology, and perceive that science and technology are the first productive force in software testing technology; Cultivate the sense of innovation in the process of training practical hands-on ability.

Course objective 2: Learn the basic concepts and principles of software testing, be familiar with the basic concepts and implementation process of software quality management, and understand the practical significance of software testing in engineering applications. Use the basic knowledge and theory of software testing, analyze test requirements, formulate test solutions and test plans, design test cases, build test environments, and construct practical and effective experimental programs. Understand the cutting-edge technology of software testing, and enhance the sense of responsibility as a “powerful country in science and technology”.

Course objective 3: According to user specific test requirements, analyze the test environment, construct test conditions, organize and implement test activities, and be able to analyze and solve engineering test problems by using software test theories and methods comprehensively. Use various automated testing tools. Recognize the application and embodiment of informatization, intelligence, and automation in the development of software testing technology. Perceive that science and technology is the embodiment of the first productive force in software testing technology.

Course objective 4: Able to independently complete the design of this course, analyze and summarize the experimental process, write defect reports and feed back to the design practice of complex projects, improve students’ ability to independently analyze and solve practical problems, and cultivate the craftsmanship spirit of assiduous research in the process of training practical hands-on ability.

III Correlations between Course Objectives and Graduation Requirements

Graduation Requirements	Graduation Requirements Index Point	Course Objectives
Graduation requirement 4: Research	4-2 Formulate experimental solutions, build experimental systems, and conduct experiments based on scientific principles and methods for the overall implementation of complex computer systems.	Course Objectives 1, 2
Graduation requirement 5: Using modern tools	5-2 Use integrated development tools, open source and third-party resources	Course Objectives 1, 3

	to develop, debug and test computer systems, and understand their limitations	
Graduation requirement 10: Communication	10-1 Make effective written and oral presentations on complex engineering issues, and communicate effectively with others, including writing reports and design documents, making statements, expressing clearly or responding to instructions.	Course Objective 1, 4

(Note: Basic courses and specialized courses must correlate with the graduation requirements as specified in the Program outline. The correlated graduation requirement index point must be put before the descriptive phrases or sentences. General courses are exempted from this rule.)

IV Correlations between Course Content and Course Objectives

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
1	**Course content 1. Instruct project design and test plans** (1) Introduce testing principles and testing requirements. (2) Students are required to look up information on their own, determine the test application system that meets the requirements, select reasonable modules or components, and write a test plan. (3) Instruct making test plans.	(6) Master the basic theories and skills of software testing, so as to have the ability to test comprehensively. (7) Be able to think, search materials and books, and design testing solutions independently. (8) Write test plans according to testing solutions. (9) Perceive the embodiment of "science and technology is the primary productive force" in software testing technology. Strengthen the sense of responsibility for a "powerful country in science and technology" and establish lofty ideals.	4	Lectures and discussion	Course Objectives 1, 2
2	**Course content 2. Design test cases** (1) Review the white-box testing methods, and design test cases using logic coverage, path	(1) Master the common methods of white box testing and black box testing; (2) According to test requirements and test plans, select logic coverage, path testing and other methods to design test cases and write test scripts;	4	Lectures and discussion	Course Objectives 2, 3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
	testing, program instrumentation and other methods. (2) Review the black box test method, and design test cases using equivalence class partition, boundary value analysis, decision table driven method, causality diagram method, function diagram and other methods. (3) Summarize selection strategy of testing methods.	(3) According to the test requirements and test plan, select equivalence class partition and boundary value analysis methods to design test cases and write test scripts;			
3	**Course content 3. Unit testing** (1) Define the process of unit testing. (2) List methods of design test cases. (3) Display the use of testing tools xUnit.	(1) Be familiar with the use of xUnit testing tools. (2) Be able to use xUnit tool to write unit test content.	6	Lecture, discussion, and experiment	Course Objectives 2, 3
4	**Course content 4. Functional testing** (1) Define the process of functional testing. (2) Introduce the automation of functional testing. (3) Display the use of functional testing tools.	(3) Master the functional testing technology and content, and master the key points, methods and common processes of functional testing. (4) Master the tool usage of common function recording templates, perform functional tests on business websites, and record the complete business operation process. (5) Execute test cases and analyze the results to find defects. (6) The automation of functional testing is the need of a powerful modern socialist country in software quality. Because modern power country not only requires	8	Lecture, discussion, and experiment	Course Objectives 1, 2, 3

Num.	Course Content	Teaching Requirement	Period	Teaching modes	Course Objectives
		high-quality software, but also requires high efficiency and low-cost software testing. The automation of software testing is one of the effective ways to improve testing efficiency and reduce testing costs.			
5	**Course content 5. Performance testing** (1) List the metrics for performance testing. (2) Introduce methods of performance testing and the design of test cases. (3) Display the use of performance testing tools	(1) Be familiar with the use of common performance testing tools. (2) Develop load test scripts, execute tests and analyze the results	8	Lecture, discussion, and experiment	Course Objectives 2, 3
6	**Course content 6. Safety testing** (1) Introduce the methods of safety testing. (2) Introduce the system safety testing process and the use of tools. (3) Display the use of safety testing tools.	(1) Master system safety testing methods, functional security testing, penetration testing methods, and sensitive information security testing methods. (2) Be familiar with the use of common safety testing tools. (3) Execute the test, analyze and summarize the safety test result report.	6	Lecture, discussion, and experiment	Course Objectives 2, 3
7	**Course content 7. Test defense and report writing** (1) Conduct assessment tests on theoretical knowledge such as test principles related to the course content. (2) Summarize the design content, organize relevant data, and write a design report.	(1) Master the basic theoretical knowledge of software testing technology. (2) Be able to summarize and analyze the experimental results, and form a complete and effective design report. (3) Learn to eliminate the problems and failures encountered in the test process by yourself, and have the craftsman spirit of assiduous research.	4	Examination and defense	Course Objectives 2, 3, 4

V Period Distribution and Teaching Modes

V.1 Period Distribution

Name	Contents	Environments	Period	Group size	Attributes （Basic /Comprehensive/ Design/ Creative Study）	Requirements(Compulsory /Optional)
Instruct project design and test plans	Introduce the test principle, test requirements, test plan and test plan. Learn the use of test tools.	Computer	4	1	Comprehensive	Compulsory
Design test cases	Select reasonable test methods to design test cases and write test scripts according to test requirements and test plans.	Computer	4	1	Design	Compulsory
Unit testing	Conduct unit testing for Java and C++ code. The code size is on the order of thousands of lines. Use tools like JUnit, CppUnit, Mocha, etc., and submit test code and test results.	Junit, CppUnit, and Mocha	6	1	Comprehensive	Compulsory
Functional testing	For Windows client, Android/iOS App or Web client, use Selenium, Appium and other suitable functional testing tools to execute test cases, and analyze the results to locate defects.	Selenium and Appium	8	1	Comprehensive	Compulsory
Performance testing	For the Web server, database server or application server, determine the key business, design an appropriate load pattern, perform load testing, and analyze the test results (tables and graphs).	Jmeter	8	1	Comprehensive	Compulsory
Safety	For security test objects,	ZAP and	6	1	Comprehensiv	Compulsory

testing	use appropriate security test tools such as ZAP, DDMS, etc. to perform the test, analyze and summarize the safety test result report.	DDMS			e	
Test defense and report writing	Summarize the defense, and write a design report, including: design objectives, requirements, indicators, block diagrams, selection of components, circuit assembly, debugging results analysis and experience, etc.	Computer	4	1	Comprehensiv e	Compulsory
Total			40			

V.2 Teaching Mode

This course is a very practical course, and the teaching focuses on the cultivation of design ability and practical ability. On the basis of understanding the basic theory, guide students to independently design test schemes, hands-on tests, and finally realize a comprehensive test defect report and test summary report after each link of testing. Task arrangement and test principles are mainly taught in class; at the same time, effective test schemes and test plans are obtained through classroom discussion, and a complete set of test cases are designed; unit testing, functional testing, performance testing and safety testing are practically operated by students. During the test execution process, learn to independently troubleshoot test failures, improve the ability to analyze and solve problems, and thus improve the comprehensive practical ability of testing gradually.

In order to enable students to exert their learning initiative, to further understand the relevant technology culture and technological development trends in the field of software testing applications, and to recognize the application and embodiment of informatization, intelligence, and automation in the development of software testing technology, this course also combines online video supporting teaching. In addition, by issuing questionnaires, understand students' learning situation, make adjustments to teaching methods and answer questions in time.

VI Assessment

The comprehensive assessment of this course includes classroom performance, project design, summary report, and defense examination. Assessment score (Hundred Marking System) = classroom performance * 10% + phase achievement * 50% + summary report * 20% + defense test * 20%.

The final grade is assessed by Five-Levels Marking System: A-Excellent (90 - 100), B-Good

(80 - 89), C-Medium (70 - 79), D-Passing (60 - 69), E-Failure (Below 60).

Assessment Methods or Approaches	Assessment Requirements	Assessment Weighting	Evaluation of Course Objectives
Classroom performance	Attendance, class interaction, and laboratory use specifications	10%	Course Objectives 1, 2, 3, 4
Phase achievement	Test cases, test scripts, and test defect report, etc.	50%	Course Objectives 1, 2, 3, 4
Summary report	Write summary report and submit in time	20%	Course Objectives 1, 2, 3, 4
Testing defense	Defense question	20%	Course Objectives 1, 2, 3, 4

Note: 1. Assessment methods or approaches mainly include classroom performance, phase achievement, summary report, and testing defense, etc.

2. Assessment requirements include class interaction and testing requirements, etc.

3. Assessment Weighting refers to the percentage that assessment methods or approaches take up in the total score.

VII Textbooks and References

1. Textbooks

[1]Ilene Burnstein. Practical Software Testing: A Process-Oriented Approach. Springer, 2006.

2. References

[2]Paul C. Jorgensen. Software Testing: A Craftsman’s Approach. Auerbach Publications, 2013.

[3]Shaomin Zhu. Experiment Tutorial of Software Testing. Tsinghua University Press, 2019.

[4]Paul Ammann. Introduction to Software Testing. Cambridge University Press, 2008.

[5]Abu Sayed Mahfuz. Software Quality Assurance: Integrating Testing, Security, and Audit. Auerbach Publications, 2016.

Written by：Tingting Wu **Reviewed by: Qi Sun**

Date：2021.05.20

Syllabus of Graduation Design (Thesis)

Course Chinese Name: Graduation Design (Thesis)   Course Code: 62550

Course English name:thesesis

Course categories and nature: professional courses, compulsory

Total school hours:320

Credits:8

Pre-courses: All courses required for computer science and technology majors

Object-oriented: Undergraduate in Computer Science and Technology

Starting department: Department of Computer Science and Technology

I. Course teaching objectives

Graduation design analysis and comprehensive practice is in the graduation design, in order to prepare for graduation design and carry out a series of pre-study learning and practice, the purpose is to enable students to understand the whole process of graduation design, understand the different types of graduation design needs to carry out the basic work, master the relevant tools necessary to write graduation thesis, understand the structure and format of the various parts of the graduation thesis, so that students can complete the literature review, translation and other graduation design peripheral work under the guidance of their tutors.

II. the curriculum teaching objectives

Graduation design is the last important link to complete the training program to achieve the goal of undergraduate training, the most comprehensive practical teaching link in the teaching plan, and an important part of the whole teaching plan, which lays a solid foundation for students to analyze and solve practical problems in the field of information in the future.

Course Objective 1: To train students to explore, rigorous reasoning, serious responsibility for work, meticulous, selfless dedication to the country, the collective, love and unity of colleagues, collaboration, things can be painstakingly investigated, with practice to test theory, all-round consideration of problems and other scientific and technological personnel should have the ideological quality and work style.

Goal 2: Develop students' ability to acquire knowledge from literature, scientific experiments, production practices, and research. Improve students' ability to seek problem solving in previous experiences or other disciplines. Develop students' ability to adjust their priorities according to changing conditions.

Course Objective 3:Develop students' ability to work independently on the subject using the knowledge they have learned. To train students to start with demand analysis, design and develop the basic ability of computer software, hardware or application system, and comprehensively cultivate and improve students' technical quality and ability to analyze and solve problems.

Goal 4: To develop students' awareness of society, humanities, environment and so on while giving solutions, so as to lay the foundation for social responsibility.

Goal 5:Develop students' ability to communicate the content of the technical program with their mentor and, in the final defense, to elaborate and answer questions about the content of the graduation thesis.

Course Objective 6:Familiarize yourself with general methods of theoretical research or engineering technical research, evaluate the engineering rationality of the solution, and lay a good foundation for going to work after graduation.

III. the corresponding relationship between the curriculum teaching objectives and graduation requirements

Table 1 How course objectives correspond to graduation requirements

Graduation requirements	Graduation requirements indicator points	Course objectives	Support analysis description
Graduation Requirements 2:Problem Analysis	2.3 Can combine literature research, analyze and demonstrate the factors of complex engineering problems, seek alternative solutions, and realize the diversity of solutions	Course Goal 2	The summary part of graduation design thesis is itself a study of combining literature
Graduation requirements 3: Design/develop solutions	3.4 Ability to consider social, health, safety, legal, cultural and environmental factors in the design process.	Course Goal 4	Solutions are not isolated in society and are bound to have more or less social connections, and solutions should consider social connections to lay the foundation for social responsibility
	3.3 Understanding the current state and trends of the computer frontier reflects a sense of innovation in the design process.	Course Goal 2	Writing a graduation thesis should understand the frontiers and developments in the field involved, so as to lay the foundation for innovation in the design process
Graduation requirements 10:Communication	10.1 Be able to clearly articulate solutions, processes and outcomes to complex computer engineering problems through written reports	Course Goal 5	Graduation thesis communicates with the tutor, and the ability to defend, and enables the tutor and the

	and oral presentations, and understand the challenges and recommendations of industry peers and the general public		defense expert to understand the work they are doing, which in itself is effective communication and communication.
Graduation requirements 11:Project management	11.1 Be able to understand and master engineering management principles and economic decision-making methods.	Course Goal 6	To evaluate the engineering rationality of the solution, we must have knowledge of the principles of engineering management and consider the feasibility of implementation.

IV. The relationship between teaching content and curriculum objectives

(1) The relationship between theoretical teaching content and curriculum objectives(160 hours).

Table 2 Relationship **between theoretical teaching content and curriculum objectives**

serial number	**Contents of teaching**	**Teaching requirements**	**Hours**	**teaching manner**	**The corresponding lesson target**
1	1. Graduation thesis selection and assignment	Understand the content and task requirements of the relevant topics, and retrieve the relevant literature	20	The combination of teacher teaching and self-planning	Course Goal 2
2	2. Open the question	On the basis of understanding the content of the selection of questions, the work to be done on the topics should be written in writing, and the opening questions should be completed	50	Dispersed, mentored	Course goals 1,2,5

serial number	Contents of teaching	Teaching requirements	Hours	teaching manner	The corresponding lesson target
3	3. Graduation design implementation	In strict accordance with the schedule of the topic, under the guidance of the instructor step-by-step implementation, the completion of the task required by the various parts of the content	230	Dispersed, mentored	Course goals 1,3,4,6
4	4. Graduation defense	According to the grade of the college, the graduation design completed the work to defend	20	Dispersed, mentored	Course Goal 5

V. Distribution of course hours and teaching methods

(1) The allocation of course hours

Table 4 Table of Assignments for Theoretical Teaching Content

Teaching methods / Teaching hours / The content of the course	Theory is taught	Exercise lessons	Discussion class	remark	subtotal
1.Graduation thesis selection	20				20
2. Open the question	50				50
3.Graduation design	230				230
4. Graduation defense	20				20
	320				**320**

(2) Teaching methods

The guidance teacher should strictly require the students to collect and consult the necessary materials, and attach importance to the guidance of students' independent work ability, analytical problem-solving ability, innovation ability and design ideas and basic scientific research methods. To guide students to write literature reviews, foreign translations, opening reports, graduation papers, etc., regularly check the progress of students' graduation papers (design), and do a good job in monitoring and recording each stage of the process. Specific graduation papers (designs) work in accordance with the following procedures:

1. Select the question, release the **task book**

Graduation design topic in principle, one person, one question, easy to be appropriate, the weight should be reasonable, the process should be complete, the topic can not be too large or too small, research topics can not be repeated continuously, research topics by the academic committee of the Academy screening, determination. Students can choose according to their own interests, hobbies and expertise, or under the guidance of teachers, allowing students with good grades to raise their own questions, but with the consent of professional guidance teachers, and reported to the college for approval.

2. Open the question

The opening questions include project overview, necessity and significance of the project, current situation and development trends at home and abroad, objectives, project main research content and key technologies, technical routes, main technical indicators, schedule, analysis of research and technology development basis and supporting conditions, economic benefits of expected results, evaluation of the prospects for early application of social benefits, special issues and data retrieval and literature review related to this design project, etc. The focus is on research objectives, the main research content of the project, key technologies and technical routes. The specific requirement is to hold the opening question defense in groups, and the passer-by will move on to the next stage.

3. Design / Paper Process

(1) Progress prosecution

According to the task book issued by the teacher and the opening design made by the students, in strict accordance with the schedule of the topic, under the guidance of the teacher step-by-step implementation. In the event of a progress anomaly in exceptional circumstances, the instructor shall make reasonable adjustments.

(2) Quality prosecution

According to the research content of each stage, the teacher focuses on checking whether the students' stage results meet the design requirements and make reasonable solutions to the existing problems.

(3) Mid-term inspection

In the middle of the research, the college should organize a comprehensive inspection to meet the design requirements of encouragement or praise, there are problems students should be timely reminder, students with serious problems should be given a warning. At the same time, the instructor is tasked with urging students to make timely improvements.

(4) Establish a attendance system

In order to ensure the normal conduct of graduation design, special experimental places should be arranged for students to complete the graduation design, daily attendance, there must be a teacher leave, more than 1 week time should be asked to the college leave.

Students who have serious problems in progress, quality, mid-term inspection and attendance may terminate their graduation design if they are shown a yellow card warning by the hospital. If the teacher did not ask for leave in advance, as a truancy treatment, where random spot checks three times less than, the score is reduced by a grade, the cumulative absenteeism time reached one-third of the whole process, cancel the defense qualification.

(5) Design instructions / paper writing

When students write graduation design (thesis), they are organized, logical, in line with the norms of scientific and technological writing, and in strict accordance with the requirements of the school's undergraduate graduation design (thesis) to write, print and bind. Students must write the full text and abstracts of the paper in Word and print it computerwise in strict accordance with the graduation design paper writing layout specification format. The same should be true of the opening report, foreign translation, etc. The papers that the student finally submits for review need to be checked, and the check rate of more than 15% will be returned for rectification, pending rectification and passing the examination before they can participate in the graduation defense.

4． Defense

（1） Graduation Design (Thesis) Review and Defense is presided over by the College Graduation Defense Committee. The Defence Committee is composed of 5-7 members of the Academy's leadership and experts, each with a Chairman, Deputy Chairman and Secretary. Depending on the needs of its work, the Defence Committee may organize a number of defence teams (not less than 3 persons per group) to be responsible for the specific defence. Members of the defence team must be persons above the instructor (or equivalent).

（2） Instructors should carefully review the student's graduation design (thesis), write the review opinion and sign it, and submit it to the defense committee.

（3） Reviewers review the student's graduation design (thesis) and write comments.

（4） The instructor and the reviewer shall return the student's graduation design (thesis) to the student two days before the student's defense. Students should carefully prepare their defense and draw up an outline of their defense.

（5） Defense is a link of graduation design (thesis), but also graduation design (thesis) assessment method, each college before the defense to determine a unified defense procedures, scoring standards, defense venues, personnel, discipline and other matters.

（6） Each student must go through the defense link to obtain a graduation design (thesis) score. The time for each student's defense is determined by the college. Typically around 30 minutes (wherein: students report no more than 15 minutes, and the respondent teacher asks questions for about 15 minutes). Teachers' questions should be asked key questions around the research content of the subject. In the event of a controversial academic issue, submit it to the Committee for Reply for consultation.

（7） When replying, there should be a record, after the defense is completed, the team leader should sign the record and submit it to the College for preservation and reference.

（8） After the defense, the defense team initially evaluated the graduation design (thesis) results according to the scoring criteria.

Sixth, the course assessment method and evaluation criteria

（1） Grades and methods of assessment of grades

Graduation design results assessment by the guidance teacher's performance, review of teacher's performance, defense results.

Guide teacher's method of evaluating achievement: mainly to the performance and completion of the quality of the thesis during the graduation design period. It includes: work attitude and learning style, discipline and care for laboratory equipment, equipment, access to literature and data collection ability, comprehensive use of theoretical knowledge to solve problems independently, on time to complete the task assigned by the tutor and the completion of the paper. ，

Review the teacher's evaluation results method: mainly to the student paper situation, the completion quality of the thesis and the results, practicality and innovation, graduation design (thesis) workload size, etc. ;

The method of evaluating the results of the defense team: The defense team shall not be less than 5 people, assess the students' understanding of the basic theory, professional knowledge and basic concepts of the profession, comprehensively use the knowledge learned to analyze the ability to solve problems, demonstrate whether it is sufficient, express clearly, whether the thesis has theoretical or practical value.

（2） grade

Instructor rating:30%,Review Teacher rating:30%,Respondent rating:40%. Final rating: excellent, good, medium, pass, fail.

（3） Rating criteria

Excellent: on schedule to successfully complete the projects stipulated in the mission book, can skillfully comprehensive use of the theory and professional knowledge learned, correct theory, calculation, analysis and experiment correct, rigorous, reasonable conclusions, independent work ability is strong, scientific style rigorous, graduation design topic itself is difficult, is the current domestic frontier or rarely reported topics, graduation design has some unique, high level.

Good: on schedule to successfully complete the projects stipulated in the mission book, can better use the theory and professional knowledge learned, correct theory, calculation, analysis, experiment correct, correct experiment, reasonable conclusion, have a certain independent ability to work, good scientific style, the topic itself is moderately difficult, graduation design has a certain level.

Medium: the successful completion of the project specified in the task book on schedule, the use of the theory and professional knowledge is basically correct, but there are shortcomings and shortcomings in non-main content, the argument is correct, calculation, analysis, experiment is basically correct, there is a certain independent ability to work, the topic itself is generally difficult, graduation design level is general.

Pass: with the specific help of the instructor, can complete the task on schedule, the independent work ability is poor and there are some small omissions and omissions, in the application of the theory and professional knowledge, there are no major principled errors, arguments, arguments are basically established, calculation, analysis, experiment is basically correct. Graduation design level basically meets the requirements.

Failure: Projects specified in the mandate were not completed on schedule, or basic essentials and basic skills were not mastered. In the application of theory and professional knowledge there should be undue errors in principle, in the program demonstration, analysis, experiments and other work shows poor ability to work independently, graduation design did not meet the minimum requirements.

7. Recommended reference materials

(i) Reference books

Determine the references according to the requirements of the mandate.

(2) Network resources

Determine network resources according to the requirements of the task book.

Written by: Shen Wei **Reviewed by: Xia Jinsong**

Date: 2021.05.20