

Taizhou University



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

Appendix D
Syllabus - Electives

Department of Computer Science and Technology
School of Information Engineering

2025

Contents

Fundamentals of Cloud Computing.....	1
Intelligent Healthcare Systems.....	8
Web Application Development.....	13
Innovative Training for Programming Contests.....	18
Cybersecurity.....	24
Scientific and Technical Writing.....	30
Fundamentals of Hadoop.....	34
Fundamentals of Spark.....	41
Big Data Development Technologies.....	47
Development of Applications Based on Large AI Models.....	50

Fundamentals of Cloud Computing

Module designation	Fundamentals of Cloud Computing
Semester(s) in which the module is taught	6th semester
Person responsible for the module	Gu Shanshan
Language	Chinese
Relation to curriculum	<p>Cloud computing is an emerging internet-based computing model that provides flexible computing resources and data services on demand. This course aims to cultivate students' ability to learn and familiarize themselves with the basic theories, technologies, methods, applications, and specific cloud computing systems of cloud computing, virtualization, and virtualization security technologies. It covers cloud computing architecture design, key cloud computing technologies, virtualization technologies, the current status of virtualization security, key virtualization security technologies, and the latest research directions. This course introduces the principles and methods of several representative cloud computing, virtualization, and virtualization security technologies in the field of cloud computing, virtualization, and virtualization security, and combines experiments for teaching. It is required students to be able to build a cloud computing environment, choose appropriate cloud computing platforms for application development, and combine virtualization security software to carry out real-time protection for the security of cloud platforms on the basis of understanding and being familiar with the concepts, principles, and technologies related to cloud computing, virtualization, and virtualization security technologies. In this way, it can lay the foundation for the subsequent implementation of related engineering applications and scientific research.</p>

Teaching methods	<p>Target students: students of Computer Science and Technology</p> <p>Type of teaching: theoretical teaching, experiment teaching</p> <p>Contact hour: 32 hours Including:</p> <p>Theoretical teaching: 16</p> <p>Experiment teaching: 16</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</p> <p>Contact hours = 32</p> <p>Self-study hours = 28</p>
Credit points	2
Required and recommended prerequisites for joining the module	Principles of Operating Systems, Computer Communication and Networks, and Principles of Databases
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> • Knowledge: <ol style="list-style-type: none"> 1. Master the basic concepts, development trends, service models (IaaS, PaaS, SaaS), infrastructure, and cloud solutions of cloud computing. Be familiar with the principles and methods of technologies such as virtualization, distributed storage and batch computing, cloud native, and containers, and master the working principles of key cloud computing technologies. 2. Understand cloud computing products such as Alibaba Cloud ECS, SLB, RDS, and OSS, and master and proficiently operate cloud computing products. 3. Master the principles and methods of technologies such as virtualization, distributed storage and batch computing, cloud native, and containers of key cloud computing technologies.

	<ul style="list-style-type: none"> • Skill: <ol style="list-style-type: none"> 1. Build a simple cloud computing environment and master the basic operations of mainstream cloud computing platforms. 2. Select appropriate cloud computing platforms for application development according to actual needs. 3. Use virtualization security software to carry out real-time protection and management for the security of cloud platforms. • Competence: <ol style="list-style-type: none"> 1. Master the methods and general procedures of cloud computing experiments, and conduct experimental operations safely and correctly. 2. Complete tasks such as the design, deployment, and debugging of cloud computing systems in a group. 3. Assume roles as an individual, team member, and leader in a group, communicate effectively with team members, and complete work independently or collaboratively.
Content	<p>Part A. Theoretical teaching (16 contact hours; 14 self-study hours)</p> <p>Chapter 1 Overview of Cloud Computing (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> ● Development history, basic concepts and characteristics ● Deployment models and service models ● Relationship between cloud computing and big data <p>Chapter 2 Cloud Computing Case Studies (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Alibaba Cloud ECS(Elastic Compute Service) Architecture Analysis ● Alibaba Cloud SLB(Server Load Balancer) Architecture Analysis ● Alibaba Cloud RDS(Relational Database Service)

	<p>Architecture Analysis</p> <ul style="list-style-type: none"> ● Alibaba Cloud OSS(Object Storage Service) Architecture Analysis ● Comprehensive Application Case: Building a Forum Website Using ECS, SLB, RDS, and OSS <p>Chapter 3 Cloud Computing Related Technologies — Principles and Applications of Virtualization Technology</p> <p>(4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Fundamental Concepts, Objectives, Classification, and Hybrid Implementation Methods of Compute Virtualization ● Analysis of a Typical Compute Virtualization Architecture—KVM Example ● Summary of Common Virtualization Product Categories ● Fundamental Concepts and Implementation Methods of Storage Virtualization ● Relationship Between Storage Virtualization and Cloud Storage ● Fundamental Concepts, Objectives, and Characteristics of Network Virtualization ● Key Technologies of Virtualized Network Architectures <p>Chapter 4 Cloud Computing Related Technologies — Distributed Storage and Batch Computing</p> <p>(4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Fundamental Ideas ● Classic Distributed System: Hadoop and Installation Lab ● Basic Concepts and Storage Principles of HDFS ● MapReduce Workflow and Programming Examples <p>Chapter 5 Cloud Computing Related Technologies—Cloud Native</p> <p>(2 contact hours; 2self-study hours)</p>
--	---

	<ul style="list-style-type: none"> ● Origin, Development, Architecture, and 12 Factors ● Overview, Frameworks, Partitioning, and Architecture of Microservices ● Security Analysis of Cloud Native Systems <p>Part B. Experiment teaching (16 contact hours; 12 self-study hours)</p> <p>Experiment 1: Cloud Computing Systems and Virtual Machines (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Objective: Use and test public cloud computing systems in an internet environment ● Provision a virtual machine (VM) on Alibaba Cloud ECS ● Configure network security groups and SSH access ● Deploy a simple web application (e.g., Nginx) on the VM ● Analyze resource utilization metrics (CPU, memory, storage) <p>Experiment 2: MapReduce Programming Model and Hadoop System (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Objective: Master distributed data processing systems in cloud environments ● Deploy a Hadoop cluster (single-node or multi-node) on Alibaba Cloud ECS ● Write and execute a MapReduce job (e.g., WordCount) to process large datasets ● Explore HDFS file operations (upload, download, replication) ● Optimize cluster performance through configuration tuning <p>Experiment 3: DevOps Tool Platforms (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Objective: Master DevOps automation workflows for software delivery and infrastructure changes ● Set up a CI/CD pipeline using GitLab CI/CD or
--	---

	<p>Jenkins</p> <ul style="list-style-type: none"> ● Integrate automated testing (unit/integration tests) ● Configure infrastructure as code (IaC) with Terraform ● Deploy applications to Kubernetes clusters via CI/CD pipelines <p>Experiment 4: Container Technology and Docker Orchestration</p> <p>(4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Objective: Master Docker containerization and orchestration. ● Build a Docker image for a sample application ● Deploy and manage containers with Docker Compose ● Set up a Kubernetes cluster (e.g., using Alibaba Cloud Container Service for Kubernetes) ● Deploy a multi-container application to Kubernetes
Examination forms	Final essay
Study and examination requirements	<p>Homework assignments shall be completed independently after each class.</p> <p>Punctuality is mandatory: late arrivals, early departures, and unauthorized absences are prohibited.</p> <p>Usual performance accounts for 50%, including assignments (10%), in-class tests (10%), technical sharing (10%), and computer experiments (20%).</p> <p>The final assessment (final essay) constitutes the remaining 50%.</p>
Reading list	<p>1. Required books</p> <p>[1] Song Yaqi, Li Li, Yan Lei. Cloud Computing Technology and Applications [M]. Beijing: Publishing House of Electronics Industry, 2022.</p> <p>2. Reference books</p> <p>[1] Liu Peng. Cloud Computing [M]. 2nd Edition. Beijing: Publishing House of Electronics Industry, 2011.</p>

	<p>[2] Xing Li, Bian Xuefen, Wang Peng. Cloud Computing and Big Data Technology [M]. Beijing: Posts & Telecom Press, 2022.</p> <p>[3] Lü Yunxiang, Bai Yanzheng, Xu Hongzhi, Zhang Lu, Wang Jiawei. Introduction to Cloud Computing [M]. 2nd Edition. Beijing: Tsinghua University Press, 2020.</p>
Data of last amendment	June 29, 2025

Intelligent Healthcare Systems

Module designation	Intelligent Healthcare Systems
Semester(s) in which the module is taught	7 th semester
Person responsible for the module	Lecturer Bao Zhenshen
Language	Chinese
Relation to curriculum	<p>The Smart Healthcare System is an elective course for majors in Computer Science and Technology, Internet of Things Engineering, and related fields. This course integrates medical knowledge with information technology to introduce the system architecture, key technologies, and application scenarios of smart healthcare. The curriculum covers core modules such as medical data management, intelligent diagnosis, and telemedicine, representing a seamless integration of computer technology, IoT technology, and the medical industry. Through analyzing the technical principles, system design, and practical case studies of smart healthcare systems, the course equips students with both theoretical foundations and practical skills for future careers in medical informatization. Students will master the design, development, and maintenance skills required for smart healthcare systems Ability to apply information technology to medical innovation.</p>
Teaching methods	<p>Target students: students of Computer Science and Technology, Internet of Things engineering.</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,	<p>Total workload = 60 hours</p> <p>Contact hours = 32 hours</p> <p>Self-study hours = 28 hours</p>

self-study hours)	
Credit points	2
Required and recommended prerequisites for joining the module	Database principle and application, computer network, Python programming, Internet of Things technology foundation
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Knowledge: <ol style="list-style-type: none"> 1. Master the system architecture, core technologies and typical application scenarios of smart medical system, and understand the strategic position and development trend of smart medical in the medical industry. 2. Understand the characteristics, classification and management methods of medical data, and master the basic principles of medical big data processing and analysis. 3. Familiar with the working principle and technical implementation of intelligent diagnosis, telemedicine, health management and other core modules of smart medical treatment. ● Skill: <ol style="list-style-type: none"> 1. Able to design medical data storage scheme by database technology, and master the methods of medical data cleaning, integration and analysis. 2. Able to design network solutions for medical devices based on Internet of Things technology, and realize real-time monitoring and data collection of medical devices. 3. It can develop simple intelligent diagnostic model and realize machine learning based disease auxiliary diagnosis function. 4. Able to design and develop telemedicine prototype system to realize simple remote diagnosis and treatment functions. ● Competence: <ol style="list-style-type: none"> 1. Master the requirements analysis, system design and development process of smart medical system, and be able to participate in the whole life cycle development of smart medical projects. 2. Able to complete the design, development, testing and deployment of smart medical system in the team, with

	<p>good teamwork and communication skills.</p> <p>3. Able to track the latest technological trends in the field of smart healthcare, have continuous learning and innovation ability, and be able to apply new technologies to the optimization and improvement of smart healthcare systems.</p>
Content	<p>Part A. Theoretical teaching (16 contact hours; 14 self-study hours)</p> <p>Chapter 1 Overview of Smart Healthcare (2 contact hours; 2 self-study hours) The concept, development process and current situation of intelligent medical care Architecture and key technologies of intelligent medical treatment Application scenarios and value analysis of intelligent medical treatment</p> <p>Chapter 2 Medical Data Management (4 contact hours; 2 self-study hours) Characteristics, classification and standards of medical data Medical database design and optimization Medical big data processing technology (data cleaning, integration, storage, analysis) Medical data security and privacy protection</p> <p>Chapter 3 Intelligent diagnostic technology (4 contact hours; 4 self-study hours) Application of machine learning in intelligent diagnosis Application of deep learning models (such as convolutional neural network and recurrent neural network) in medical image diagnosis Application of expert system and knowledge graph in disease diagnosis Design and implementation of intelligent diagnostic system</p> <p>Chapter 4 Intelligent diagnostic technology (2 contact hours; 2 self-study hours) The model and architecture of telemedicine Design and implementation of remote consultation system Remote physiological index monitoring technology Application of 5G technology in telemedicine</p>

	<p>Chapter 5 Case study of intelligent medical application (4 contact hours; 4 self-study hours) Analysis of typical cases of smart medical treatment at home and abroad (such as smart hospital, regional medical information platform, family health management system) Requirements analysis and design points of intelligent medical system Implementation and operation challenges of intelligent medical system</p> <p>Part B. Experiment teaching (16 contact hours; 14 self-study hours) Experiment 1: Design and implementation of medical database (4 contact hours; 2 self-study hours) Design hospital information management system database Implement the operation of adding, deleting, modifying and checking medical data Experiment 2: Medical data visualization analysis (4 contact hours; 4 self-study hours) Implement visual presentation of medical data based on Python Conduct simple medical data analysis and mining Experiment 3: Development of intelligent diagnosis model (4 contact hours; 4 self-study hours) A simple disease classification model is implemented based on the machine learning framework Model training and evaluation Experiment 4: Development of telemedicine prototype system (4 contact hours; 4 self-study hours) Realize the remote video consultation function Real-time monitoring and transmission of physiological indicators are realized.</p>
Examination forms	<p>Comprehensive assessment, including:</p> <ol style="list-style-type: none"> 1. Closed-book written test (60%) 2. Regular grades (20%) 3. Laboratory report (20%)
Study and examination requirements	<p>Attend classes on time and actively participate in classroom discussions and interactions Complete the after-class homework and experimental</p>

	<p>tasks independently. The experimental report should record the process, results and analysis in detail</p> <p>The project practice is completed in the form of a group, and complete project documents (requirement analysis, design documents, code, test report, etc.) should be submitted and demonstrated on site</p> <p>Regular grades account for 20%, including homework (10%), experiments (20%) and classroom performance (10%)</p> <p>Experimental report accounts for 20%</p> <p>Closed-book written test at the end of the semester accounts for 60%</p>
Reading list	<p>1. Required books</p> <p>[1] Jiaolin Qiu, Hua Tian, Jie Zheng, Ren Cheng, Lang Qin. Smart Healthcare [M]. Second Edition. Beijing: Tsinghua University Press, 2015.</p> <p>2. Reference books</p> <p>[1] Shi Yinghuan, Li Shumeng. AI and Healthcare [M]. Guangdong: Guangdong Science and Technology Press, 2023.</p> <p>[2] Zhao Wenlong. Introduction to Smart Medicine [M]. Liaoning: Science Press, 2024.</p>
Data of last amendment	June 10, 2025

Web Application Development

Module designation	Web Application Development
Semester(s) in which the module is taught	5 th semester
Person responsible for the module	Lecturer Li Sheng
Language	Chinese
Relation to curriculum	<p>Web Application System Development is a basic course for computer majors. Before learning this course, students must first learn Java programming. JSP technology forms the foundation of Java Web development. Built on Java Servlet and the Java platform, it delivers advantages like separating dynamic from static pages and enabling "write once, run anywhere" across hardware platforms. This technology empowers the creation of secure, cross-platform dynamic websites. The knowledge and techniques covered in this course demonstrate broad development potential both domestically and internationally, with most major websites currently adopting JSP as their primary programming language. A thorough understanding of JSP's technical architecture is essential for mastering J2EE technologies. Solid mastery of JSP technology is crucial for.. The foundation of an efficient Web application architecture.</p>
Teaching methods	<p>Target students: students of Computer Science and Technology.</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,	<p>Total workload = 90 hours</p> <p>Contact hours = 48 hours</p> <p>Self-study hours = 42 hours</p>

self-study hours)	
Credit points	3
Required and recommended prerequisites for joining the module	Advanced mathematics, Data structure, Java programming
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Knowledge: <p>(1) The working principle of Web, the construction of Web environment, the basic of JSP technology, the basic syntax of JSP, and the built-in objects of JSP;</p> <p>(2) JSP database application, JavaBean and Servlet technology and application development theory and method;</p> <p>(3) Filter technology, EL expression and the design method of custom JSP tag library;</p> <p>(4) Performance indicators and evaluation methods of JSP web system.</p> ● Skill: <p>Be able to use the basic principles of JSP Web learned, put forward the design scheme of the application system, analyze the advantages and disadvantages of the proposed scheme in terms of environmental resources, design complexity, system performance, etc., and select the cost-effective design scheme;</p> ● Competence: <p>Be able to correctly analyze the requirements of application projects, carry out detailed design, and master common debugging methods and version control technology. Be able to scientifically and objectively evaluate the feasibility and rationality of project design schemes.</p>
Content	<p>Part A. Theoretical teaching</p> <p>(32 contact hours; 28 self-study hours)</p> <p>Chapter 1 Web fundamentals</p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Common WEB servers; ● IIS Web server configuration; ● Client-side technology; ● Server-side technology;

	<p>Chapter 2 HTML Language Basics (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Basic structure of HTML files; ● Common HTML tags; ● HTML events; ● DIV+CSS layout; ● JavaScript. <p>Chapter 3 JavaWeb development environment construction (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Working principle of Java Web; ● Installation and configuration of Tomcat; ● Configure Tomcat in MyEclipse; ● Create a Web project using MyEclipse; ● Experiment 3. <p>Chapter 4 JSP Technology Foundation (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> ● Introduction to JSP; ● JSP script elements; ● JSP instruction elements; ● JSP action elements; ● Implicit objects of JSP; ● Experiment 4. <p>Chapter 5 JSP Access to Database (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● MySQL database; ● Project Case 1-Online bookstore database creation; ● Access the database using JDBC; ● JDBC driver type; ● Experiment 5 <p>Chapter 6 JavaBean Technology (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● What is JavaBean; ● Use JavaBean in JSP. <p>Chapter 7 Servlet Basics (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● What is Servlet; ● The life cycle of Servlet; ● ServletAPI hierarchy; ● Introduction to main Servlet API; ● Project case 3-Online bookstore background design; ● Project Case 4-Front desk design of online
--	---

	<p>bookstore.</p> <p>Chapter 8 Filter (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Introduction to Servlet filters; ● Servlet filter architecture; ● JSP Chinese garbled problem. <p>Chapter 9 EL and JSTL (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Basic knowledge of EL expression; ● Application examples of EL expressions. <p>Part B. Experiment teaching (16 contact hours; 14 self-study hours)</p> <p>Experiment 1 IIS Web server (2 contact hours; 1 self-study hours) Deploy the server and run it</p> <p>Experiment 2 HTML page design (4 contact hours; 3 self-study hours) Simple login page design</p> <p>Experiment 3 JSP Web environment (2 contact hours; 2 self-study hours) Develop and run JSP programs.</p> <p>Experiment Project 4 JSP dynamic table (4 contact hours; 4 self-study hours) Implement the design of JSP dynamic table.</p> <p>Experiment 5 JDBC database access (4 contact hours; 4 self-study hours) Use JDBC to connect to the database and perform database related operations.</p>
Examination forms	Closed-book written exam
Study and examination requirements	<p>The final examination of this course is closed-book, and the midterm examination is closed-book. The regular performance consists of classroom performance, homework, regular tests and other assessment items. The final examination accounts for 70% of the overall score, and the regular performance accounts for 30%.</p>
Reading list	<p>1. Required books [1] Wang Chunming, Shi Shenghui. JSP Web Technology and Application Tutorial [M]. Second Edition. Micro Course edition. Beijing: Tsinghua University Press, 2018.</p> <p>2. Reference books [1] Tomorrow Technology. Java Web From Entry to</p>

	Proficiency [M]. Beijing: Tsinghua University Press, 2012.
Data of last amendment	Feb, 2024

Innovative Training for Programming Contests

Module designation	Innovative Training for Programming Contests
Semester(s) in which the module is taught	4 th semester
Person responsible for the module	Lecturer Lu Zenan
Language	Chinese
Relation to curriculum	<p>This course is closely aligned with the core curriculum of Computer Science and related majors. It builds upon foundational courses such as Data Structures, Algorithm Design and Analysis, and Programming Fundamentals, providing students with enhanced capabilities in algorithmic thinking, programming proficiency, and problem-solving skills through hands-on training and competitive programming exercises.</p> <p>The course strengthens students' ability to apply knowledge from multiple courses in an integrated manner, fostering advanced algorithm design and innovative solutions to complex problems. It further develops key competencies such as engineering practice, teamwork, project-based collaboration, and creative thinking, directly supporting the program's learning outcomes related to "the ability to analyze complex engineering problems and design effective solutions" and "the capacity for innovation and engineering practice."</p> <p>In addition, the course complements advanced topics such as Introduction to Artificial Intelligence, Big Data Technologies, and Software Engineering, preparing students for future academic research, industry projects, and algorithm engineering roles.</p>

Teaching methods	<p>Target students: Students majoring in Computer Science and Technology.</p> <p>Type of teaching: integration of theory and practice, case-based teaching, blended teaching</p> <p>Contact hour: 32 hours</p> <p>Including:</p> <p>Theoretical teaching: 16 hours</p> <p>Experiment teaching: 16 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 joining the module	Discrete Mathematics, Data Structures
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Module Objectives: <p>This module aims to develop students' algorithmic thinking and problem-solving skills through innovative programming contest training. It enhances their ability to apply advanced algorithms and data structures, fosters creativity and teamwork, and prepares them for participation in high-level programming competitions. The module also encourages students to integrate algorithmic techniques into broader engineering and research contexts.</p> <ul style="list-style-type: none"> ● Skill: <p>Upon completing this module, students will acquire the following skills:</p> <ol style="list-style-type: none"> 1. Ability to implement and optimize advanced algorithms and data structures. 2. Proficiency in designing efficient solutions for complex and open-ended programming problems.

	<p>3. Strong practical programming and debugging abilities under time-constrained conditions.</p> <p>4. Experience in using online judging platforms (e.g., Codeforces, LeetCode, AtCoder) for self-directed learning and practice.</p> <p>5. Team collaboration skills in solving programming tasks during contests and projects.</p> <p>● Competence:</p> <p>Through this module, students will develop the following competences:</p> <p>1. Competence in analyzing complex algorithmic problems and designing innovative solutions.</p> <p>2. Ability to integrate knowledge across different areas of computer science and apply it to real-world problems.</p> <p>3. Capability to work effectively in competitive and collaborative environments.</p> <p>4. Readiness to participate successfully in national and international programming contests.</p> <p>5. Enhanced problem-solving mindset and continuous self-learning competence, supporting long-term career development in technical and research roles.</p>
Content	<p>Part A. Theoretical Teaching (16 contact hours; 14 self-study hours)</p> <p>Chapter 1. Introduction to Competitive Programming (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Overview of programming contests and platforms ● Contest structure, rules, and scoring mechanisms ● Core competencies and preparation strategies <p>Chapter 2. Algorithmic Paradigms (3 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Recursion and divide & conquer ● Greedy methods and optimality ● Dynamic programming: concepts and techniques

	<p>Chapter 3. Essential Data Structures (3 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Arrays, stacks, queues ● Union-Find sets, segment trees ● Binary indexed trees (Fenwick trees) <p>Chapter 4. Graph Algorithms (3 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Graph traversal: DFS and BFS ● Shortest paths: Dijkstra and Bellman-Ford ● Minimum spanning trees: Kruskal and Prim <p>Chapter 5. Number Theory and Combinatorics (3 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Modular arithmetic, Euclidean algorithm ● Combinatorial mathematics: nCr, permutations ● Prime sieves and factorization <p>Chapter 6. String Algorithms and Basic Geometry (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● String matching: KMP, rolling hash ● Tries and prefix matching ● Computational geometry: convex hull, distance <p>Part B. Experimental Teaching (16 contact hours; 14 self-study hours)</p> <p>Unit 1. Setting Up Competitive Programming Environment (2 contact hours; 1 self-study hour)</p> <ul style="list-style-type: none"> ● Using online judges: Codeforces, LeetCode, AtCoder ● Contest IDEs and input/output handling <p>Unit 2. Algorithm Implementation Practice I (3 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Problem sets on greedy, divide & conquer, recursion ● In-class live coding and solution discussion <p>Unit 3. Algorithm Implementation Practice II (3 contact hours; 3 self-study hours)</p>
--	---

	<ul style="list-style-type: none"> ● Problem sets on DP and data structures ● Debugging and optimization techniques <p>Unit 4. Graph and Number Theory Practice (3 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Mixed problem set involving graph search and modular math ● Efficiency analysis and complexity control <p>Unit 5. Mini Team Contest (Simulation) (3 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Realistic 2–3 hour team competition under constraints ● Team strategy, collaboration and stress handling <p>Unit 6. Post-Contest Analysis and Innovation Workshop (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Summary of common errors and optimization paths ● Encouragement of algorithm innovation and open-ended extensions
Examination forms	Assessment
Study and examination requirements	<p>Students are required to complete all assignments independently after each class. Punctuality and attendance are strictly enforced — late arrival, early departure, or absence without prior approval is not permitted. The final grade consists of the following components:</p> <ul style="list-style-type: none"> ● Continuous assessment: 60% <ul style="list-style-type: none"> Assignments: 20% Practical experiments: 25% Periodic evaluation: 15% ● Short paper: 40%
Reading list	<p>1. Required Reading</p> <p>[1] Antti Laaksonen. Guide to Competitive Programming: Learning and Improving Algorithms</p>

	<p>Through Contests [M]. Cham: Springer, 2020.</p> <p>2. Recommended Reading</p> <p>[1] Steven S. Skiena. The Algorithm Design Manual (2nd Edition) [M]. London: Springer, 2008.</p> <p>[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms (3rd Edition) [M]. Cambridge: The MIT Press, 2009.</p>
Data of last amendment	June 29, 2025

Cybersecurity

Module designation	Cybersecurity
Semester(s) in which the module is taught	5 th semester
Person responsible for the module	Professor Ren Xiangmin
Language	Chinese
Relation to curriculum	This course serves as a core elective for Computer Science and Technology majors, combining theoretical instruction with hands-on practice. Through its study, students will develop a proper understanding of cybersecurity principles, cultivate a sense of responsibility in safeguarding national cybersecurity, comply with relevant laws and regulations, gain comprehensive knowledge of fundamental cybersecurity theories and practical technologies, master essential methods for network system security protection, enhance awareness of cybersecurity defense, and acquire skills in cybersecurity analysis and design.
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	Computer Composition Principle, Database Principle And Application, Operating System, Computer Network

<p>Module objectives/intended learning outcomes</p>	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Knowledge: <ol style="list-style-type: none"> 1. Master the basic knowledge and principles of computer network security. 2. Establish a correct view of network security. Without network security, there is no national security. Have a sense of mission and responsibility to safeguard national network security, and consciously abide by network security laws and regulations. ● Skill: <ol style="list-style-type: none"> 1. Familiar with the basic network security technology, including intrusion detection and intrusion prevention, cryptography, encryption technology, identity authentication, firewall technology, computer virus detection and defense, operating system and database security, etc. 2. Master network security detection technology. ● Competence: <ol style="list-style-type: none"> 1. Be able to have network security management ability and carry out network security assessment by applying network security standards. 2. Ability to continue learning and team work spirit. 3. Have the spirit of craftsmanship, social responsibility and good professional ethics.
<p>Content</p>	<p>Part A. Theoretical teaching (32 contact hours; 28 self-study hours)</p> <p>Chapter 1 Basics of Network Security (2 class hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● The concept, technical characteristics, research objectives and content of network security ● Analysis of threats and factors facing the network ● Network security model, network security guarantee system and key technologies ● We will protect laws and regulations on network security technology ● Safety technology assessment criteria and guidelines ● Principles and steps of network security design and construction <p>Chapter 2 Network Security Technology Foundation (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Network protocol security

	<ul style="list-style-type: none"> ● Cyber security hierarchy ● Security services and security mechanisms ● Virtual private network VPN technology ● Wireless LAN security ● Common network commands <p>Chapter 3 Network Security Management Technology (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Network security management concept, content and function ● Safety management principles and systems, protection system specifications and policies ● Network security management technology, strategy and host network protection, network security management solutions ● Technology related to physical security protection ● Software security related technology <p>Chapter 4 Hacker Defense and Intrusion Detection (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● The purpose and steps of the hacker attack ● Commonly used attack methods by hackers ● Measures to prevent hackers ● Hackers attack processes and defend against hackers ● The concept, function, characteristics, classification, detection process and common detection methods of intrusion detection system <p>Chapter 5 Cryptography and Encryption Technology (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Related concepts of cryptography, cryptography and cryptosystems ● Data and network encryption ● Cryptography and key management ● Practical encryption technology, including: symmetric/asymmetric encryption, one-way encryption technology, wireless network encryption technology, practical comprehensive encryption method, encryption high and new technology and development ● Basic concepts and usage of data compression <p>Chapter 6 Authentication and Access Control (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● The concept, types and methods of identity
--	---

	<p>authentication technology</p> <ul style="list-style-type: none"> ● Login authentication and authorization management ● Digital signature technology and application ● Access control technology and application ● Security audit technology and application <p>Chapter 7 Computer and Mobile Phone Virus Prevention (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● History and trends of computer virus development ● Definition, classification, characteristics, structure, mode of transmission and production of viruses ● Virus detection, removal, protection, viruses and anti-virus development trends ● Malware concepts, classification, protection and removal ● The latest antivirus software application <p>Chapter 8 Firewall Application Technology (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● The concept of firewalls ● Firewall functionality ● Different categories of firewalls ● The main application of firewalls <p>Chapter 9 Operating System and Site Security (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Be safe with Windows operating system and familiar with the security configuration of Windows system ● Security of UNIX operating system, understand UNIX system security configuration ● Security of Linux operating system, master the security configuration of Linux system ● Website security and website security policies ● System and information recovery, master the process of system recovery <p>Chapter 10 Database System Security Technology (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● The concept, composition and security requirements of database system security ● Security framework and characteristics of database system ● There are three aspects of database data protection: security, integrity and concurrency control ● Common database attack methods and can be used
--	---

	<p>in defense practice</p> <ul style="list-style-type: none"> ● Concepts, strategies, and methods for data backup and recovery in a database system ● Network database security management measures <p>Chapter 11 E-commerce Security</p> <ul style="list-style-type: none"> ● (4 contact hours; 4 self-study hours) ● Security requirements for e-commerce ● SSL and SET protocols for e-commerce ● Based on SSL protocol Web, server construction ● Security of mobile e-commerce and WPKI technology <p>Chapter 12 Design of Network Security Scheme (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Cyber security program concept and content ● Safety program objectives, design principles and quality standards ● Requirements analysis and main tasks for the security program ● Security plan analysis and design, security solution case ● Implementation plan and technical support, test report and program writing <p>Part B. Experiment teaching (16 contact hours; 14 self-study hours)</p> <ol style="list-style-type: none"> 1. Network security simulation environment and website security detection (2 contact hours; 2 self-study hours) 2. Vulnerability detection (4 contact hours, 2 contact hours self-study) 3. Encryption software application (4 contact hours, 4 contact hours self-study) 4. Identity authentication and Trojan protection (2 hours, 2 hours self-study) 5. Basic firewall configuration (2 contact hours, 2 contact hours self-study) 6. Operating system and database security (2 contact hours, 2 contact hours self-study)
Examination forms	Open-book written exam
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 (20%) and periodic assessments (10%).</p> <p>The final assessment (closed written test) accounts for 60%.</p>
Reading list	<p>1. Required books</p> <p>[1] Jia Tiejun. Cyber Security Technology and Application [M]. 4th edition. Beijing: China Machine Press, 2022.</p> <p>2. Reference books</p> <p>[1] Ying Liyou. Cyber Security Fundamentals [M]. 2nd edition. Beijing: China Industry and Information Technology Publishing Group, 2017.</p> <p>[2] William Stallings, Bai Guoqiang. Cyber Security Fundamentals: Application and Standards [M]. 5th edition. Beijing: Tsinghua University Press, 2014.</p> <p>[3] Joseph Migga Kizza. Introduction to Computer Network Security [M]. Beijing: Electronic Industry Press, 2012.</p>
Data of last amendment	June 10, 2025

Scientific and Technical Writing

Module designation	Scientific and Technical Writing
Semester(s) in which the module is taught	6th semester
Person responsible for the module	Associate professor Xu Yan
Language	Chinese and English
Relation to curriculum	This is a compulsory major course designed for undergraduate students in computer science and technology. It aims to cultivate students' academic expression and scientific writing abilities, laying the foundation for future thesis writing, project documentation, and academic publication. The course also supports the development of scientific thinking, logical organization, and research dissemination skills, thereby enhancing comprehensive professional competence.
Teaching methods	Target students: Undergraduate students in computer science and related majors Type of teaching: Theoretical teaching and practical training Contact hours: 32 Including: Theoretical teaching: 16 hours Practical (writing and editing) sessions: 16 hours Computer practice: 0 hours Class size: 40–60 students
Workload (incl. contact hours, self-study hours)	Total workload: 60 hours Contact hours: 32 Self-study hours: 28
Credit points	2
Required and recommended prerequisites for joining the module	None
Module objectives/intende	Learning outcomes: ● Knowledge:

<p>d learning outcomes</p>	<p>Understand the structure, purpose, and language characteristics of scientific papers.</p> <p>Grasp the norms and ethical principles of academic publishing.</p> <ul style="list-style-type: none"> ● Skill: Write different sections of a scientific paper clearly and logically. Apply academic conventions in formatting, referencing, and language use. Review and revise scientific writing through peer feedback and editing. ● Competence: Independently complete scientific writing assignments based on real research topics. Demonstrate academic responsibility and research integrity. Prepare for thesis writing and participation in academic publication.
<p>Content</p>	<p>Part A. Theoretical teaching (16 contact hours; 14 self-study hours)</p> <p>Chapter 1 Scientific Writing Overview (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Purpose and significance of scientific writing ● Types and categories of scientific papers ● Writing process and publishing flow <p>Chapter 2 Structure and Language of Scientific Papers (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● IMRaD structure: Introduction, Methods, Results, and Discussion ● Logical organization of content ● Tense usage and voice in academic writing ● Sentence clarity and cohesion <p>Chapter 3 Writing the Abstract and Introduction (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Types and components of abstracts (descriptive, informative) ● Key elements of a strong introduction ● Stating background, problem, and purpose ● Effective use of citations in introductions <p>Chapter 4 Literature Review and Methodology Sections</p>

	<p>(4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Synthesizing existing research ● Avoiding plagiarism in literature reviews ● Describing research methods clearly and precisely ● Reproducibility and clarity in methodology <p>Chapter 5 Academic Ethics and Publishing Norms</p> <p>(2 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Academic integrity and authorship criteria ● Ethical issues in publishing (e.g., duplicate submission, data fabrication) ● Peer review process and reviewer expectations ● Choosing appropriate journals <p>Part B. Experiment teaching</p> <p>(16 contact hours; 14 self-study hours)</p> <p>Experiment 1 Abstract and Title Writing Exercise</p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Practice writing clear, concise titles ● Draft structured abstracts based on sample research topics ● Peer comparison and feedback <p>Experiment 2 Drafting Introduction and Methods</p> <p>(4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Complete an introduction section for a chosen topic ● Write methodology based on hypothetical or real data ● Instructor and peer review <p>Experiment 3 Writing Results and Discussion</p> <p>(4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Formatting results: text, tables, and figures ● Interpreting results without overstating conclusions ● Comparing with related work ● Structuring the discussion section logically <p>Experiment 4 Reference Formatting Practice</p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Using reference managers (e.g., EndNote, Zotero) ● Applying APA/IEEE/Chicago citation styles ● Practice correcting reference list errors <p>Experiment 5 Peer Review and Editing</p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Reading and critiquing peers' work ● Giving constructive written feedback ● Implementing edits and tracking revisions <p>Experiment 6 Final Paper Submission and Reflection</p>
--	---

	<p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Assembling final draft of the scientific paper ● Preparing a reflection report on the writing process ● Instructor review and closing evaluation
Examination forms	Portfolio and in-class performance
Study and examination requirements	<ul style="list-style-type: none"> ● After-class assignments must be completed independently after each session ● Attendance is mandatory; no late arrivals, early departures, or unexcused absences ● Continuous assessment (assignments, participation, revisions): 60% ● Final portfolio (complete academic paper + revision notes): 40%
Reading list	<p>1. Required books</p> <p>[1] Li Ping. Scientific Writing for Researchers [M]. Beijing: Tsinghua University Press, latest edition.</p> <p>2. Reference books</p> <p>[1] Day, Robert A., and Gastel, Barbara. How to Write and Publish a Scientific Paper [M]. Cambridge: Cambridge University Press, latest edition.</p> <p>[2] Swales, John M., and Feak, Christine B. Academic Writing for Graduate Students [M]. Ann Arbor: University of Michigan Press, latest edition.</p>
Data of last amendment	March 2023

Fundamentals of Hadoop

Module designation	Fundamentals of Hadoop
Semester(s) in which the module is taught	4 th semester
Person responsible for the module	Lecturer Zilong Xu
Language	Chinese
Relation to curriculum	This course is an elective course for undergraduate students majoring in Data Science and Big Data Technology. The aim of the course is to enable students to understand and master the content of four areas, namely the origin and characteristics of big data systems, the architectural design and functional objectives of big data systems, the development of big data system programmes, and the analysis of enterprise big data cases. Additionally, the course utilises hands-on laboratory sessions to enhance students' practical skills in big data development.
Teaching methods	Target students: students of Computer Science Type of teaching : theoretical teaching, experiment teaching Contact hour: 48 Including: Theoretical teaching: 16 hours Experiment teaching: 16 hours Computer practice: 0 hours Size of class: 40-60 students
Workload (incl. contact hours, self-study)	Total workload: 60 hours Contact hours: 32 hours Self-study hours: 28 hours

hours)	
Credit points	2.5
Required and recommended prerequisites for joining the module	Linear Algebra, Probabilistic Theory, Java Programming
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> • Knowledge: <ol style="list-style-type: none"> 1. Understand and master the origins and characteristics of big data systems. 2. Understand and master the architectural design and functional objectives of big data systems. 3. Understand and master big data system program development 4. Understand and master enterprise big data case analysis • Skill: <ol style="list-style-type: none"> 1. Students should master the use of HDFS. 2. MapReduce Development. 3. Development of the HBase database. 4. Hive Data Warehouse Development. 5. Big data case studies, etc. • Competence: <ol style="list-style-type: none"> 1. Students should be able to construct methods for solving practical problems using big data analysis technology. 2. Practise your ability to flexibly apply classic algorithms. 3. Inspired from classical algorithms, we cultivate the ability to identify and solve problems, as well as innovative thinking skills.

Content	<p>Part A. Theoretical teaching (16 contact hours; 14 self-study hours)</p> <p>Chapter 1 Overview of big data (2 contact hours; 2 self-contact hours)</p> <ul style="list-style-type: none"> ● Understand the concept, characteristics, measurement units, and types of big data. ● Understand the concept, characteristics, measurement units, and types of big data. ● Understand the design concepts, design objectives, and design principles of big data systems. ● Understand the overall logical architecture design and operational logic of big data systems, as well as the current mainstream architecture of big data systems. <p>Chapter 2 Big Data Application Development Approach and Environment Configuration (2 contact hours; 2 self-contact hours)</p> <ul style="list-style-type: none"> ● Master the development process of read and write operations in big data system applications. ● Mastering big data analysis development techniques and approaches ● Mastering Big Data Hadoop Environment Configuration <p>Chapter 3 HDFS distributed file system (2 contact hours; 2 self-contact hours)</p> <ul style="list-style-type: none"> ● Understand HDFS design objectives and basic concepts. ● Understand HDFS design objectives and basic concepts. ● Master program development operations for HDFS, including directory pipelines, file listings, reading, importing, and exporting. <p>Chapter 4 MapReduce Distributed programming (4 contact hours; 2 self-contact hours)</p> <ul style="list-style-type: none"> ● Understand the design philosophy and basic concepts of MapReduce.
---------	---

	<ul style="list-style-type: none"> ● Understanding the system architecture, job execution mechanism, and key technologies of MapReduce. ● Master the customisation of MapReduce data types and their usage. ● Master MapReduce development and customise input and output data formats. ● Master the development of using entire files in the HDFS file system as input data. ● Master the development of using MapReduce to aggregate small files into one large file. ● Master the development of compressed data processing programs. ● Master the development of multi-data source connections, including Map-side development and Reduce-side development. ● Master the use of Hadoop global parameters and global files. <p>Chapter 5 Hbase distributed database (2 contact hours; 2 self-contact hours)</p> <ul style="list-style-type: none"> ● Understand the design objectives and basic concepts of the HBase distributed database. ● Understand the logical and physical architecture of HBase. ● Master the operation of HBase distributed database shell commands. ● Master the development of the HBase database system, including creating tables, deleting tables, and querying all tables. ● Master the development of the HBase database system, including inserting records, querying data, combining queries, modifying and deleting records, etc. <p>Chapter 6 Hive data warehouse development (2 contact hours; 2 self-contact hours)</p> <ul style="list-style-type: none"> ● Understand the working principles and characteristics of the Hive data warehouse;
--	---

	<ul style="list-style-type: none"> ● Understand the Hive architecture design, including data types, data storage methods, and query methods; ● Master the HQL language syntax of the Hive data warehouse system; ● Master HQL operations for creating, viewing, modifying, and deleting tables; ● Master using HQL statements to import files from HDFS into the data warehouse; ● Master the use of partitioned tables, bucket tables, and external tables; ● Master HQL operations such as join queries, subqueries, and creating views; ● Master connecting to the Hive data warehouse for data queries. <p>Chapter 7 Mahout algorithm (2 contact hours; 2 self-contact hours)</p> <ul style="list-style-type: none"> ● Understand the principles and characteristics of Mahout algorithms; ● Master Mahout configuration and usage. ● Master running a Mahout case study. <p>Part B. Experiment teaching (16 contact hours; 14 self-contact hours)</p> <p>Experiment 1 Hadoop environment configuration (2 contact hours; 2 self-contact hours)</p> <ul style="list-style-type: none"> ● JDK installation, SSH password-less login, environment variable configuration; Hadoop pseudo-distributed deployment; core configuration file descriptions (core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml) <p>Experiment 2 HDFS distributed file system (2 contact hours; 2 self-contact hours)</p> <ul style="list-style-type: none"> ● Start Hadoop, use HDFS Shell to complete file uploads/downloads, view directories, set permissions, and view the web interface; analyse data block distribution.
--	--

	<p>Experiment 3 MapReduce distributed programming (4 contact hours; 4 self-contact hours)</p> <ul style="list-style-type: none"> ● WordCount Mini-Program: Write a WordCount program (Java), run it, and view the task log and execution plan; understand the input and output formats. ● Log Analysis: Use MapReduce to process web access logs, count user visits, popular URLs, etc.; use Combiner and Partitioner. <p>Experiment 4 Hbase distributed database (4contact hours; 4 self-contact hours)</p> <ul style="list-style-type: none"> ● Master the development of the HBase database system, including inserting records, querying data, combining queries, modifying, and deleting records. <p>Experiment 5 Hive data warehouse development (4 contact hours; 4 self-contact hours)</p> <ul style="list-style-type: none"> ● Hive Basic Usage Experiment: Start Hive, create external tables and internal tables; import data from HDFS; run HiveQL queries and view statistics results.
Examination forms	Comprehensive assignment
Study and examination requirements	<p>After-class assignment shall be done independently by students after each class.</p> <p>No late arrivals, no early departures, and no unauthorized absences.</p> <p>Usual performance (assignments) accounts for 40%.</p> <p>Final assessment (comprehensive assignment) accounts for 60%.</p>
Reading list	<p>1. Required books</p> <p>[1] Ziyu Lin. Big Data Basic Programming, Experiments, and Case Studies Tutorial [M]. Beijing: Tsinghua University Press, 2020.</p>

	2. Reference books [1] Tom White. Hadoop Authoritative Guide [M]. Translator: Hai Wang, Dong Hua, Yu Liu, Yuehai Lv. Beijing: Tsinghua University Press, 2017.
Data of last mendment	July 7, 2025

Fundamentals of Spark

Module designation	Fundamentals of Spark
Semester(s) in which the module is taught	4 th semester
Person responsible for the module	Lecturer Zhenshen Bao
Language	Chinese
Relation to curriculum	<p>This course is a required professional course for the Data Science and Big Data Technology programme. Spark is a parallel computing framework based on in-memory computing, suitable for parallel computing in data mining and machine learning. Learning Spark big data processing technology is an indispensable step for entering the field of big data application development. This course aims to guide students in mastering the relevant knowledge of Spark big data technology and developing the ability to use Spark. Through this course, students will gain an understanding of the basic principles and usage of the Spark big data computing framework, learn how to install and configure clusters, master Spark programming, and utilise representative Spark components. They will be able to solve complex engineering problems in the big data field using Spark, as well as analyse, design, and implement distributed systems tailored to specific requirements.</p>
Teaching methods	<p>Target students: students of Computer Science.</p> <p>Type of teaching : theoretical teaching, experiment teaching</p> <p>Contact hour: 32 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)	Total workload: 60 hours Contact hours: 32 hours Self-study hours: 28 hours
Credit points	2
Required and recommended prerequisites for joining the module	Database Principles, Operating Systems, Advanced Mathematics, Linear Algebra, Probability Theory and Mathematical Statistics, Python Programming 、 Introduction to Data Science and Big Data Technology
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> • Knowledge: <ol style="list-style-type: none"> Understand and master the basic design and operating principles of Spark, as well as basic knowledge about environment setup and usage. Understand and master the design principles and related operations of Spark SQL, as well as the working mechanisms, differences, and basic implementation steps of Spark Streaming and Structured Streaming. Understand the basic concepts of machine learning and Spark MLlib. • Skill: <ol style="list-style-type: none"> Be able to effectively express big data problems that Spark can solve based on basic knowledge of Spark. Be able to independently set up the Spark environment. Proficient in the fundamentals of Elastic Distributed Dataset (RDD) programming, including RDD creation, operations, persistence, and partitioning. Be able to select appropriate stream computing programmes based on task requirements and design big data application solutions that meet requirements. • Competence: <ol style="list-style-type: none"> It can perform basic processing and analysis of

	<p>large-scale structured and unstructured data.</p> <p>2. Able to select RDD programming technology and implement it according to project requirements.</p> <p>3. Able to reasonably select machine learning models based on user needs and design objectives, and implement them using Spark MLlib.</p>
Content	<p>Part A. Theoretical teaching (16 contact hours; 14 self-study hours)</p> <p>Chapter 1 Overview of big data technology (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● The concept and impact of big data ● Key technologies and computing models of big data ● Introduction to representative software of big data technology <p>Chapter 2 Spark Design and Operating Principles (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Spark design principles; ● Spark installation and operating architecture; ● RDD design and operating principles, deployment modes <p>Chapter 3 Building a big data experimental environment (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Installation of Linux system ● Installation of Hadoop ● Installation of MySQL ● Installation of Kafka ● Installation and usage of Anaconda <p>Chapter 4 Spark Environment Setup and Usage Methods (6 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Install Spark; ● Run code in pyspark; ● Develop Spark standalone applications; ● Set up a Spark cluster environment;

	<ul style="list-style-type: none"> ● Run Spark applications on the cluster. <p>Chapter 5 Spark Environment Setup and Usage Methods</p> <p>(10 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Install Spark; ● Run code in pyspark; ● Develop Spark standalone applications; ● Set up a Spark cluster environment; ● Run Spark applications on the cluster. <p>Chapter 6 Spark SQL</p> <p>(8 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Introduction to Spark SQL; ● Overview of DataFrame; ● Creation, saving, and common operations of DataFrame; ● Conversion from RDD to DataFrame; ● Spark SQL database read/write <p>Chapter 7 Spark Streaming</p> <p>(4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Introduction to stream computing; ● Introduction to Spark Streaming; ● DStream operations. <p>Chapter 8 Structured Streaming</p> <p>(4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Overview of Structured Streaming; ● Basic steps for writing Structured Streaming programs; ● Input sources for Structured Streaming programs; ● Output operations for Structured Streaming programs <p>Chapter 9 Spark MLlib</p> <p>(8 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Introduction to Spark MLlib; ● Machine learning workflow in Spark; ● Feature extraction, transformation, and selection based on Spark MLlib; ● Classification and regression based on Spark
--	--

	<p>MLlib.</p> <p>Part B. Experiment teaching (16 contact hours; 14 self-study hours)</p> <p>Experiment 1 Spark environment configuration (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Master the use of Vim, master the installation of JDK, master the method of setting up the Spark environment, be familiar with the basic methods of starting the Spark environment, using PySpark, and writing and running Spark using Jupyter Notebook. <p>Experiment 2 Basic RDD Programming Practice (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Master the creation of RDDs and pair RDDs, and use pair RDDs for counting. Learn how to write programs using Spark RDD programming to count words in a file. ● Master the creation of RDDs and pair RDDs, and use pair RDDs for counting. Learn how to write programs using Spark RDD programming to count words in a file. <p>Experiment 3 Spark SQL Programming Beginner's Practice (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Use Spark SQL to calculate the average age of employees: Load data from the employee.json file and use Spark SQL's selectExpr to directly calculate the average age. <p>Experiment 4 Spark Machine Learning Library MLlib Programming Practice (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Linear regression prediction of house prices: Predict house prices (labels) based on house area (features).
--	---

Examination forms	Comprehensive assignment
Study and examination requirements	<p>After-class assignment shall be done independently by students after each class.</p> <p>No late arrivals, no early departures, and no unauthorized absences.</p> <p>Usual performance (assignments) accounts for 40%.</p> <p>Final assessment (comprehensive assignment) accounts for 60%.</p>
Reading list	<p>1. Required books</p> <p>[1] Ziyu Lin. Spark Programming Basics [M]. Python Edition 2nd Edition. Beijing: Posts & Telecom Press, 2024.</p> <p>2. Reference books</p> <p>[1] Ziyu Lin. Principles and Applications of Big Data Technology-Concepts. Storage. Processing. Analysis. and Applications [M]. 4th Edition. Beijing: Posts & Telecom Press, 2024.</p> <p>[2] Fang Xiao. Liangjun Zhang. Spark Big Data Technology and Applications [M]. 2nd Edition. Beijing: Posts & Telecom Press, 2022.</p>
Data of last amendment	July 7, 2025

Big Data Development Technologies

Module designation	Big Data Development Technologies
Semester(s) in which the module is taught	6th semester
Person responsible for the module	Teaching assistant Cai Chengfei
Language	Chinese
Relation to curriculum	This is a professional core course for undergraduate students majoring in Computer Science and Technology. It emphasizes the mastery of big data development frameworks and programming paradigms. The course guides students through the application of distributed computing, data storage, and processing technologies. It provides theoretical understanding and hands-on training in the development and deployment of large-scale data processing systems.
Teaching methods	Target students: Undergraduate students majoring in Computer Science and Technology Type of teaching: Theoretical and experimental teaching Contact hours: 32 Theoretical teaching: 16 hours Experimental teaching: 16 hours Computer practice: Included in experiments Class size: 40–60 students
Workload (incl. contact hours, self-study hours)	Total workload: 60 hours Contact hours: 32 Self-study hours: 28
Credit points	2
Required and recommended prerequisites for joining the module	<ul style="list-style-type: none"> ● Introduction to Computer Science ● Database Principles and Applications
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Knowledge: <ol style="list-style-type: none"> 1. Understand the architecture and ecosystem of big data technologies. 2. Master the principles and usage of distributed

	<p>frameworks like Hadoop and Spark.</p> <p>3. Comprehend the foundations of data storage, processing, and scheduling in distributed systems.</p> <p>● Skill:</p> <p>1. Use development tools to build and debug big data programs.</p> <p>2. Implement data analysis tasks using Hadoop and Spark frameworks.</p> <p>3. Design basic big data pipelines and perform data transformation operations.</p> <p>● Competence:</p> <p>1. Analyze and solve real-world data processing problems in distributed environments.</p> <p>2. Work in teams to complete big data application projects.</p> <p>3. Demonstrate adaptability to evolving technologies and self-driven learning.</p>
Content	<p>Part A. Theoretical teaching (16 contact hours; 14 self-study hours)</p> <p>Chapter 1 Overview of Big Data Technology (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Concepts and characteristics of big data ● Common big data application scenarios ● Architecture and key technologies <p>Chapter 2 Distributed Storage with HDFS (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Hadoop Distributed File System (HDFS) design ● Block storage and file replication ● Access and management tools <p>Chapter 3 Data Processing with MapReduce (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Programming model and workflow ● Map, Shuffle, Reduce phases ● Sample applications <p>Chapter 4 Spark Computing Framework (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Spark core concepts and components ● Resilient Distributed Dataset (RDD) ● Basic transformations and actions <p>Part B. Experimental teaching (16 contact hours; 14 self-study hours)</p>

	<p>Experiment 1 Working with HDFS (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● File upload, download, and management ● HDFS shell commands and Java API <p>Experiment 2 MapReduce Programming (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Implementing word count and log analysis ● Job configuration and performance monitoring <p>Experiment 3 Spark Programming Basics (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● RDD creation and operations ● Pair RDDs and aggregation tasks <p>Experiment 4 Mini Project: Big Data Pipeline (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● End-to-end data processing with HDFS and Spark ● Team project and report
Examination forms	Project report, lab performance, and in-class participation
Study and examination requirements	<p>Completion of all experiments and project tasks</p> <p>Mandatory attendance and punctuality</p> <p>Continuous assessment (lab + class performance): 60%</p> <p>Final project (code, documentation, and demo): 40%</p>
Reading list	<p>1. Required books</p> <p>[1] Tang Yuan, Liang Yuan. Big Data Technology and Application Development [M]. Beijing: Tsinghua University Press, 2022.</p> <p>2. Reference books</p> <p>[1] Tom White. Hadoop: The Definitive Guide [M]. O'Reilly Media, 2015.</p> <p>[2] Holden Karau et al. Learning Spark [M]. O'Reilly Media, 2015.</p>
Data of last amendment	March 2023

Development of Applications Based on Large AI Models

Module designation	Development of Applications Based on Large AI Models
Semester(s) in which the module is taught	6 th semester
Person responsible for the module	Lecturer Liu Shuai
Language	Chinese
Relation to curriculum	<p>AI Large Model Application Development is an elective course for Computer Science and Technology majors. This course focuses on cutting-edge artificial intelligence technologies, systematically introducing the fundamental principles, development frameworks, and industry applications of large models. It aims to explain the basic working mechanisms and practical implementations of large language models, with particular emphasis on their usage methods and application scenarios. The curriculum begins with an introduction to natural language processing fundamentals, followed by an exploration of the evolution of language models. Subsequent sections cover parameter configuration, operational methods, performance evaluation, as well as open-source models (such as Alibaba Qianwen) and closed-source models (including various GPT variants), along with their applications in foreign language learning and scientific research. Finally, the application of large models in various fields of daily life and practical scenarios will be briefly introduced. Finally, the ethical and legal issues of large models and their future development direction will be briefly introduced.</p>

Teaching methods	<p>Target students: students of Computer Science.</p> <p>Type of teaching: theoretical teaching, experiment teaching</p> <p>Contact hour: 32 hours 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 joining the module	Python Programming, Data Structures, Algorithm Design and Analysis, Introduction to Artificial Intelligence, Machine Learning
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> • Knowledge: <ol style="list-style-type: none"> 1. Understand the basic principles of large language models (LLMs), including the core mechanisms of Transformer architectures (such as BERT, GPT, T5, etc.). 2. Master the characteristics, parameter setting and capability evaluation methods of open source and closed source large models (such as Ali Qianwen and GPT series). 3. Familiar with typical application scenarios of large models in language learning, scientific research, code generation and other fields. 4. Understand the ethical and legal issues involved in large models and future development trends. • Skill: <ol style="list-style-type: none"> 1. Be able to call mainstream large language models

	<p>through web pages or API, and complete basic tasks (such as text generation, translation, question answering, etc.).</p> <ol style="list-style-type: none"> 2. Master the Prompt Engineering technique to optimize the output of the model. 3. Have the ability to debug large model parameters (such as temperature and top-k sampling) to adapt to different requirements. 4. Evaluate the applicability of the large model by analyzing its actual performance through experiments. <p>• Competence:</p> <ol style="list-style-type: none"> 1. Design and implement innovative application solutions based on large models according to the needs of the field. 2. Critically analyze the limitations of large models and weigh their technical advantages and potential risks. 3. Use large model tools to improve efficiency and solve interdisciplinary problems in team collaboration.
Content	<p>Part A. Theoretical teaching (16 contact hours; 14 self-study hours)</p> <p>Chapter 1 Course description (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Course Introduction ● Demonstration of the use of large language models ● Language model based on Transformer architecture ● Encoder The principle of the language model BERT ● Decoder The principle of the GPT model ● Encoder-Decoder The principle of model T5 <p>Chapter 2 The foundation of large language models (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● How big is big? ● Algorithms, computing power and data

	<ul style="list-style-type: none"> ● Common open source large language models ● Common closed source large language models ● Specialized domain large language models <p>Chapter 3 How to use large language models (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Use of web version ● API call ● Hint project ● Parameter Settings <p>Chapter 4 Application of Large Language Models (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Application in language learning ● Application in humanities and social sciences research ● Application of code, graph and other aspects <p>Part B. Experiment teaching (16 contact hours; 14 self-study hours)</p> <p>Install related software. (4 contact hours; 2 self-study hours)</p> <p>Large model API call and parameter setting. (6 contact hours; 6 self-study hours)</p> <p>Case study of large model usage. (6 contact hours; 6 self-study hours)</p>
Examination forms	Curriculum design
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 result consists of regular score (20%), experimental score (20%) and final score (60%).</p>
Reading list	<p>1. Required books</p> <p>[1] Zhang Qi, GUI Tao, Zheng Rui, Huang Xuanjing. Large-scale Language Models: From Theory to Practice [M]. Beijing: Electronic Industry Press, 2024.</p> <p>2. Reference books</p>

	<p>[1] Lan Yijie. AI Prompt Engineering: Practical From Zero to Learning Large Language Models Using Prompt Engineering [M]. Beijing: Peking University Press, 2024.</p> <p>[2] Yang Qingfeng. ChatGPT Storm: Big Language Models - Generative AI and AIGC Disruptive Innovation Paradigm [M]. Beijing: Electronic Industry Press, 2023.</p> <p>[3] Liu Cong. ChatGPT Principles and Practical Large Language Model Algorithms [M], Beijing: Machinery Industry Press, 2023.</p>
Data of last amendment	June 29, 2025