

Taizhou University



Self-Assessment Report for ASIIN Programme  
Accreditation for the Bachelor Degree of  
Computer Science and Technology

Appendix D

Syllabus - Engineering Foundations

Department of Computer Science and Technology  
School of Information Engineering

2025

## Contents

Python Programming.....	2
C Programming.....	8
Digital Logic Circuits.....	14
Data Structure.....	18
Database Principles and Applications.....	24
Computer Networks.....	29
Computer Organization and Architecture.....	34
Operating Systems.....	45
Introduction to Computer Science.....	49
Design and Analysis of Algorithms.....	54
Java Programming.....	58
Introduction to Artificial Intelligence.....	63
Principles of Compilers.....	69
Linux Operating System.....	73

## Python Programming

Module designation	Python Programming
Semester(s) in which the module is taught	1 <sup>st</sup> semester
Person responsible for the module	Yuan Hongjuan
Language	Chinese
Relation to curriculum	<p>Python Programming is a core required course for majors including Computer Science and Technology, Internet of Things Engineering, Data Science and Big Data Technology, and Artificial Intelligence. Grounded in real-world application scenarios, this course teaches fundamental Python programming syntax to equip students with essential design methodologies. It cultivates critical thinking skills through problem analysis, abstract modeling, algorithm development, and practical coding solutions, while laying a solid theoretical and practical foundation for advanced courses like C Programming, Data Structures, and Algorithm Design &amp; Analysis. The program also fosters teamwork and innovation capabilities among participants, Develop a standard and detailed, good programming style, improve the level of programming.</p>
Teaching methods	<p>Target students: students of Computer Science and Technology, Internet of Things Engineering, Data Science and Big Data Technology, artificial intelligence</p> <p>Type of teaching: theoretical teaching, experiment teaching</p> <p>Contact hour: 48 hours Including:</p> <p>Theoretical teaching: 32 hours</p> <p>Experiment teaching: 16 hours</p> <p>Computer practice: 0 hours</p> <p>Size of class: 40-60 students</p>

Workload (incl. contact hours, self-study hours)	Total workload = 90 hours Contact hours = 48 hours Self-study hours = 42 hours
Credit points	3
Required and recommended prerequisites for joining the module	None.
Module objectives/intended learning outcomes	<p><b>Learning outcomes:</b></p> <ul style="list-style-type: none"> <li>• <b>Knowledge:</b> <ol style="list-style-type: none"> <li>1. Master the basic syntax of Python language, and complete the basic programming of sequential structure, selection structure and loop structure by combining basic data types and IPO programming methods.</li> <li>2. Master the combination data types of Python, including functions and methods related to lists, tuples, sets, dictionaries and other types, understand the definition and invocation principles of functions, and carry out modular design according to actual needs.</li> <li>3. Master the basic operations of files in Python, especially the processing of CSV format files, understand the basic process of file reading, processing and writing, as well as the changes of data form during the processing of CSV files.</li> <li>4. Master the common standard library and third-party library of Python, understand the random number generation principle of random library, as well as the generation mechanism and principle of Chinese word segmentation jieba library and wordcloud wordcloud library.</li> </ol> </li> <li>• <b>Skill:</b> <ol style="list-style-type: none"> <li>1. Master the execution process and principle of the basic control structure of the program, master the</li> </ol> </li> </ul>

	<p>drawing method of the program design flow chart, be able to read and analyze the basic program, and use the IPO programming method to complete the design, debugging and operation of the basic program.</p> <p>2. Understand the process of function call, be able to use modular design and code reuse ideas, and combine combined data types to complete the design, debugging and operation of complex programs.</p> <p>3. Master the basic steps of file operation, read, process and output CSV format files according to actual needs. Combine combined data types to realize the storage and migration of different forms of data in the process of CSV file processing, as well as statistical analysis of data.</p> <p>4. Master common standard libraries and third-party libraries of Python, and flexibly use random library, jieba library, wordcloud library, etc., to carry out comprehensive programming for Chinese word frequency statistics and graphical display.</p> <p>• <b>Competence:</b></p> <p>1. Master the basic programming methods, use the idea of modular design and code reuse, and complete the tasks of complex program writing and debugging.</p> <p>2. Have good teamwork and exploratory spirit, and cultivate programming ability to further develop applications in the field of big data and artificial intelligence.</p>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Python syntax elements</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● The concept of programming languages</li> <li>● Overview of Python language</li> <li>● Python language development environment configuration</li> <li>● Basic programming methods</li> </ul>

	<ul style="list-style-type: none"> <li>● Analysis of Python syntax elements</li> <li>● Python input and output</li> </ul> <p><b>Chapter 2 Data types, Operators and Expressions</b> (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Definition and naming of variables</li> <li>● Number types and operations</li> <li>● String types and operations</li> <li>● Operators and expressions</li> </ul> <p><b>Chapter 3 Procedure control structure</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Algorithms and flow charts</li> <li>● Select structured programming</li> <li>● Loop structure programming</li> <li>● Exception handling of programs</li> </ul> <p><b>Chapter 4 Combined data types</b> (6 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● List types and operations</li> <li>● Basic operations of strings</li> <li>● Dictionary types and operations</li> <li>● Basic operations on tuples and sets</li> </ul> <p><b>Chapter 5 Functions and Modules</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Function definition and invocation</li> <li>● Parameter passing of functions</li> <li>● Different types of parameters</li> <li>● anonymous function</li> <li>● recursive function</li> <li>● The scope of variables</li> <li>● Module definition and import</li> </ul> <p><b>Chapter 6 File and Data formatting</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Document overview</li> <li>● Read, write and locate files</li> <li>● CSV and formatting and processing of 2d data</li> </ul> <p><b>Chapter 7 Application of Python Standard Library</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● turtle storeroom</li> </ul>
--	---

	<ul style="list-style-type: none"> <li>● random storeroom</li> </ul> <p><b>Chapter 8 Application of Python third-party libraries</b></p> <p>(4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● jieba storeroom</li> <li>● wordcloud library</li> </ul> <p><b>Part B. Experiment teaching</b></p> <p>(16 contact hours; 14 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Experimental program control structure (4 contact hours; 2 self-study hours)</li> <li>2. Function and Module Experiment (4 contact hours; 4 self-study hours)</li> <li>3. Combination data type experiment (4 contact hours; 4 self-study hours)</li> <li>4. Application of files and Python libraries (4 contact hours; 4 self-study hours)</li> </ol>
Examination forms	Closed-book written exam (paperless test with Python123 platform question bank)
Study and examination requirements	<p>Homework should be completed independently by the students after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>Regular grades account for 40%, including homework (10%), experiments (15%) and MOOCs (15%).</p> <p>The final assessment (closed written test) accounts for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Yuan Hongjuan. Python Programming — From Basic to Application [M]. Beijing: China Machine Press, 2023</p> <p><b>2. Reference books</b></p> <p>[1] Song Tian Basic Programming of Python Language [M]. 2nd edition. Beijing: Higher Education Press, 2022.</p> <p>[2] Zhang Li. Python Programming Tutorial [M]. Beijing: Higher Education Press, 2018.</p>

	[3] Jiang Hong. Python Programming and Algorithm Basic Tutorial [M]. Beijing: Higher Education Press, 2019.
Data of last amendment	June 29, 2025



## C Programming

Module designation	C Programming
Semester(s) in which the module is taught	2 <sup>nd</sup> semester
Person responsible for the module	Associate professor Li Li
Language	Chinese
Relation to curriculum	<p>The course "C Programming" aims to cultivate students' logical thinking abilities, enabling them to master programming concepts and acquire foundational knowledge, core concepts, essential skills, and methodologies in computer programming. It develops practical problem-solving capabilities through hands-on computer applications, equipping students with basic C language programming skills. This course lays a solid foundation for subsequent studies and practical implementation in the "Data Structures" course.</p>
Teaching methods	<p>Target students: Computer Science and Technology major students</p> <p>Type of teaching: Theoretical teaching + experimental teaching</p> <p>Contact hour: 64 hours Including:</p> <p>Theoretical teaching: 48 hours</p> <p>Experiment teaching: 16 hours</p> <p>Computer practice: 0 hours</p> <p>Size of class: 40-60 students</p>
Workload (incl. contact hours,	<p>Total workload = 80 hours</p> <p>Contact hours = 64 hours</p>

self-study hours)	Self-study hours = 56 hours
Credit points	4
Required and recommended prerequisites for joining the module	None.
Module objectives/intended learning outcomes	<p><b>Learning outcomes:</b></p> <ul style="list-style-type: none"> <li>• <b>Knowledge:</b> <ul style="list-style-type: none"> <li>● Understand the modular and structured programming ideas;</li> <li>● Master the basic data types, statement formats and function structures of C language</li> </ul> </li> <li>• <b>Skill:</b> <ul style="list-style-type: none"> <li>● Can use C language for structured programming;</li> <li>● Solve some simple application problems;</li> <li>● Use this as a tool to further study some control means and control technologies;</li> </ul> </li> <li>• <b>Competence:</b> <ul style="list-style-type: none"> <li>● Have the ability of dialectical thinking;</li> <li>● Have a scientific and realistic style of study and innovative consciousness and spirit;</li> <li>● Strengthen the awareness of professional ethics and safety norms.</li> </ul> </li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (48 contact hours; 42 self-study hours)</p> <p><b>Chapter 1 Programming and C language</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● What is a computer program</li> <li>● What is a computer language</li> <li>● The development and characteristics of C language</li> <li>● The simplest C program</li> <li>● Steps and methods for running C programs</li> </ul>

	<ul style="list-style-type: none"> <li>● The task of program design</li> </ul> <p><b>Chapter 2 Data types, operators, and expressions</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Data types in C;</li> <li>● Constants in C;</li> <li>● Variables in C;</li> <li>● Operators and expressions;</li> <li>● Data type conversion</li> </ul> <p><b>Chapter 3 sequential structure programming</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Examples of sequential programming</li> <li>● C statements</li> <li>● Data input and output</li> </ul> <p><b>Chapter 4 Selection structure programming</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Select structure and condition judgment</li> <li>● Use if statements to implement selection structures</li> <li>● Relational operators and expressions</li> <li>● Logical operators and logical expressions</li> <li>● Conditional operators and expressions</li> <li>● Select nested structures</li> <li>● Use switch statement to implement multi-branch selection structure</li> <li>● Select a comprehensive example of structural programming</li> </ul> <p><b>Chapter 5 Loop Structure Programming</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Why do we need cycle control</li> <li>● Use a while statement to implement a loop.</li> <li>● Use a do...while statement to implement a loop.</li> <li>● Use a for statement to implement a loop.</li> <li>● Nesting of loops</li> <li>● Comparison of several cycles</li> <li>● Change the state of circular execution</li> <li>● Examples of loop programs</li> </ul> <p><b>Chapter 6 Processing batch data using arrays</b> (6 contact hours; 6 self-study hours)</p>
--	---

	<ul style="list-style-type: none"> <li>● Define and reference a one-dimensional array</li> <li>● Define and reference two-dimensional arrays</li> <li>● Character arrays</li> </ul> <p><b>Chapter 7 Implementing modular programming with functions</b></p> <p>(6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Why use functions</li> <li>● How to define a function</li> <li>● Call the function</li> <li>● Declaration and prototype of the called function</li> <li>● Nested function calls</li> <li>● Recursion of functions</li> <li>● Arrays as function arguments</li> <li>● Local variables and global variables</li> <li>● Storage mode and lifetime of variables</li> <li>● Declaration and definition of variables</li> <li>● Internal functions and external functions</li> </ul> <p><b>Chapter 8: Good use of pointers</b></p> <p>(10 contact hours; 10 self-study hours)</p> <ul style="list-style-type: none"> <li>● What are pointers</li> <li>● Pointer variables</li> <li>● Refer to the array by pointer</li> <li>● Refer to strings by pointer</li> <li>● Pointers to functions</li> <li>● A function that returns the pointer value</li> <li>● Pointer arrays and multiple pointers</li> <li>● Dynamic memory allocation and the pointer variable to it</li> <li>● Summary of relevant pointers</li> </ul> <p><b>Chapter 9: User establishes data type</b></p> <p>(8 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Define and use structure variables</li> <li>● Use a structure array</li> <li>● Structural pointer</li> <li>● Handle the linked list with pointers</li> <li>● Common body type</li> <li>● Use enumeration types</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>● Use typedef to declare new type names</li> <li>● Computer Training 7: Custom data types</li> </ul> <p><b>Chapter 10 Input and Output of Documents</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic knowledge about C files</li> <li>● Open and close files</li> <li>● Read and write data files in sequence</li> <li>● Random read/write data files</li> <li>● Error detection of file read/write</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Sequential structure programming (2 contact hours; 2 self-study hours)</li> <li>2. Selecting structural programming (2 contact hours; 2 self-study hours)</li> <li>3. Loop structure programming (2 contact hours; 2 self-study hours)</li> <li>4. Processing batch data using arrays (4 contact hours; 3 self-study hours)</li> <li>5. Modular programming with functions (4 contact hours; 3 self-study hours)</li> <li>6. Users establish their own data types (2 contact hours; 2 self-study hours)</li> </ol>
Examination forms	Closed-book written exam
Study and examination requirements	<p>Homework should be completed by the students independently after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>Project performance 5%, course work 15%, computer practice 20%.</p> <p>The final assessment (academic plan) accounts for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Zhang Yusheng. C Programming [M]. Shanghai Jiaotong University Press, 2021.</p>

	<p>[2] Zhang Yusheng. C Programming Training tutorial [M]. Shanghai Jiaotong University Press, 2021.</p> <p><b>2. Reference books</b></p> <p>[1] Tan Haoqiang. C Programming [M]. 5th Edition. Beijing: Tsinghua University Press, 2017.</p> <p>[2] Tan Haoqiang. C Programming Study Guide [M]. 5th Edition. Beijing: Tsinghua University Press, 2017.</p> <p>[3] Ivor Horton. Yang Hao (trans.). C Language Introduction Classic [M]. 5th Edition. Beijing: Tsinghua University Press, 2008.</p>
Data of last amendment	June 10, 2025

## Digital Logic Circuits

Module designation	Digital Logic Circuits
Semester(s) in which the module is taught	1 <sup>st</sup> semester
Person responsible for the module	Lecturer Wang Shi
Language	Chinese
Relation to curriculum	<p>The course "Digital Logic Circuits" aims to equip students with essential knowledge in electronic technology and digital system design. It helps students understand and master fundamental concepts, operational principles, and analytical methods of digital logic circuits, while keeping abreast of modern trends and technological frontiers in digital system design. The program equips students with core skills in analyzing and designing digital logic circuits, along with testing techniques. Students will develop independent problem-solving capabilities, enabling them to apply acquired knowledge for basic circuit analysis, design optimization, and improvement of existing circuits. Additionally, the course enhances logical thinking abilities, teamwork spirit, and adaptability to emerging technologies. Innovation ability and systematic theoretical learning and rich practical training enable students to acquire solid professional basic knowledge and strong engineering practice ability, which lays a solid foundation for their future development in computer and related fields.</p>
Teaching methods	<p>Target students: students of Computer Science and Technology, Internet of Things Engineering and Cyberspace Security</p> <p>Type of teaching: theoretical teaching, experiment teaching</p> <p>Contact hour: 48 hours</p> <p>Including:</p> <p>Theoretical teaching: 32 hours</p> <p>Experiment teaching: 16 hours</p> <p>Computer practice: 0 hours</p> <p>Size of class: 40-60 students</p>

Workload (incl. contact hours, self-study hours)	Total workload = 90 hours Contact hours = 48 hours Self-study hours = 42 hours
Credit points	3
Required and recommended prerequisites for joining the module	College Physics, Introduction to Computer Science
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> <li>● <b>Knowledge:</b> <ol style="list-style-type: none"> <li>(1) Master the basic concepts, principles, design principles and methods of digital logic circuits;</li> <li>(2) Understand the whole digital system design process from requirement to final realization and new developments and technologies in the field of digital electronics.</li> </ol> </li> <li>● <b>Skill:</b> <ol style="list-style-type: none"> <li>(1) Be able to analyze the performance characteristics and potential problems of the circuit, and put forward optimization methods by using the knowledge learned;</li> <li>(2) Have the ability to try new ideas and new technologies to solve practical problems.</li> </ol> </li> <li>● <b>Competence:</b> <ol style="list-style-type: none"> <li>(1) Be able to abide by industry norms and maintain an honest and trustworthy professional attitude;</li> <li>(2) Be able to skillfully use simulation tools to complete project work;</li> <li>(3) Hands-on ability and team spirit.</li> </ol> </li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Number system and code system</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● The meaning, research object, characteristics and application of digital circuit</li> <li>● The relationship between different number systems and the law of mutual conversion</li> <li>● Several commonly used BCD codes in digital systems</li> </ul> <p><b>Chapter 2 Fundamentals of Digital Logic</b> (4 contact hours; 4 self-study hours)</p>



	<ul style="list-style-type: none"> <li>● Three basic operations and composite logical operations in Boolean algebra</li> <li>● Logical functions and their representation methods, definition and properties of minimal terms</li> <li>● Basic formulas, laws and rules of Boolean algebra</li> <li>● The method of formula reduction in logical algebra, the significance of reduction and the concept of minimal, examples of formula reduction</li> <li>● Simplification of logical algebra, minimal terms and Kano diagram, simplification of logical functions by Kano diagram, simplification of logical functions with irrelevant terms</li> <li>● Simplification of multi-output logic functions</li> <li>● Transformation of logical function forms</li> </ul> <p><b>Chapter 3 Gate circuits</b> (2 contact hours; 2 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Switching characteristics of crystal diode and transistor</li> <li>2. Basic CMOS logic gate circuit</li> <li>3. Different output structures of CMOS logic gate circuits</li> <li>4. External characteristics of TTL gate circuit</li> </ol> <p><b>Chapter 4 Combinational logic circuits</b> (6 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Characteristics, analysis and design methods of combinational logic circuits</li> <li>● Functions, usage methods and typical application circuit examples of several typical MSI combinational logic circuits (encoder, decoder, adder, numerical comparator, data selector)</li> <li>● Design combinational circuits using typical combinational logic circuit modules</li> </ul> <p><b>Chapter 5 Semiconductor memory</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Classification and structure of memory</li> <li>● Classification of functions and triggering modes of triggers</li> <li>● Circuit structure and working principle of various types of clocks (synchronous, master-slave, edge)</li> <li>● Description of the trigger</li> </ul> <p><b>Chapter 6 Sequential logic circuits</b> (8 contact hours; 8 self-study hours)</p> <ul style="list-style-type: none"> <li>● Characteristics, description and analysis of sequential circuits</li> </ul>
--	---

	<ul style="list-style-type: none"> <li>● Classification of counters</li> <li>● The logic function and usage method of common sequential logic circuit modules (counter, register)</li> <li>● The method of forming an N-ary counter by integrating counters</li> <li>● Design method of synchronous sequential logic circuit</li> </ul> <p><b>Chapter 7 Generation and Shaping of Pulse waveform</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● The significance of pulse signal parameters;</li> <li>● Working principle and function of multivibrator, monostable trigger and Schmitt trigger;</li> <li>● Structure and application of 555 timer.</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Logical operations (4 contact hours, 4 self-study hours)</li> <li>2. Combinational logic circuits (4 contact hours, 4 self-study hours)</li> <li>3. Sequential logic circuit (8 contact hours, 6 self-study hours)</li> </ol>
Examination forms	Closed-written exam
Study and examination requirements	<p>Homework should be completed by the students independently after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>Project performance 5%, course work 15%, computer practice 20%.</p> <p>The final assessment (academic plan) accounts for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Kang Huaguang. Digital Part of Electronic Technology Fundamentals [M]. 6th edition. Beijing: Higher Education Press, 2014.</p> <p>[2] Jia Lixin. Digital Circuits [M]. 3rd edition. Beijing: Electronic Industry Press, 2017.</p> <p>[3] Zhang Zhiliang. Fundamentals of Digital Electronic Technology [M]. Beijing: China Machine Press, 2007</p> <p><b>2. Reference books</b></p> <p>None.</p>
Data of last amendment	June 10, 2025

## Data Structure

Module designation	Data Structure
Semester(s) in which the module is taught	3th semester
Person responsible for the module	Jiang Min
Language	Chinese
Relation to curriculum	The prerequisites of this course are Introduction to Computer Science, C Programming and Discrete Mathematics; the follow-up courses are mainly Operating System, Database Principles and Applications, and Fundamentals of Software Engineering.
Teaching methods	<p>Target students: Computer Science and Technology major students</p> <p>Type of teaching: Theoretical teaching and experimental teaching</p> <p>Contact hour: 80 hours Including:</p> <p>Theoretical teaching: 48 hours</p> <p>Experiment teaching: 32 hours</p> <p>Computer practice: 0 hours</p> <p>Size of class: 40-60 students</p>
Workload (incl. contact hours, self-study hours)	<p>Total workload = 150</p> <p>Contact hours = 80</p> <p>Self-study hours = 70</p>
Credit points	5
Required and recommended prerequisites for joining the module	Introduction to Computer Science, C Programming and Discrete Mathematics
Module objectives/intended learning outcomes	<p><b>Learning outcomes:</b></p> <ul style="list-style-type: none"> <li>• <b>Knowledge:</b></li> </ul> <p>Master the basic concepts, logical characteristics,</p>

	<p>storage representation, algorithm description and basic algorithm application of various data structures..</p> <ul style="list-style-type: none"> <li>• <b>Skill:</b></li> </ul> <p>Have the ability to apply theoretical knowledge to build models to solve practical engineering problems, design algorithms and evaluate algorithms; have the ability to correctly design relevant algorithms according to requirements, rationally design the logical structure of data, and effectively store and process data.</p> <ul style="list-style-type: none"> <li>• <b>Competence:</b></li> </ul> <p>Have the awareness of lifelong learning, uphold the spirit of inquiry and innovation in the process of analysis and evaluation algorithm, and maintain a rigorous and realistic scientific attitude.</p>
Content	<p><b>Part A. Theoretical teaching</b> (48 contact hours; 42 self-study hours)</p> <p><b>Chapter 1 Introduction</b> (3 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● The development of data structures and their place in computer science</li> <li>● Basic concepts and terms, abstract data types</li> <li>● Algorithm description and algorithm analysis.</li> </ul> <p><b>Chapter 2 Linear Table</b> (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> <li>● Definition of linear list, related concepts (prior, successor, length, bit order, empty list, etc.), definition of abstract data structure types of linear list</li> <li>● The sequential representation and operation implementation of the linear list; the implementation method of the insertion and deletion operations of the sequential list</li> <li>● Related concepts of linear linked list: head node, node, data field, pointer field, pointer (chain); storage structure of linear linked list, implementation method of insertion and deletion</li> </ul>

	<p>operations and time complexity analysis</p> <ul style="list-style-type: none"> <li>● The storage structure of circular linked list, the implementation method of insertion and deletion operations and the time complexity analysis.</li> </ul> <p><b>Chapter 3 Stack and Queue</b> (9 contact hours; 8 self-study hours)</p> <ul style="list-style-type: none"> <li>● Stack related concepts: stack, stack top, stack bottom, empty stack definition</li> <li>● Stack "last in, first out" feature</li> <li>● Stack abstract data type definition</li> <li>● The representation and implementation of the stack: sequential stack and chain stack</li> <li>● Examples of stack applications: number conversion, bracket matching verification, expression evaluation (application of stack to implement the process of expression evaluation)</li> <li>● Definition of queue and related concepts, the "first come first served" feature of queue, the definition of abstract data types of queue</li> <li>● Linked queue--linked representation and implementation of a queue</li> <li>● Circular queue--the order representation and implementation of the queue: the storage structure of the circular queue, the implementation of the insertion and deletion operations of the circular queue</li> </ul> <p><b>Chapter 4 Trees and Binary Trees</b> (9 contact hours; 9 self-study hours)</p> <ul style="list-style-type: none"> <li>● Abstract data type definitions of trees. Definitions of terms such as nodes, degree of nodes, leaves (terminal nodes), branch nodes (non-terminal nodes), tree degree, parent, child, sibling, ancestor, descendant, depth, ordered tree, unordered tree, and forest</li> <li>● Abstract data type definition of binary tree</li> <li>● The 5 properties of a binary tree; the concept of a full binary tree and a complete binary tree</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>● Storage structure of binary tree: sequential storage structure, chain storage structure</li> <li>● Breadth-first binary tree: the concept and recursive algorithm of pre-order traversal, middle-order traversal, and post-order traversal</li> <li>● The storage structure of the tree: parent representation, child representation, child sibling representation</li> <li>● A method for converting a tree or forest to a binary tree</li> <li>● Huffman tree and its applications: path length, tree path length, weighted path length of a tree, definition of optimal binary tree (Huffman tree); Huffman coding algorithm</li> </ul> <p><b>Chapter 5 Graph</b> (9 contact hours; 8 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts and terms: vertex, arc (arc head, arc tail), directed graph, undirected graph, complete graph, network, subgraph, degree (outdegree, indegree), path, loop (ring), simple loop, connected graph (strongly connected graph), connected component (strongly connected component), etc.</li> <li>● The storage structure of the graph: adjacency matrix and adjacency list</li> <li>● Graph traversal: depth first, breadth first</li> <li>● Connectivity problem of a graph: definition of minimum generation tree; algorithms for constructing minimum generation tree (Prim algorithm and Kruskal algorithm)</li> <li>● Applications of graphs: topological sorting, reconnected graph and key nodes, critical path and single source shortest path problem (Dijkstra algorithm)</li> </ul> <p><b>Chapter 6 Search</b> (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> <li>● The search process, algorithm and performance</li> </ul>
--	--

	<p>analysis of sequential search; the definition of average search length</p> <ul style="list-style-type: none"> <li>● Half search process, algorithm and performance analysis of ordered list</li> <li>● The search process, algorithm and performance analysis of index sequence table</li> <li>● Definition and search process of binary sorting tree; insertion and deletion of binary sorting tree; analysis of search in binary sorting tree; concept and construction process of balanced binary tree (AVL tree)</li> <li>● Hash Table: Basic concepts: hash (Hash) function, collision, synonyms, hash table, hash, hash address (hash address), etc.; construction requirements of hash function, division and remainder method; methods to deal with collisions: open address method, chain address method; search and analysis of hash table</li> </ul> <p><b>Chapter 7 Internal sequencing</b> (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> <li>● The definition of sorting; the time and space characteristics and stability of sorting methods</li> <li>● Insertion sort: direct insertion sort and Hill sorting algorithm</li> <li>● Quicksort algorithm</li> <li>● Selection sort: heap sort algorithm</li> <li>● Merge sort algorithm</li> <li>● Comparison of various internal sorting methods</li> </ul> <p><b>Part B. Experiment teaching</b> (32 contact hours; 28 self-study hours)</p> <ul style="list-style-type: none"> <li>● Experiment 1: a comprehensive experiment on stacks and queues. (8 contact hours; 7 self-study hours)</li> <li>● Experiment 2: Comprehensive experiment on trees and graphs. (8 contact hours; 7 self-study hours)</li> <li>● Experiment 3: Comprehensive experiment of</li> </ul>
--	--

	<p>search. (8 contact hours; 7 self-study hours)</p> <ul style="list-style-type: none"> <li>● Experiment 4: Comprehensive experiment on internal sorting. (8 contact hours; 7 self-study hours)</li> </ul>
Examination forms	Closed-book written exam
Study and examination requirements	<p>Homework should be completed by the students independently after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>The usual performance is 10%, course work is 10%, course experiment is 10%, and online learning is 10%. The final written test accounts for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Yan Weimin, Li Dongmei, Wu Weimin. Data Structures [M]. C Language Edition, 3rd Edition with Micro-lecture Videos. Beijing: People's Posts and Telecommunications Press, 2024.</p> <p><b>2. Reference books</b></p> <p>[1] Li Dongmei, Zhang Qi. Data Structure Exercise Analysis and Experiment Guide [M]. Beijing: People's Posts and Telecommunications Press, 2017.</p> <p>[2] Yan Weimin, Wu Weimin, Data Structures [M]. C Language Edition. Beijing: Tsinghua University Press, 2019.</p> <p>[3] Mark Allen Weiss. Data Structures and Algorithm Analysis (C++ Description) [M]. Beijing: Publishing House of Electronics Industry, 2017.</p>
Data of last amendment	June 10, 2025



## Database Principles and Applications

Module designation	Database Principles and Applications
Semester(s) in which the module is taught	3 <sup>rd</sup> semester
Person responsible for the module	Associate professor Zhou Aiping
Language	Chinese
Relation to curriculum	This foundational course in Computer Science and Technology plays a vital role in developing students' professional competencies. Through this program, learners will master core concepts including data models, relational databases, SQL (the standard language for relational databases), relational data theory, and database design principles. Students will acquire essential technical skills such as database security, integrity management, recovery techniques, and concurrency control mechanisms. The curriculum cultivates professional ethics, a craftsman's dedication to excellence, and an innovative spirit of exploration. It also equips students with practical skills in developing database applications using object-oriented programming languages and SQL Server, laying a solid foundation for advanced courses. Practical theoretical basis and operational skills.
Teaching methods	<p>Target students: students of Computer Science and Technology.</p> <p>Type of teaching: theoretical teaching, experiment teaching</p> <p>Contact hour: 64 hours</p> <p>Including:</p> <p>Theoretical teaching: 48 hours</p> <p>Experiment teaching: 16 hours</p> <p>Computer practice: 0 hours</p> <p>Size of class: 40-60 students</p>
Workload (incl. contact hours, self-study hours)	<p>Total workload = 120 hours</p> <p>Contact hours = 64 hours</p> <p>Self-study hours = 56 hours</p>

Credit points	4
Required and recommended prerequisites for joining the module	Programming, data structures, computer networks
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> <li>● <b>Knowledge:</b> Understand the history of data management technology, be familiar with the basic concepts of database, understand the basic principles of data model, relational database, SQL language, database design, etc., master the database security, integrity, recovery and concurrency control and other technologies.</li> <li>● <b>Skill:</b> Can operate on a database management system with SQL language, can design database mode and develop database application system, can select appropriate technology to solve the security, integrity, fault and concurrency control of the database for specific problems.</li> <li>● <b>Competence:</b> 1. Have the spirit of paying attention to the forefront of the discipline and deeply exploring the new technology in the field of database. 2. Ability to work in a team, coordinate and organize.</li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (48 contact hours; 42 self-study hours)</p> <p><b>Chapter 1 Introduction</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Database system overview;</li> <li>● Data model;</li> <li>● The structure of the database system;</li> <li>● Composition of database system.</li> </ul> <p><b>Chapter 2 Relational Database</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Relational data structures and their formal definitions;</li> <li>● Relationship operations;</li> <li>● The integrity of relationships;</li> <li>● Relational algebra.</li> </ul> <p><b>Chapter 3 relational database standard language</b></p>

	<p><b>SQL</b> (8 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● SQL summary;</li> <li>● Student-course database;</li> <li>● Data definition;</li> <li>● Query and pivot;</li> <li>● Data updates;</li> <li>● Treatment of null values;</li> <li>● View.</li> </ul> <p><b>Chapter 4 Database Security</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Database security overview;</li> <li>● Database security control;</li> <li>● Viewing mechanism;</li> <li>● SQL Server security technology.</li> </ul> <p><b>Chapter 5 Database Integrity</b> (4 contact hours; 4 self-study hours)</p> <p>Entity integrity</p> <ul style="list-style-type: none"> <li>● Reference integrity;</li> <li>● User-defined integrity;</li> <li>● Completeness constraints on clauses;</li> <li>● Say with certainty;</li> <li>● Trigger</li> </ul> <p><b>Chapter 6 Relational Data Theory</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● The formulation of questions;</li> <li>● Normalization;</li> <li>● Axiomatic systems that depend on data.</li> </ul> <p><b>Chapter 7 Database Design</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Database design overview;</li> <li>● Requirement analysis;</li> <li>● Concept structure design;</li> <li>● Logical structure design;</li> <li>● Physical design of database;</li> <li>● Implementation and maintenance of the database.</li> </ul> <p><b>Chapter 8 Database Programming</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Flushbonading SQL;</li> <li>● Proceduring SQL;</li> <li>● Storage process and function;</li> <li>● ODBC programming</li> </ul>
--	---

	<p><b>Chapter 9 Relationship query processing and query optimization</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Query processing of relational database system;</li> <li>● Query optimization of relational database system;</li> <li>● Algebraic optimization;</li> <li>● Physical optimization.</li> </ul> <p><b>Chapter 10 Database Recovery Technology</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of transactions;</li> <li>● Database recovery technology;</li> <li>● Types of faults;</li> <li>● Implementation techniques for recovery;</li> <li>● Recovery policy;</li> <li>● Recovery techniques with checkpoints;</li> <li>● Database mirroring.</li> </ul> <p><b>Chapter 11 Concurrency control</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Overview of concurrency control;</li> <li>● Blockade;</li> <li>● Locking protocol;</li> <li>● Live lock and dead lock;</li> <li>● Serializability of concurrent scheduling;</li> <li>● Two lock protocols;</li> <li>● The granularity of the block;</li> <li>● The isolation level of SQL Server.</li> </ul> <p><b>Chapter 12 Review</b> (2 contact hours; 2 self-study hours)</p> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Creation and management of database (4 contact hours; 4 self-study hours)</li> <li>2. Relational Database Standard Language SQL (6 contact hours; 4 self-study hours)</li> <li>3. Database security and integrity (2 contact hours; 2 self-study hours)</li> <li>4. Database Programming (4 contact hours; 4 self-study hours)</li> </ol>
Examination forms	Closed-book written exam
Study and examination	Homework should be completed by the students independently after each lesson.

requirements	<p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>The grade consists of the regular grade (40%) and the final grade (60%).</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Wang Shan. Database System Introduction [M]. 6th Edition. Beijing: Higher Education Press, 2023.</p> <p><b>2. Reference books</b></p> <p>[1] Li Yuejun. Database Principles and Applications (MySQL Edition) [M]. Beijing: Tsinghua University Press, 2023.</p>
Data of last amendment	June 10, 2025

## Computer Networks

Module designation	Computer Networks
Semester(s) in which the module is taught	3th semester
Person responsible for the module	Professor Shuai Xiaoying
Language	Chinese
Relation to curriculum	Through this course, students will enhance their ability to analyze and resolve network issues by combining theoretical knowledge with practical application, thereby laying a solid foundation for further learning and research. The curriculum integrates theory with practice, cultivating students' capacity to apply knowledge in real-world scenarios, logical reasoning skills, and collaborative innovation capabilities. It emphasizes understanding and respecting the spirit of craftsmanship, encouraging learners to actively embody this ethos while developing strong scientific literacy. Additionally, the program guides students in shaping their career aspirations and fosters recognition of socialist core values through hands-on engagement.
Teaching methods	Target students: students of Computer Science and Technology, Internet of Things Engineering, Data Science and Big Data Technology, Cyberspace Security Type of teaching: theoretical teaching, experiment teaching Contact hour: 64 hours Including: Theoretical teaching: 48 hours Experiment teaching: 16 hours Computer practice: 0 hours Size of class: 40-60 students
Workload (incl. contact hours, self-study hours)	Total workload = 120 hours Contact hours = 64 hours Self-study hours = 56 hours
Credit points	4
Required and	Programming, Data Structure.

recommended prerequisites for joining the module	
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> <li>● <b>Knowledge:</b> Master network architecture and typical network protocols, understand the basic principles of network communication and network switching equipment.</li> <li>● <b>Skill:</b> Can analyze the basic process of network data transmission, design small network topology structure, use switches and routers to achieve network interconnection, can use network knowledge and technology to solve network application problems.</li> <li>● <b>Competence:</b> Pay attention to the development trend and application prospect of the discipline, and have the spirit of exploration of new technology, craftsman spirit and social responsibility.</li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (48 contact hours; 42 self-study hours)</p> <p><b>Chapter 1 Introduction</b> (6 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Definition of computer network</li> <li>● The function, type and composition of computer networks</li> <li>● Network architecture <ul style="list-style-type: none"> <li>(1) OSI reference model</li> <li>(2) TCP/IP model</li> </ul> </li> <li>● Main performance indicators of the network <ul style="list-style-type: none"> <li>(1) Delay</li> <li>(2) Throughput</li> <li>(3) Bandwidth</li> </ul> </li> </ul> <p><b>Chapter 2 Physical Layer</b> (8 contact hours, 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of the physical layer</li> <li>● Basic knowledge of data communication</li> <li>● The transport media below the physical layer</li> <li>● Broadband access technology</li> </ul> <p><b>Chapter 3 Data link layer</b> (8 contact hours, 6 self-study hours)</p>

	<ul style="list-style-type: none"> <li>● Basic concepts of the data link layer               <ul style="list-style-type: none"> <li>(1) Link</li> <li>(2) Data link layer services</li> </ul> </li> <li>● Link control               <ul style="list-style-type: none"> <li>(1) Frame format</li> <li>(2) Error control</li> </ul> </li> <li>● Multiple access               <ul style="list-style-type: none"> <li>(1) Static channel access</li> <li>(2) Random channel access</li> <li>(3) Alternating channel access</li> </ul> </li> <li>● Ethernet               <ul style="list-style-type: none"> <li>(1) Ethernet frame format</li> <li>(2) CSMA/CD protocol</li> <li>(3) Address resolution protocol</li> </ul> </li> <li>● Switching basics</li> <li>● Experiment Project 2 Network interconnection</li> </ul> <p><b>Chapter 4 Network Layer</b> (10 contact hours, 10 self-study hours)</p> <ul style="list-style-type: none"> <li>● Two services provided by the network layer</li> <li>● IP protocol               <ul style="list-style-type: none"> <li>(1) IPv4 addressing</li> <li>(2) Subnet division</li> <li>(3) IPv4 datagram format</li> <li>(4) IP datagram forwarding process</li> </ul> </li> <li>● ICMP protocol</li> <li>● Routing protocol               <ul style="list-style-type: none"> <li>(1) Router basics</li> <li>(2) RIP protocol</li> <li>(3) OSPF protocol</li> </ul> </li> <li>● Experiment Project 2 Network interconnection</li> </ul> <p><b>Chapter 5 Transport Layer</b> (10 contact hours, 10 self-study hours)</p> <ul style="list-style-type: none"> <li>● Functions, services and protocols of the transport layer</li> <li>● User Datagram Protocol (UDP)</li> <li>● Transmission control protocol TCP               <ul style="list-style-type: none"> <li>(1) TCP header format</li> <li>(2) TCP connection</li> </ul> </li> <li>● How reliable transmission works</li> <li>● TCP flow control</li> <li>● TCP congestion control</li> <li>● Experimental Project 1: Network Application and</li> </ul>
--	---



	<p>Analysis</p> <p><b>Chapter 6 Application layer</b> (6 contact hours, 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Overview <ul style="list-style-type: none"> <li>(1) C/S service model</li> <li>(2) P2P service model</li> </ul> </li> <li>● DNS <ul style="list-style-type: none"> <li>(1) Working principle of DNS</li> <li>(2) Domain name structure</li> </ul> </li> <li>● World Wide Web WWW <ul style="list-style-type: none"> <li>(1) Overview of the World Wide Web</li> <li>(2) Hypertext Transfer Protocol HTTP</li> </ul> </li> <li>● File Transfer Protocol FTP <ul style="list-style-type: none"> <li>(1) Basic working principle of FTP</li> </ul> </li> <li>● Email <ul style="list-style-type: none"> <li>(1) Simple Mail Transfer Protocol SMTP</li> <li>(2) Email reading protocol</li> </ul> </li> <li>● Experimental Project 1: Network Application and Analysis</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Network application and analysis (8 contact hours; 7 self-study hours)</li> <li>2. Internet connectivity (8 contact hours; 7 self-study hours)</li> </ol>
Examination forms	Closed-book written exam
Study and examination requirements	<p>Homework should be completed by the students independently after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>Project performance 5%, course work 15%, computer practice 20%.</p> <p>The final assessment (academic plan) accounts for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>Shuai Xiaoying. Computer Network [M]. 1st edition. Beijing: China University of Science and Technology Press, 2017.</p> <p><b>2. Reference books</b></p> <p>[1] Xie Xiren. Computer Network [M]. 8th edition. Beijing: Electronic Industry Press, 2021.</p> <p>[2] Andrew S. Tanenbaum, Computer Networks [M]. translated by Pan Aimin. 6th edition. Beijing: Tsinghua</p>

	University Press, 2022. [3] Ruijie Network. Internet Interconnection and Implementation [M]. 1st edition. Beijing: Beijing Hope Electronic Publishing House, 2005.
Data of last amendment	June 10, 2025

## Computer Organization and Architecture

Module designation	Computer Organization and Architecture
Semester(s) in which the module is taught	4 <sup>th</sup> semester
Person responsible for the module	Lecturer Xie Danyan
Language	Chinese
Relation to curriculum	This course takes "Introduction to Computer", "Digital Logic Circuit" and other courses as the prerequisite, and "Operating system", "Linux operating system", "Intelligent application development" and other courses as the follow-up courses.
Teaching methods	<p>Target students: students of Computer Science, Internet of Things Engineering, Data Science and Big Data Technology, Cyberspace Security</p> <p>Type of teaching: theoretical teaching, experiment teaching</p> <p>Contact hour: hours</p> <p>Including:</p> <p>Theoretical teaching: 48 hours</p> <p>Experiment teaching: 32 hours</p> <p>Computer practice: 0 hours</p> <p>Size of class: 40-60 students</p>
Workload (incl. contact hours, self-study hours)	<p>Total workload = 150 hours</p> <p>Contact hours = 80 hours</p> <p>Self-study hours = 70 hours</p>
Credit points	5
Required and recommended prerequisites for joining the module	None.
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> <li>● <b>Knowledge:</b></li> </ul> <p>Be able to elaborate the working principle of von Neumann computer, explain the structure and working principle of memory, arithmetic unit, instruction system, controller and I/O system, establish the system view of</p>

	<p>software and hardware coordination, and be able to use the above theory to express the structure and working principle of model system.</p> <ul style="list-style-type: none"> <li>● <b>Skill:</b> Be able to analyze and apply computer system hardware by means of literature research, reasoning and experiments based on the working principle of Von Neumann computer, and be familiar with common hardware design techniques and rules.</li> <li>● <b>Competence:</b> Have the spirit of cooperation, the ability to coordinate work and organize management, be able to pay attention to the development trend and application prospect of the subject, and have the spirit of exploring new technologies.</li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (48 contact hours; 42 self-study hours)</p> <p><b>Chapter 1 Introduction to Computer Systems</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Introduction to computer systems The concept of computer hardware and software The hierarchy of computer systems Computer composition and computer architecture</li> <li>● The basic composition of a computer Characteristics of von Neumann computer The hardware block diagram and working process of the computer</li> <li>● The main technical indicators of computer hardware, including machine word length, storage capacity, operation speed.</li> </ul> <p><b>Chapter 2 System Bus</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of the bus</li> <li>● Classification of bus (1) In-chip bus (2) System bus (3) Communication bus</li> <li>● Bus characteristics and performance indicators (1) Bus characteristics (2) Bus performance indicators (3) Bus standard</li> </ul>

	<ul style="list-style-type: none"> <li>● Bus configuration <ul style="list-style-type: none"> <li>(1) Single bus structure</li> <li>(2) Multi-bus structure</li> <li>(3) Examples of bus structure</li> </ul> </li> <li>● Bus control <ul style="list-style-type: none"> <li>(1) Bus judgment and control</li> <li>(2) Bus communication control</li> </ul> </li> </ul> <p><b>Chapter 3 Storage</b> (8 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Overview of storage <ul style="list-style-type: none"> <li>(1) Classification of memory</li> <li>(2) The hierarchy of memory</li> </ul> </li> <li>● Main memory <ul style="list-style-type: none"> <li>(1) Semiconductor memory chip</li> <li>(2) Random access memory</li> <li>(3) Read only memory</li> <li>(4) The connection between memory and CPU</li> <li>(5) Measures to improve the speed of memory access</li> </ul> </li> <li>● Cache <ul style="list-style-type: none"> <li>(1) The basic structure and working principle of Cache</li> <li>(2) Cache--main memory address mapping</li> <li>(3) Replacement algorithm</li> </ul> </li> </ul> <p><b>Chapter 4 Input and Output System</b> (6 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Summary <ul style="list-style-type: none"> <li>(1) Development overview of input-output system</li> <li>(2) Composition of input-output system</li> <li>(3) I/O and the host's addressing mode, transmission mode, contact mode and device addressing</li> </ul> </li> <li>● I/O equipment <ul style="list-style-type: none"> <li>(1) Input devices</li> <li>(2) Output device</li> <li>(3) Other I/O devices</li> <li>(4) Multimedia technology</li> </ul> </li> <li>● I/O joggle <ul style="list-style-type: none"> <li>(1) Functions and composition of I/O interface</li> <li>(2) Interface type</li> </ul> </li> <li>● Program query mode <ul style="list-style-type: none"> <li>(1) The process of program query</li> <li>(2) Working principle and interface circuit of program</li> </ul> </li> </ul>
--	--

	<p>query mode</p> <ul style="list-style-type: none"> <li>● Program interrupt mode <ul style="list-style-type: none"> <li>(1) The generation of I/O interrupt</li> <li>(2) Working principle of program interrupt mode and program interrupt interface circuit</li> <li>(3) I/O interrupt service processing process</li> <li>(4) The process of interrupting the service program</li> </ul> </li> <li>● DMA way <ul style="list-style-type: none"> <li>(1) Characteristics of DMA mode</li> <li>(2) The function and composition of DMA interface circuit</li> <li>(3) The working process of DMA</li> </ul> </li> </ul> <p><b>Chapter 5 Computer Operation Methods</b> (10 contact hours; 10 self-study hours)</p> <ul style="list-style-type: none"> <li>● Unsigned numbers and signed numbers</li> <li>● Fixed-point and floating-point representation of numbers <ul style="list-style-type: none"> <li>(1) Fixed-point representation</li> <li>(2) Floating point representation</li> <li>(3) Comparison between fixed point number and floating point number</li> </ul> </li> <li>● Fixed point arithmetic <ul style="list-style-type: none"> <li>(1) Shift operation</li> <li>(2) Addition and subtraction</li> <li>(3) Multiplication</li> </ul> </li> <li>● Floating point arithmetic <ul style="list-style-type: none"> <li>(1) Floating-point addition and subtraction</li> <li>(2) Floating-point multiplication</li> <li>(3) Hardware configuration required for floating-point operation</li> </ul> </li> <li>● ALU <ul style="list-style-type: none"> <li>(1) ALU circuit</li> <li>(2) Fast carry chain</li> </ul> </li> </ul> <p><b>Chapter 6 Instruction System</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Machine instruction <ul style="list-style-type: none"> <li>(1) The general format of instructions</li> <li>(2) Instruction word length</li> </ul> </li> <li>● Operation type and operation type <ul style="list-style-type: none"> <li>(1) Operator type</li> <li>(2) The storage mode of data in memory</li> <li>(3) Operation type</li> </ul> </li> </ul>
--	---

	<ul style="list-style-type: none"> <li>● AM <ul style="list-style-type: none"> <li>(1) Instruction addressing</li> <li>(2) Data addressing</li> </ul> </li> <li>● Example of instruction format <ul style="list-style-type: none"> <li>(1) Various factors to be considered in the design instruction</li> <li>(2) Examples of instruction format</li> <li>(3) Example of instruction format design</li> </ul> </li> <li>● RISC technology <ul style="list-style-type: none"> <li>(1) The emergence and development of RISC technology</li> <li>(2) Main features of RISC</li> </ul> </li> </ul> <p><b>Chapter 7 Structure and Function of CPU</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● The structure of the CPU <ul style="list-style-type: none"> <li>(1) Functions of CPU</li> <li>(2) The structure block diagram of the CPU</li> <li>(3) CPU registers</li> <li>(4) Control unit and interrupt system</li> </ul> </li> <li>● Instruction cycle <ul style="list-style-type: none"> <li>(1) The basic concept of instruction cycle</li> <li>(2) Data flow in the instruction cycle</li> </ul> </li> <li>● Directive flow <ul style="list-style-type: none"> <li>(1) The principle and influencing factors of instruction flow</li> <li>(2) Pipeline performance</li> </ul> </li> <li>● Interrupt system <ul style="list-style-type: none"> <li>(1) Interrupt request mark and interrupt judgment</li> <li>(2) Finding the entry address of the service program interrupt</li> <li>(3) Interrupt response</li> <li>(4) Protect and restore the site</li> <li>(5) Interrupt shielding technology</li> </ul> </li> </ul> <p><b>Chapter 8 Functions of control unit</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Analysis of microprocessor commands <ul style="list-style-type: none"> <li>(1) Instruction fetch cycle</li> <li>(2) Address cycle</li> <li>(3) Execution cycle</li> <li>(4) Interrupt cycle</li> </ul> </li> <li>● Function of the control unit <ul style="list-style-type: none"> <li>(1) External characteristics of the control unit</li> </ul> </li> </ul>
--	--

	<p>(2) Examples of control signals</p> <p><b>Chapter 9 Design of control unit</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Combinational logic design <ul style="list-style-type: none"> <li>(1) Block diagram of combinational logic control unit</li> <li>(2) Micro-operation beat arrangement</li> <li>(3) Combination logic design steps</li> </ul> </li> <li>● Microcoding <ul style="list-style-type: none"> <li>(1) The emergence of microprogram design idea</li> <li>(2) Block diagram and working principle of microprogram control unit</li> <li>(3) The encoding mode of micro-instructions</li> <li>(4) The formation of microinstruction address</li> <li>(5) Microinstruction format</li> </ul> </li> </ul> <p><b>Part B. Experiment teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Experiment 1 Register experiment</b> (4 contact hours, 2 self-study hours) Data is written into four registers in the register file, making R0=55H, R1=AAH, R2=0FH, and R3=F0H. R0, R1, R2 and R3 are read from the register stack respectively to verify the correctness of the written data.</p> <p><b>Experiment 2: Calculator experiment</b> (4 contact hours, 2 self-study hours) Verify the arithmetic and logical functions of the operator.</p> <p><b>Experiment 3 Storage experiment</b> (4 contact hours, 4 self-study hours) According to the table, connect the relevant control signal and the binary switch, check carefully, and then turn on the power supply. Set the digital switches SW0-SW7 (with SW0 being the least significant bit) to 00H, then store this data as an address in AR1. Next, reset the binary switch control and write the number 00H from the digital switches into RAM unit 0. Following this procedure, store data 10H in memory unit 10H, 20H in unit 20H, 30H in unit 30H, and 40H in unit 40H, totaling five data entries. Using the left port of the dual-port memory, sequentially read out the contents from memory cells 00H, 10H, 20H, 30H, and 40H. Verify whether the content in each cell matches its corresponding address number.</p>
--	---



	<p>Record the data. Note: Multiple components are prohibited from simultaneously transmitting data to the bus. When performing read operations, SW_BUS tri-state gates must be disabled. Additionally, when addressing AR1, the dual-port memory cannot be selected.</p> <p>Through the right port (instruction port) of the dual-port memory, the contents of the 00H,10H,20H,30H and 40H units of the memory are sequentially placed into the instruction register IR, and the observation result is checked whether it is the same as (2), and the data is recorded.</p> <p><b>Experiment 4: Composition of data pathway</b> (4 contact hours, 4 self-study hours)</p> <p>Be familiar with the signals required by each module.</p> <p>Using an 8-bit data switch, the following values are programmed into four general-purpose registers in the RF: R0=0FH, R1=0F0H, R2=55H, and R3=0AAH. The procedure for setting R0 to 0FH involves: First, using switches SW0-SW7 to set 0FH in ER with WR1=0, WR0=0, and WRD=1, then transferring ER's data to RF. The programming process for other registers follows a similar sequence.</p> <p>Data from R0 to R3 is simultaneously loaded into the DR2 register and DBUS. The system monitors whether the data matches the values stored in R0 to R3, then records the results. Data on DBUS can be directly displayed via indicator lights, while data from DR2 is routed through the Arithmetic Logic Unit (ALU) to DBUS via a direct path.</p> <p>Send an address of 0FH to AR1 using 8-bit digital switches SW0-SW7, then write the 0FH from R0 into the dual-port RAM. In the same way, sequentially write the data from R1 to R3 into the RAM cells 0F0H,55H, and 0AAH.</p> <p>Write the data of 0AAH unit into R0, 55H unit into R1, 0F0H unit into R2, and 0FH unit into R3 respectively. Then read out the data from R3, R2, R1, and R0 onto DBUS, verify whether the read data is correct through the indicator light, and record the data.</p> <p>Conduct parallel input/output (PI/O) tests using the RS protocol. 1) Select port B (RS) as R0, port A (RD) as R1,</p>
--	--

	<p>and port WR as R2. Set WRD to 1 and observe the parallel I/O results. Switch RS to R2 to verify the write operation's effectiveness, then record the data. 2) Keep both RS (B) and WR ports at R2 with WRD=1 while inserting new data into ER. Observe the I/O results: Does the RS port output the original data or the newly inserted data?</p> <p>What faults are found in the process of data transmission? How to overcome them?</p> <p><b>Experiment 5 Composition of conventional microprocessor controller</b> (4 contact hours, 4 self-study hours)</p> <p>Familiarize yourself with the definition of microinstruction formats. According to this definition, the eight microinstructions of the console instruction microprogram are encoded in hexadecimal and listed in the table below. The functions of the three console instructions are specified by the status of two binary switches: SWC, SWB, and SWA (KRD = 001B, KWE = 010B, PR = 000B). This table must be completed during the preparatory phase.</p> <p>Set different combinations of IR7 and IR4 to execute the nine machine instruction microprogram via single-pulse mode. Track microprogram transfers and execution status using micro-addresses and P-field indicator lights. Test microcommand signals output from the small socket with logic probes, record microcommand signals for four machine instructions: ADD, SUB, LDA, and STA, and create a customized table.</p> <p><b>Experiment 6 Composition of CPU and execution of machine instructions</b> (4 contact hours, 4 self-study hours)</p> <p>Decoding a simple program composed of the machine instruction system.</p> <p>Based on the circuit diagram from previous experiments, complete the wiring connections. The controller serves as the control component, while the data path (including all modules mentioned above) functions as the execution component, and the timing generator acts as the timing component. The wiring should connect the console, timing section, data path, and microprogrammed controller. Specifically, to transmit operands to the</p>
--	---

	<p>general-purpose register set (RF), the RS1, RS0, RD1, and RD0 on the data path must be connected to IR3 through IR0, while WR1 and WR0 should also be connected to IR1 and IR0 respectively.</p> <p>Load the program's machine code from Task (1) into memory via console operations. Configure data for general-purpose registers R2 and R3 along with memory-related units using digital switches SW7-SW0 as required. Note: Since setting general-purpose registers may corrupt memory data, it's recommended to configure register values before modifying memory content. Alternatively, content can be written using the PS2 keyboard in the upper-level software or experimental bench monitoring system.</p> <p>Perform a single-pulse (DP) execution of the program to record data from four registers in the general-purpose register file (RF) and data stored in RAM via the STA instruction (read from corresponding RAM units after program completion), then compare these with theoretical analysis values. During single-pulse execution, monitor the micro-address indicator, IRBUS indicator, DBUS indicator, AR2 indicator, AR1 indicator, and judgment field indicator to track the detailed process of instruction fetching and execution (allowing observation of each micro-instruction).</p> <p>Run the program again using single instruction mode (DZ), closely monitoring the IR/DBUS indicator lights and AR2/AR1 status (where each machine instruction can be observed). After execution, record the data in four RF registers and the RAM content stored by the STA instruction, then compare them with theoretical analysis values. Note: When executing programs in single instruction mode, the original data in four general-purpose registers and RAM, along with the results of the first program run, are relevant.</p> <p>The program is executed again in a continuous manner (with DB, DP, and DZ all set to 0), which corresponds to the computer's normal operation. Since the program contains a stop instruction STP, it automatically halts when this command is encountered. After execution, the data from four registers in RF and the data stored in RAM via the STA instruction are recorded for comparison with</p>
--	---

	<p>theoretical analysis values. Similarly, the original data before program execution and the results from the second execution are relevant.</p> <p><b>Experiment 7: Principle of interruption experiment</b> (4 contact hours, 4 self-study hours)</p> <p>Understand the meaning and variation conditions of each signal in the interrupt system, and manually assemble the following main program and interrupt service program into hexadecimal machine code. This task should be completed during the preview.</p> <p>Referring to the CPU composition and machine instruction execution experiment, plus the interrupt system, complete the circuit connection of this experiment. Before connecting the power supply, check the wiring carefully to confirm that there is no error.</p> <p>Store the program code of task (1) in memory and set the data of the general register set and memory-related units as needed. The value of register R1 should be set to 21H so that the program can be executed in a loop.</p> <p>From address 20H, execute the program. During the program execution, press the INTR button on the console once. After entering the interrupt, execute it in single-pulse (DP) mode until returning to the main program. The list records the changes of relevant signals in the interrupt system, especially recording the breakpoint address and the value of R0.</p> <p>Repeat (4) twice. (A total of 3 times)</p> <p>Change the content of 20H unit in RAM from INTS to INTC, redo (4), and record the phenomenon.</p> <p><b>Experiment 8 Design and debugging of microprogram controller for water flow</b> (4 contact hours, 4 self-study hours)</p> <p>On the basis of microprogram controller experiment, a microprogram controlled instruction level scalar stream model computer is designed.</p> <p>According to the design drawings, the assembly was carried out on the general test bench and debugging was successful.</p> <p>On the basis of successful assembly and debugging, design drawings and other documents are sorted out.</p>
Examination forms	Closed-written exam

Study and examination requirements	<p>Homework should be completed by the students independently after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>The usual score is 30%, the skill score is 20%, and the exam score is 50%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Tang Shuofei. Computer Composition Principles [M]. Beijing: Higher Education Press, 2008.</p> <p><b>2. Reference books</b></p> <p>[1] Zhang Sifa. Computer Composition and Assembly Language [M]. Beijing: Higher Education Press, 2007.</p> <p>[2] Upton, Eben, Duntemann, Jeff, Roberts, Ralph, Mamtara, Tim, Everard, Ben. Learning Computer Architecture with Raspberry Pi [M]. Wiley. 2015.</p> <p>[3] Wang Aiyang. Computer Composition and Structure [M]. Beijing: Tsinghua University Press, 1998.</p> <p>[4] Hu Yueming. Computer Composition and System Structure [M]. Beijing: Shanghai Scientific and Technical Literature Publishing House, 1999.</p> <p>[5] Li Yamin. Computer Composition and System Structure [M]. Beijing: Tsinghua University Press, 2000.</p> <p>[6] Bai Zhongying, Han Zhaoxuan. Computer Composition Principles [M]. Beijing: Science Press, 1998.</p>
Data of last amendment	June 10, 2025

## Operating Systems

Module designation	Operating Systems
Semester(s) in which the module is taught	2 <sup>nd</sup> semester
Person responsible for the module	Associate professor Yu Hang
Language	Chinese
Relation to curriculum	As the most critical computer system software and one of the most dynamic academic disciplines, the operating system holds a uniquely pivotal position in computer systems. Essential components like assembly programs, compilers, database management systems, and numerous application software all rely on its support to function effectively. Having become an indispensable component of modern computer systems, the operating system serves as a foundational course for computer science majors, acting as a bridge between theoretical knowledge and practical applications within the curriculum framework.
Teaching methods	Target students: students of Type of teaching: theoretical teaching, experiment teaching Contact hour: 48 hours Including: Theoretical teaching: 48 hours Experiment teaching: 0 hours Computer practice: 0 hours Size of class: 40-60 students
Workload (incl. contact hours, self-study hours)	Total workload = 90 hours Contact hours = 48 hours Self-study hours = 42 hours
Credit points	3
Required and recommended prerequisites for joining the module	Database Principle And Application, Programming Foundation
Module	Learning outcomes:

<p>objectives/intended learning outcomes</p>	<ul style="list-style-type: none"> <li>● <b>Knowledge:</b> <ol style="list-style-type: none"> <li>1. Understand the basic characteristics and main functions of the operating system.</li> <li>2. Master multithreading, concurrency, time-sharing, jobs, processes, mutual exclusion and synchronization.</li> <li>3. Master the management strategy of computer system resources (processor, memory, equipment and files).</li> </ol> </li> <li>● <b>Skill:</b> <ol style="list-style-type: none"> <li>1. Able to determine specific solutions according to system requirements or project design objectives.</li> <li>2. Solve some simple application problems.</li> </ol> </li> <li>● <b>Competence:</b> <ol style="list-style-type: none"> <li>1. Have the ability of dialectical thinking.</li> <li>2. Have a scientific and realistic style of study and innovative consciousness and spirit.</li> <li>3. Strengthen the awareness of professional ethics and safety norms.</li> </ol> </li> </ul>
<p>Content</p>	<p><b>Part A. Theoretical teaching</b> (48 contact hours; 42 self-study hours)</p> <p><b>Chapter 1 Overview of Operating Systems</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Main functions of the operating system</li> <li>● Basic features of the operating system</li> </ul> <p><b>Chapter 2 Process and thread management</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of processes</li> <li>● Process control</li> <li>● Threads</li> </ul> <p><b>Chapter 3 Process Synchronization and Communication</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Concepts and differences between synchronization and mutual exclusion</li> <li>● The semaphore mechanism solves the problem of synchronization and mutual exclusion</li> <li>● Typical process synchronization problem details</li> <li>● Process communication</li> </ul> <p><b>Chapter 4 Processor Scheduling</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Three-level scheduling system</li> </ul>

	<ul style="list-style-type: none"> <li>● Process scheduling goals and scheduling methods</li> <li>● Evaluation criteria of scheduling algorithm</li> <li>● Typical process scheduling algorithms</li> <li>● Thread scheduling algorithm</li> <li>● Real-time scheduling algorithm</li> </ul> <p><b>Chapter 5 Deadlock</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts and causes of deadlock</li> <li>● Necessary conditions for deadlock</li> <li>● Deadlock handling</li> <li>● Banker's algorithm</li> <li>● Thread deadlock</li> </ul> <p><b>Chapter 6 Memory Management</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Overview of memory management</li> <li>● Continuous distribution mode</li> <li>● Basic paging memory management mode</li> <li>● Basic segmented memory management mode</li> <li>● Basic paragraph page memory management mode</li> </ul> <p><b>Chapter 7 Virtual Storage Management</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Coverage and exchange technology</li> <li>● Virtual storage management</li> <li>● Request page storage management mode</li> <li>● Page replacement algorithm</li> <li>● Request segmented storage management mode</li> <li>● Request segment paging storage management</li> <li>● Arrays as function arguments</li> </ul> <p><b>Chapter 8 I/O Device Management</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● I/O device management overview</li> <li>● I/O system</li> <li>● I/O software</li> <li>● Equipment allocation and recovery</li> <li>● Buffer management</li> <li>● Disk memory management</li> <li>● I/O control</li> </ul> <p><b>Chapter 9 File System</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of files</li> <li>● File system</li> <li>● File structure</li> </ul>
--	---



	<ul style="list-style-type: none"> <li>● File directory and directory query</li> <li>● File storage space management</li> <li>● File sharing and protection</li> <li>● Reliability of the file system</li> </ul> <p><b>Part B. Experiment teaching</b> (0 contact hours; 0 self-study hours)</p>
Examination forms	Closed-book written exam
Study and examination requirements	<p>Homework should be completed independently by students after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>Class performance 20%, coursework 20%.</p> <p>The final exam accounts for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Yu Hongying. Computer Operating System Textbook [M]. 3rd Edition. Beijing: Tsinghua University Press, 2018.</p> <p>[2] Li Dongmei, Huang Ying, Hu Rong. Operating Systems [M]. 3rd Edition. Zhenjiang: Jiangsu University Press, 2013.</p>
Data of last amendment	June 10, 2025

## Introduction to Computer Science

Module designation	Introduction to Computer Science
Semester(s) in which the module is taught	1 <sup>st</sup> semester
Person responsible for the module	Associate professor Song Zhenyu
Language	Chinese
Relation to curriculum	The Introduction to Computer Science course serves as a foundational prerequisite for students majoring in Computer Science and Technology. Designed as an introductory guide, it systematically introduces core computer science knowledge. The program aims to provide students with a comprehensive understanding of computing's historical evolution, theoretical frameworks, and learning methodologies. By mastering fundamental concepts and developing computational thinking skills, it effectively stimulates learners' interest and initiative. This course lays the groundwork for students to deepen their expertise, enhance comprehensive competencies, and cultivate well-rounded professional capabilities.
Teaching methods	Target students: students of Computer Science Type of teaching: theoretical teaching, experiment teaching Contact hour: 16 hours Including: Theoretical teaching: 16 hours Experiment teaching: 0 hours Computer practice: 0 hours Size of class: 40-60 students
Workload (incl. contact hours, self-study hours)	Total workload = 30 hours Contact hours = 16 hours Self-study hours = 14 hours
Credit points	1
Required and recommended prerequisites for joining the module	None.
Module	Learning outcomes: ● <b>Knowledge:</b>

objectives/intended learning outcomes	<ol style="list-style-type: none"> <li>1. Stimulate students' interest, so that students have a good start in professional entry, professional ethics and career development.</li> <li>2. Understand the history of computer development and build a complete professional knowledge system framework.</li> </ol> <p>● <b>Skill:</b></p> <ol style="list-style-type: none"> <li>1. Understand the basic knowledge and learning methods of computer science, and have a preliminary computational thinking.</li> <li>2. To preliminarily understand and evaluate the impact of computer engineering practice on environment, society and culture, and to train students to pay attention to the development trend and application prospect of the discipline as well as the spirit of exploration of new technology.</li> </ol> <p>● <b>Competence:</b></p> <p>Cultivate the initial awareness of self-learning and lifelong learning, as well as patriotic mission and responsibility.</p>
---------------------------------------	--

Content	<p><b>Part A. Theoretical teaching</b> (16 contact hours; 14 self-study hours)</p> <p><b>Chapter 1 Growth and development of IT enterprises</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Development history of well-known IT enterprises at home and abroad;</li> <li>● The development and application of computer technology;</li> <li>● Development trends of computer science and technology;</li> <li>● Professional norms and ethical responsibilities in the field of computer.</li> </ul> <p><b>Chapter 2 Mathematics and Computational Science</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Why to learn mathematics;</li> <li>● The relationship between mathematics and computer science;</li> <li>● Mathematical theories related to computer science;</li> <li>● How to learn math well.</li> </ul> <p><b>Chapter 3 Professional personnel training program and curriculum system</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Computer Science and Technology training program;</li> <li>● Curriculum system of Computer Science and Technology;</li> <li>● Computer science and major advantages, knowledge and skills required for employment, professional knowledge and technical learning methods, four-year academic plan.</li> </ul> <p><b>Chapter 4 Fundamentals of Computer Science and Technology</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Computer hardware system;</li> <li>● Computer soft system;</li> <li>● Representation of data in a computer.</li> </ul> <p><b>Chapter 5 Prospects and research directions of computer science</b></p>
---------	--

	<p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Learning methods for computer science;</li> <li>● Future research content of computer science;</li> <li>● How to graduate and advance smoothly.</li> </ul> <p><b>Chapter 6 Artificial Intelligence</b></p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● The history of artificial intelligence;</li> <li>● Innovation and status quo of artificial intelligence technology;</li> <li>● Future development trends of artificial intelligence technology;</li> <li>● Architecture for artificial intelligence research.</li> </ul> <p><b>Chapter 7 Big Data Technology</b></p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● The history of big data;</li> <li>● The innovation and status quo of big data technology;</li> <li>● Future development trend of big data technology;</li> <li>● Architecture for big data application research.</li> </ul> <p><b>Chapter 8 Social Professional Ethics</b></p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● The content of computer professional quality;</li> <li>● Computer ethics and professional ethics.</li> </ul> <p><b>Part B. Experiment teaching</b> (0 contact hours; 0 self-study hours)</p>
Examination forms	Submit a study plan
Study and examination requirements	<p>Homework should be completed by the students independently after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>Regular grades account for 20%, including regular performance (20%).</p> <p>The final assessment (academic plan) accounts for 80%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Huang Guoxing. Introduction to Computer [M]. 5th edition. Beijing: Tsinghua University Press, 2024.</p> <p><b>2. Reference books</b></p> <p>[1] Dong Rongsheng. Introduction to Computer Science--Ideas and Methods [M]. 3rd edition. Beijing: Higher Education Press, 2016.</p> <p>[2] Jiang Jiafu. College Computer Practice Tutorial [M].</p>

	Beijing: University of Posts and Telecommunications Press, 2013. [3] Wang Hongmei, Hu Ming. Introduction to Computer Science [M]. Beijing: Tsinghua University Press, 2011.
Data of last amendment	June 9, 2025

## Design and Analysis of Algorithms

Module designation	Design and Analysis of Algorithms
Semester(s) in which the module is taught	4 <sup>th</sup> semester
Person responsible for the module	Associate professor Jiang Min
Language	Chinese
Relation to curriculum	This course exercises students' ability of computational thinking and collaborative innovation, enables them to have good scientific literacy and social responsibility, and lays a foundation for the subsequent study of basic courses such as Software Engineering and Introduction to Artificial Intelligence.
Teaching methods	Target students: students of Computer Science and Data Science and Big Data Type of teaching: theoretical teaching, experiment teaching Contact hour: 48 hours Including: Theoretical teaching: 32 hours Experiment teaching: 16 hours Computer practice: 0 hours Size of class: 40-60 students
Workload (incl. contact hours, self-study hours)	Total workload = 90 hours Contact hours = 48 hours Self-study hours = 42 hours
Credit points	3
Required and recommended prerequisites for joining the module	C Programming, Discrete Mathematics and Data Structures
Module objectives/intended learning outcomes	Learning outcomes: ● <b>Knowledge:</b> 1. Understand the basic concepts and principles of algorithm design and analysis. 2. Master algorithm design, analysis and optimization strategies for common problems.

	<ul style="list-style-type: none"> <li>● <b>Skill:</b> <ol style="list-style-type: none"> <li>1. The ability to solve practical problems with appropriate algorithms based on requirements.</li> <li>2. Ability to think computationally and collaborate innovatively.</li> </ol> </li> <li>● <b>Competence:</b> <ol style="list-style-type: none"> <li>1. Able to determine specific solutions according to the requirements or design objectives of intelligent application system, big data analysis system, mobile application system and other systems, and consider social, health, safety, legal, cultural and environmental factors in the design process.</li> <li>2. Be able to reflect innovative consciousness in the design process and solve complex problems innovatively.</li> </ol> </li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Basic Knowledge</b> (6 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of the algorithm</li> <li>● Pseudo code description of the algorithm</li> <li>● Mathematical basis of the algorithm</li> <li>● STL Basics</li> </ul> <p><b>Chapter 2 Divide and Conquer Strategy</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● The concept of recursion</li> <li>● The basic idea of the divide and conquer method</li> <li>● Analysis techniques of divide and conquer algorithms</li> <li>● Ways to improve the partitioning algorithm</li> <li>● Binary search technique</li> <li>● Multiplication of large integers</li> <li>● Matrix multiplication</li> <li>● Merge sort</li> </ul> <p><b>Chapter 3 Dynamic Programming</b> (8 contact hours; 8 self-study hours)</p> <ul style="list-style-type: none"> <li>● Matrix multiplication problem</li> <li>● Design idea of dynamic programming</li> <li>● Basic elements of dynamic programming algorithm</li> <li>● Investment issues</li> <li>● Backpacking</li> <li>● Longest common subsequence LCS</li> </ul>



	<p><b>Chapter 4 The Law of Greed</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Programme of activities</li> <li>● The design idea of greedy method</li> <li>● Basic elements of greedy algorithm and correctness proof</li> <li>● Handling cases where greedy method cannot get optimal solution</li> <li>● Huffman coding</li> <li>● Minimum spanning tree</li> <li>● Single source shortest path</li> <li>● The greedy method seeks the short test path</li> </ul> <p><b>Chapter 5 Backtracking and Branch Limits</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● The basic idea and applicable conditions of backtracking algorithm</li> <li>● Design steps of backtracking algorithm</li> <li>● Efficiency estimation and improvement approaches</li> <li>● Branch limits</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Implement the algorithm of finding the Kth smallest element in a sequence by using the divide and conquer method. (4 contact hours; 2 self-study hours)</li> <li>2. Dynamic programming solves the backpack problem. (4 contact hours; 4 self-study hours)</li> <li>3. The greedy algorithm seeks the shortest path. (4 contact hours; 4 self-study hours)</li> <li>4. Solve the eight queens problem by backtracking. (4 contact hours; 4 self-study hours)</li> </ol>
Examination forms	Closed-book written exam
Study and examination requirements	<p>Homework should be completed by the students independently after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>Project performance 10%, homework and online learning 10%, course experiment 20%.</p> <p>The final written test accounts for 70%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Li Chunbao. Algorithm Design and Analysis [M]. 2nd</p>

	<p>Edition. Beijing: Tsinghua University Press, 2018.</p> <p><b>2. Reference books</b></p> <p>[1] Qu Wanling, Liu Tian, Zhang Li'ang, Wang Hanpin. Algorithm Design and Analysis [M]. 2nd Edition. Beijing: Tsinghua University Press, 2016.</p> <p>[2] Qu Wanling, Liu Tian, Zhang Liang, Wang Hanqin. Algorithm Design and Analysis Exercise Solutions and Learning Guide [M]. 2nd edition. Beijing: Tsinghua University Press, 2016.</p> <p>[3] Anany Levitin, Pan Yan. Basic Algorithm Design and Analysis [M]. 3rd Edition. Beijing: Tsinghua University Press, 2015.</p> <p>[4] Mark Allen Weiss. Data Structures and Algorithm Analysis, C++ Language Description (US) [M]. Beijing: Publishing House of Electronics Industry, 2017.</p>
Data of last amendment	June 10, 2025

## Java Programming

Module designation	Java Programming
Semester(s) in which the module is taught	3 <sup>rd</sup> semester
Person responsible for the module	Lecturer Yan Xuemei
Language	Chinese
Relation to curriculum	"Java Programming Design" is a core required course for Computer Science and Technology majors, designed to equip students with the skills to design algorithms, develop programs, and debug code. Through this curriculum, students will master object-oriented programming concepts and apply Java technologies along with essential development tools for software coding. The program cultivates computational thinking through hands-on coding practice, stimulates innovative problem-solving approaches, and enhances practical problem analysis and resolution capabilities during software design processes. This integrated approach ensures balanced development of theoretical knowledge and technical proficiency, fostering well-rounded professional competence Strong scientific literacy and social responsibility.
Teaching methods	Target students: students of Computer Science and Technology. Type of teaching: theoretical teaching, experiment teaching Contact hour: 48 hours Including: Theoretical teaching: 32 hours Experiment teaching: 16 hours Computer practice: 0 hours Size of class: 40-60 students
Workload (incl. contact hours, self-study hours)	Total workload = 90 hours Contact hours = 48 hours Self-study hours = 42 hours
Credit points	3

Required and recommended prerequisites for joining the module	C Programming Design, Introduction to Computer Science
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> <li>● <b>Knowledge:</b> Be able to master the object-oriented programming idea, be able to master the basic syntax of Java language, be able to program correctly and skillfully, and be able to write a certain scale of applications.</li> <li>● <b>Skill:</b> Be able to master the basic methods of graphical user interface design, event processing and common classes, and have the ability to analyze, describe and algorithm design of general Java language computer programs.</li> <li>● <b>Competence:</b> It can enable students to have the basic ability of an excellent software developer, the awareness and ability to analyze and solve problems, cultivate students' software development programming thinking, and lay a foundation for them to use Java knowledge and skills to solve their own professional practical problems in the future.</li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Overview of Java Language</b> (1 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Features of Java;</li> <li>● Java source file and Java bytecode file;</li> <li>● The main class of the Java application and Java applet;</li> <li>● Java Virtual Machine;</li> <li>● Types and structures of Java programs;</li> <li>● The difference between a Java application and a Java applet.</li> </ul> <p><b>Chapter 2 Java Language Development Environment</b> (1 credit hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Download and install Java development tools;</li> <li>● Configuration of JDK development environment;</li> <li>● Compile and run Java applications and Java applet in the JDK environment;</li> </ul>

	<ul style="list-style-type: none"> <li>● Naming rules of Java source files;</li> </ul> <p><b>Chapter 3 Java Language Basics</b> (4 credit hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Data types;</li> <li>● Variables;</li> <li>● Basic type variables;</li> <li>● Conversion rules of data types;</li> <li>● Statement format for data input from the keyboard;</li> <li>● Operator</li> </ul> <p><b>Chapter 4 Process Control and Arrays</b> (4 credit hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Statements and compound statements;</li> <li>● Branch structure;</li> <li>● Loop structure;</li> <li>● Jump statement;</li> <li>● Definition of one-dimensional and multi-dimensional arrays;</li> <li>● Access to array elements;</li> <li>● Strings and Applications.</li> </ul> <p><b>Chapter 5 Category and Object</b> (4 credit hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Definition of classes;</li> <li>● Member variables and member methods;</li> <li>● Modifiers for classes and members;</li> <li>● Creation and use of objects;</li> <li>● Access to member variables and method calls;</li> <li>● Parameter passing;</li> <li>● Anonymous objects;</li> </ul> <p><b>Chapter 6 Inheritance and Polymorphism</b> (4 credit hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Private members and public members of a class;</li> <li>● Method reloading;</li> <li>● Construction method;</li> <li>● Static members of classes;</li> <li>● Application of objects;</li> <li>● Creation of subclasses;</li> <li>● Accessing members of a parent class in a subclass;</li> <li>● Methods that cover parent classes.</li> </ul> <p><b>Chapter 7 Interface and Implementation</b> (4 credit hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Creation of subclasses;</li> <li>● Accessing members of a parent class in a subclass;</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>● Methods that cover parent classes;</li> <li>● Abstract classes and methods;</li> <li>● Interface and implementation of the interface;</li> <li>● Implement multiple inheritance of classes using interfaces;</li> <li>● Internal classes and anonymous classes;</li> <li>● Package (class library);</li> </ul> <p><b>Chapter 8 Common Java Classes</b> (2 credit hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Common Java classes;</li> <li>● Use of common Java classes.</li> </ul> <p><b>Chapter 9 Graphic Interface Design</b> (4 credit hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Swing graphical user interface toolkit;</li> <li>● Swing component classification;</li> <li>● The internal structure of the Swing container;</li> <li>● Component creation;</li> <li>● Window layout management design.</li> </ul> <p><b>Chapter 10 Event Handling</b> (4 credit hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Entrusted event model;</li> <li>● Conditions for being a listener;</li> <li>● Events in Java;</li> <li>● Listener interface and adapter class;</li> <li>● Components and the events they trigger and the corresponding event handling;</li> <li>● Menu bar design;</li> <li>● Toolbar design;</li> <li>● Sliding bar design;</li> <li>● Apply the file selection dialog;</li> <li>● Application of color selection pane;</li> <li>● Timer application;</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Compile and run Java program. (4 contact hours; 2 self-study hours)</li> <li>2. Parameter transfer. (4 contact hours; 4 self-study hours)</li> <li>3. Class inheritance. (4 contact hours; 4 self-study hours)</li> <li>4. Operation event. (4 contact hours; 4 self-study hours)</li> </ol>
--	---

Examination forms	Closed-book written exam
Study and examination requirements	<p>1. This course adopts a combination of process assessment, skill assessment and final assessment</p> <p>2. Process assessment consists of classroom performance, study notes, daily assignments and experiments</p> <p>3. The final examination will be in the form of closed written examination</p> <p>4. Composition of results</p> <p>The comprehensive score = 30% of the regular score + 70% of the final score</p> <p>Regular score = 30% of regular homework + 30% of regular assessment + 40% of experiment</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Geng Xiangyi, Zhang Yueping. Java Programming Design Tutorial [M]. 4th Edition. Beijing: Tsinghua University Press, 2022.</p> <p><b>2. Reference books</b></p> <p>[1] Chen Guojun. Java Basic of Programming [M]. 7th edition. Beijing: Tsinghua University Press, 2022.</p> <p>[2] Kathy Sierra, Head First Java in Practice [M]. 3rd edition. Beijing: China Electric Power Press, 2023.</p>
Data of last amendment	June, 2025

## Introduction to Artificial Intelligence

Module designation	Introduction to Artificial Intelligence
Semester(s) in which the module is taught	5 <sup>th</sup> semester
Person responsible for the module	Associate professor Chen Lin
Language	Chinese
Relation to curriculum	This course aims to help students understand that artificial intelligence is a highly integrated interdisciplinary field developed through the study of computer science, cybernetics, information theory, neuropsychology, philosophy, linguistics, and other disciplines. The course's objective is to equip students with fundamental principles of AI and familiarize them with essential technologies commonly used in this field.
Teaching methods	Target students: students of Computer Science and Technology. Type of teaching: theoretical teaching, experiment teaching Contact hour: 48 hours Including: Theoretical teaching: 32 hours Experiment teaching: 16 hours Computer practice: 0 hours Size of class: 40-60 students
Workload (incl. contact hours, self-study hours)	Total workload = 90 hours Contact hours = 48 hours Self-study hours = 42 hours
Credit points	3
Required and recommended prerequisites for joining the module	Advanced Mathematics, Linear Algebra, Discrete Mathematics, Programming, Probability and Statistics, Operations Research
Module objectives/intended learning outcomes	Learning outcomes: ● <b>Knowledge:</b> 1. Understand the development status, research content and great contribution of artificial intelligence to human



	<p>beings.</p> <p>2. Master the basic principles, methods and important algorithms of artificial intelligence</p> <p>● <b>Skill:</b></p> <p>1. Have an accurate and comprehensive grasp of knowledge representation, problem search principle, knowledge reasoning and other aspects.</p> <p>2. To understand the main applications of artificial intelligence, combined with specific application systems, so that students have a relatively perceptual understanding and simple application of artificial intelligence technologies such as advanced search technology and machine learning.</p> <p>3. Can apply artificial intelligence technology to improve the analysis and solution of complex problems.</p> <p>● <b>Competence:</b></p> <p>1. Let students acquire knowledge while cultivating noble sentiments and improving the realm of life.</p> <p>2. Make students understand the legal system related to the development of artificial intelligence, such as intellectual property rights, data property, and the determination of tort liability.</p>
Content	<p><b>Part A. Theoretical teaching</b></p> <p>(32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Introduction</b></p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● The fundamental concept of artificial intelligence</li> <li>● A Brief History of Artificial Intelligence</li> <li>● Basic content of artificial intelligence research</li> <li>● Major research areas of artificial intelligence</li> </ul> <p><b>Chapter 2 Knowledge Representation</b></p> <p>(6 contact hours; 4 self-study hours)</p> <p>1. Concepts of knowledge and knowledge representation</p> <ul style="list-style-type: none"> <li>● The concept of knowledge</li> <li>● The characteristics of knowledge</li> <li>● Classification of knowledge</li> <li>● Representation of knowledge</li> </ul> <p>2. First-order predicate logic notation</p> <ul style="list-style-type: none"> <li>● Proposition</li> <li>● Predicates</li> <li>● Predicative formula</li> </ul>

	<ul style="list-style-type: none"> <li>● The nature of predicate formulas</li> <li>● First-order predicate logic knowledge representation method</li> <li>● The characteristics of first-order predicate logic knowledge representation</li> </ul> <p>3. Generative notation</p> <ul style="list-style-type: none"> <li>● Generative</li> <li>● Production system</li> <li>● Examples of productive systems</li> <li>● Characteristics of productive systems</li> </ul> <p>4. Framework notation</p> <ul style="list-style-type: none"> <li>● The general structure of the framework</li> <li>● Examples of using frameworks to represent knowledge</li> <li>● Characteristics of framework representation</li> </ul> <p>5. Semantic network representation</p> <ul style="list-style-type: none"> <li>● Semantic network</li> <li>● Semantic network representation of basic propositions</li> <li>● The representation of conjunctions in semantic networks</li> </ul> <p><b>Chapter 3 Deterministic reasoning method</b> (8 contact hours; 6 self-study hours)</p> <p>1. Basic concepts of reasoning</p> <ul style="list-style-type: none"> <li>● Definition of reasoning</li> <li>● Inference mode and its classification</li> <li>● The direction of reasoning</li> <li>● Conflict resolution strategy</li> </ul> <p>2. Natural deductive reasoning</p> <p>3. The method of formulating predicate into set of clauses</p> <p>4. Heberlen's theorem</p> <p>5. Robinson's principle of convergence</p> <p>6. Reverse synthesis</p> <p>7. Apply the principle of reduction to solve problems</p> <p><b>Chapter 4 Method of uncertain reasoning</b> (8 contact hours; 8 self-study hours)</p> <p>1. Basic concepts in uncertain reasoning</p> <p>2. Probability method</p> <ul style="list-style-type: none"> <li>● Classical probability method</li> <li>● Reverse probability method</li> </ul> <p>3. Credibility method</p> <ul style="list-style-type: none"> <li>● The concept of credibility</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>● C-F model</li> </ul> <p>4. Theory of evidence</p> <ul style="list-style-type: none"> <li>● Probability distribution function</li> <li>● Trust function</li> <li>● Likelihood function</li> <li>● The relationship between trust function and likelihood function</li> <li>● Orthogonal sum of probability distribution function</li> <li>● Uncertainty reasoning based on evidence theory</li> </ul> <p>5. Fuzzy reasoning method</p> <ul style="list-style-type: none"> <li>● The proposal and development of fuzzy logic</li> <li>● Fuzzy set</li> <li>● Operations of fuzzy sets</li> <li>● Fuzzy relations and synthesis of fuzzy relations</li> <li>● Fuzzy reasoning</li> <li>● Fuzzy decision</li> </ul> <p><b>Chapter 5 Search and Solution Strategies</b> (4 contact hours; 4 self-study hours)</p> <p>1. The concept of search</p> <ul style="list-style-type: none"> <li>● Basic and main problems of search</li> <li>● Search strategy</li> </ul> <p>2. State space knowledge representation method</p> <ul style="list-style-type: none"> <li>● State space notation</li> <li>● Description of state space diagram</li> </ul> <p>3. Blind graph search strategy</p> <ul style="list-style-type: none"> <li>● Backtracking strategy</li> <li>● Width priority search strategy</li> <li>● Depth first search strategy</li> </ul> <p>4. AND/OR graph search strategy</p> <p><b>Chapter 9 Machine Learning</b> (4 contact hours; 4 self-study hours)</p> <p>1. Basic concepts of machine learning</p> <ul style="list-style-type: none"> <li>● Learning</li> <li>● machine learning</li> <li>● Machine learning system</li> <li>● The development of machine learning</li> <li>● Classification of machine learning</li> </ul> <p>2. symbolic learning</p> <ul style="list-style-type: none"> <li>● Mechanical learning</li> <li>● Guided learning</li> <li>● inductive learning</li> <li>● Example learning</li> </ul>
--	---

	<ul style="list-style-type: none"> <li>● Observation and discovery learning</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <p>Experiment 1: Monkey picking banana problem (4 contact hours; 2 self-study hours)</p> <p>There is a monkey in the room, a box, and a string of bananas hanging from the ceiling. In order to get the bananas, the monkey must move the box under the bananas, and then climb on the box to get the bananas.</p> <ol style="list-style-type: none"> <li>1. Define the necessary predicates and list the initial state and target state of the problem;</li> <li>2. Use predicates, conjunctions and quantifiers to represent environmental states;</li> <li>3. From the initial state to the target state, the monkey needs to complete a series of operations, and define the operation class predicate to represent its actions;</li> <li>4. According to the action plan, replace the state step by step until the target state is reached.</li> </ol> <p>Experiment 2 Robot Box Problem (4 contact hours; 4 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Use the predicate formula to represent the initial state, target state and robot operation;</li> <li>2. Convert the predicate formula into a set of clauses;</li> <li>3. Use the reduction principle to reduce the clauses in the set of clauses.</li> </ol> <p>Experiment 3: The Problem of the Monk and the Wild Man (4 contact hours; 4 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Get the state space diagram according to the relationship between the number of monks and the number of savages;</li> <li>2. Select an appropriate search algorithm to give the monk and the savage a safe way to cross the river;</li> <li>3. Implement the problem in a certain language and give the solution to the problem.</li> </ol> <p>Experiment 4. Implementing the eight-digit problem by state space search method (4 contact hours; 4 self-study hours)</p> <ol style="list-style-type: none"> <li>1. List the state equations according to the initial state and termination state of the eight digits;</li> <li>2. Select a blind search algorithm or a heuristic search</li> </ol>
--	--

	algorithm to program the solution of the eight digit problem; 3. Use reasonable analysis to make reasonable results.
Examination forms	Short paper
Study and examination requirements	Homework should be completed by the students independently after each lesson. No late arrivals, early departures or unapproved absences are allowed. The result consists of regular score (20%), experimental score (20%) and final score (60%).
Reading list	<b>1. Required books</b> [1] Wang Wanliang. Artificial Intelligence and Its Applications [M]. Beijing: Higher Education Press, 2016. <b>2. Reference books</b> [1] Cai Zixing, Meng Zuqiang. Basic Artificial Intelligence [M]. Beijing: Higher Education Press, 2016. [2] Lu Dexu. Artificial Intelligence Tutorial: Programming Practice of humanoid Robots [M]. Qingdao: Qingdao Publishing House, 2018. [4] Lu Dexu. Artificial Intelligence Tutorial [M]. Qingdao: Qingdao Publishing House, 2018. [5] Stephen Lucci, Danny Kopec. Artificial Intelligence [M]. translated by Lin Ci. Beijing: People's Posts and Telecommunications Press, 2018.
Data of last amendment	June 10, 2025

## Principles of Compilers

Module designation	Principles of Compilers
Semester(s) in which the module is taught	5 <sup>th</sup> semester
Person responsible for the module	Professor Huang Xingping
Language	Chinese
Relation to curriculum	This course is a core requirement for Computer Science and Technology majors. Through this program, students will gain comprehensive understanding of compiler design principles and implementation techniques in programming languages, including compiler architecture methodologies. The curriculum covers fundamental methods and key implementations such as syntax analysis technologies, intermediate code generation symbol tables, code optimization, parallel compilation techniques, and runtime memory space organization. By mastering these concepts, students will deepen their comprehension of high-level programming languages and develop the ability to apply knowledge across different domains.
Teaching methods	Target students: students of Computer Science. Type of teaching: theoretical teaching, experiment teaching Contact hour: 32 hours Including: Theoretical teaching: 32 hours Experiment teaching: 0 hours Computer practice: 0 hours Size of class: 40-60 students
Workload (incl. contact hours, self-study hours)	Total workload = 60 hours Contact hours = 32 hours Self-study hours = 28 hours
Credit points	2
Required and recommended prerequisites for	Data structure, C language programming, operating system principle

joining the module	
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> <li>● <b>Knowledge:</b> Understand the fundamental concepts of compiler principles, master the basic working principles and components of compilers, including lexical analysis, syntactic analysis, semantic analysis, optimization, and code generation. Be able to comprehend key technologies and algorithms in the compilation process, and establish connections between the compilation process and program execution.</li> <li>● <b>Skill:</b> Basic methods of decompiler design and implementation, including an overview of compiler architecture, simple construction techniques of compilers, etc. Students can use the knowledge to analyze simple compilation problems and try to solve some basic compilation tasks.</li> <li>● <b>Competence:</b> Develop the ability to integrate theory with practice, capable of designing and implementing basic compilers or compiler components through experiments and project-based learning. Possess teamwork spirit and coordination skills, maintain enthusiasm for exploring new technologies while staying attuned to trends and application prospects in compilation technology development. Additionally, focus on cultivating innovative thinking and problem-solving capabilities to adapt to evolving programming environments and demands.</li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Basic Concepts of Compiler</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● What is a compiler;</li> <li>● Overview of compilation process;</li> <li>● Structure of compiler;</li> <li>● Combination of compilation stages;</li> <li>● Compilation techniques and software tools.</li> </ul> <p><b>Chapter 2 Basic Knowledge of Formal Language</b> (6 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Intuitive concept of grammar;</li> <li>● Symbols and symbol strings;</li> </ul>

	<ul style="list-style-type: none"> <li>● The formal definition of grammar and language;</li> <li>● Classification of grammar;</li> <li>● Context-free syntax and its syntax tree;</li> <li>● Sentence pattern analysis;</li> <li>● Some remarks on the practical application of grammar.</li> </ul> <p><b>Chapter 3 Lexical analysis</b> (8 contact hours; 8 self-study hours)</p> <ul style="list-style-type: none"> <li>● Design of lexical analysis program;</li> <li>● Word description tools;</li> <li>● Finite automata;</li> <li>● Equivalence between regular expressions and finite automata;</li> <li>● Conversion between formal grammar and finite automata;</li> <li>● Automatic construction tools for lexical analysis programs.</li> </ul> <p><b>Chapter 4 Grammar Analysis</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● The established top-down analysis approach;</li> <li>● LL (1) Grammar determination;</li> <li>● Equivalent transformations from LL(1) grammar to LL(1) grammar;</li> <li>● Uncertain top-down analysis;</li> <li>● The identified top-down analysis approach.</li> </ul> <p><b>Chapter 5 Semantic analysis and intermediate code generation</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Attribute grammar;</li> <li>● Overview of grammar-guided translation;</li> <li>● The form of intermediate code;</li> <li>● Translation of simple assignment statements;</li> <li>● Translation of bool expressions;</li> <li>● Translation of the control structure;</li> <li>● Translation of the explanatory section;</li> <li>● Translation of arrays and structures.</li> </ul> <p><b>Chapter 6 Review</b> (2 contact hours; 2 self-study hours)</p> <p><b>Part B. Experiment teaching</b> (0 contact hours; 0 self-study hours)</p>
Examination forms	Closed-book written exam



Study and examination requirements	<p>Homework should be completed by the students independently after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>Regular grades account for 40%, including homework (20%) and regular performance (20%).</p> <p>The final assessment (closed written test) accounts for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Wang Shengyuan, Dong Yuan, Zhang Suqin, Lv Yingzhi, Jiang Weidu. Compilation Principles [M]. Beijing: Tsinghua University Press, 2015.</p> <p><b>2. Reference books</b></p> <p>[1] Li Wensheng. Principles and Techniques of Compilation [M]. Beijing: Tsinghua University Press, 2016.</p> <p>[2] Alfred V.Aho, Monica S.Lam, Ravi Sethi. Principles of Compilation [M]. translator: Zhao Jianhua, Zheng Tao, Dai Xinyu. Beijing: China Machine Press, 2008.</p> <p>[3] Li Wensheng. Learning Guide and Exercise Analysis of Compilation Principles and Techniques [M]. 2nd Edition. Beijing: Tsinghua University Press, 2016.</p> <p>[4] Huang Xianying. Principles and Practice of Compilation [M]. 3rd Edition. Beijing: Tsinghua University Press, 2019.</p> <p>[5] Hu Yuanyi. Tutorial of Compilation Principles [M]. 4th Edition. Xi 'an: Xi 'an University of Electronic Science and Technology Press, 2018.</p> <p>[6] Hu Yuanyi. Exercise analysis and Computer Practice Guide for Principles of Compilation [M]. 4th Edition. Xi'an: Xidian University Press, 2017.</p> <p>[7] Wu Chunxiang. Principles of Compilation--Exercises and Analysis [M]. Beijing: Tsinghua University Press, 2001.</p>
Data of last amendment	June 10, 2025

## Linux Operating System

Module designation	Linux Operating System
Semester(s) in which the module is taught	5 <sup>th</sup> semester
Person responsible for the module	Associate professor Hua Li
Language	Chinese
Relation to curriculum	<p>The course covers: an overview of the Linux operating system, graphical interfaces, command line basics, file management, process management, and Linux programming.</p> <p>The course aims to provide students with a comprehensive understanding of the core concepts and practical applications of mainstream Linux operating systems. It cultivates students' ability to apply acquired knowledge in integrated practice, enhances their interest in specialized courses, and ultimately develops their capacity to analyze problems and solve real-world challenges using computer skills. Furthermore, it fosters students' passion for their major and motivates them to pursue further academic development.</p>
Teaching methods	<p>Target students: students of Data Science and Big Data</p> <p>Type of teaching: theoretical teaching, experiment teaching</p> <p>Contact hour: 32 hours</p> <p>Including:</p> <p>Theoretical teaching: 16 hours</p> <p>Experiment teaching: 16 hours</p> <p>Computer practice: 0 hours</p> <p>Size of class: 40-60 students</p>
Workload (incl. contact hours, self-study hours)	<p>Total workload = 60 hours</p> <p>Contact hours = 32 hours</p> <p>Self-study hours = 28 hours</p>
Credit points	2
Required and recommended prerequisites for	Introduction To Data Science And Big Data Technology, Operating System, C Programming

joining the module	
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> <li>● <b>Knowledge:</b> Have a comprehensive understanding of the basic knowledge and basic use of the mainstream Linux operating system, and cultivate students' ability to apply the knowledge for comprehensive practice.</li> <li>● <b>Skill:</b> 1.Be able to understand the classification of operating systems and the development history of Linux, and be proficient in the installation and remote login methods of a Linux distribution (Ubuntu version is recommended). 2.Linux file and directory management, Linux system user and user group management, Linux disk management, text editing tools Vi/Vim, document compression and packing 3.Basic knowledge of shell, basic ability to write shell scripts and use regular expressions.</li> <li>● <b>Competence:</b> Through the study of this course, students can understand the pride of building a strong cyber country and the collectivism responsibility of teamwork in the process of practical operation on Linux, so as to form good professional ethics, abide by professional ethics and establish correct labor values in their future work.</li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (16 contact hours; 14 self-study hours)</p> <p><b>Chapter 1 Overview of Linux operating system</b> (1 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● The development history of UNIX and UNIX-like operating systems, and the birth of Linux;</li> <li>● Representative Linux distributions;</li> <li>● Traditional enterprise server field, smart phone, tablet computer, Internet of Things, Internet of Vehicles and other application scenarios</li> </ul> <p><b>Chapter 2 Basic graphical interface</b> (1 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● X Window System, KDE and GNOME;</li> <li>● Download the latest version of Ubuntu image file;</li> <li>● Overview of desktop environment, common</li> </ul>

	<p>applications and basic system Settings;</p> <p><b>Chapter 3 Linux Operating System Command Line Basics</b></p> <p>(2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Shell Introduction</li> <li>● Linux operating system command prompt</li> <li>● Linux command syntax format, pipeline, redirection, command arrangement, command continuation, command substitution, command alias</li> <li>● Use the man and info commands for help</li> </ul> <p><b>Chapter 4: Files and Directory Management</b></p> <p>(2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Create empty file command touch, file copy command cp, file link command ln;</li> <li>● File move command mv and file delete command rm;</li> <li>● Display the current path command pwd, change the working directory command cd;</li> <li>● List directory contents command ls, create directory command mkdir, delete directory command rmdir;</li> </ul> <p><b>Chapter 5 User and Group Management</b></p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Linux user account and Ubuntu user account;</li> <li>● User configuration files: Password and shadow files;</li> <li>● Create user account command useradd and adduser;</li> <li>● Modify user account commands: Password, usermod and change;</li> <li>● Delete user account commands userdel and deluser;</li> <li>● Group account configuration file group, gshadow files;</li> <li>● Create group account commands groupadd and addgroup;</li> <li>● Modify the group account attribute command groupmod;</li> <li>● Delete group accounts and manage group accounts command gpasswd;</li> </ul> <p><b>Chapter 6 Disk Storage Management</b></p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Introduction to disk partitioning and formatting;</li> <li>● Disk partition naming rules, disk partition management command fdisk;</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>● Create the file system command mkfs;</li> <li>● Mount the disk partition command mount and unmount the disk partition command umount;</li> <li>● File system check and repair command fsck, view disk usage df;</li> <li>● Check the disk usage of files and directories with the command du;</li> </ul> <p><b>Chapter 7 Process Management</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● The concept of process, program and process, process state, process classification, process priority;</li> <li>● Check the status of the current process with the command ps and continuously check the running status of the process with the command top;</li> <li>● Check the process tree command pstree.</li> <li>● List the process open file information command lsof.</li> <li>● The command to adjust the priority of a process is nice; the command to change the priority of a process is renice.</li> <li>● The command kill to send a signal to a process, and the command killall to kill a process by name.</li> <li>● Process startup and suspension.</li> <li>● Display task status command jobs, move the task to the foreground with fg, move it to the background with bg, start the task running away from the terminal with nohup.</li> </ul> <p><b>Chapter 8 Shell Programming</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Script, definition of variables, types, input and output;</li> <li>● Arithmetic expressions and logical expressions;</li> <li>● Branch structure: if statement, case statement;</li> <li>● Loop structure: for statement, while statement, until statement;</li> <li>● Definition, invocation, parameter passing and return value of the function;</li> <li>● Advanced: software installation methods and package management tools in Linux operating system;</li> <li>● apt, overview and configuration of apt source, basic usage of apt command, and operation examples of apt command;</li> </ul> <p><b>Chapter 9 Big Data</b></p>
--	---

	<p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Hadoop、HDFS、MapReduce;</li> <li>● Update the software package information and install the Java environment;</li> <li>● Install the big data development platform;</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <ol style="list-style-type: none"> <li>1. Basis of graphical interface (2 contact hours; 2 self-study hours)</li> <li>2. Linux command line (2 contact hours; 2 self-study hours)</li> <li>3. Files and directory management (2 contact hours; 2 self-study hours)</li> <li>4. User and group management (2 contact hours; 2 self-study hours)</li> <li>5. Disk storage management (2 contact hours; 2 self-study hours)</li> <li>6. Process management (2 contact hours; 2 self-study hours)</li> <li>7.Shell programming (4 contact hours; 2 self-study hours)</li> </ol>
Examination forms	Closed-book written exam
Study and examination requirements	<p>Homework should be completed independently by students after each lesson.</p> <p>No late arrivals, early departures or unapproved absences are allowed.</p> <p>Online learning 10%, Class performance 20%, coursework 30%.</p> <p>The final exam accounts for 40%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Zhangping. Ubuntu Linux Operating system case tutorials [M]. Beijing: People's Posts and Telecommunications Press, 2021.</p> <p><b>2. Reference books</b></p> <ol style="list-style-type: none"> <li>1. Ling Jing, Bi Guofeng. Practical Tutorial of Linux Operating System [M]. Beijing: Publishing House of Electronics Industry, 2020.</li> <li>2. Du Yan, Lian Zhe, Li Song. Practical Tutorial of Ubuntu Linux Operating System [M]. Beijing: People's Posts and Telecommunications Press, 2017.</li> </ol>

Data of last amendment
---------------------------

June 10, 2025
---------------