

Taizhou University



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

Appendix D

Syllabus - Engineering Application

Department of Computer Science and Technology  
School of Information Engineering

2025

## Contents

Fundamentals of Data Mining.....	1
Software Engineering.....	6
Intelligent Application System Development.....	13
Advanced Java Programming.....	18
Mobile Application Development.....	23
Software Project Management.....	27
Human-Computer Interaction Technology.....	33
Machine Learning.....	38
Digital Image Processing.....	44
Data Visualization Techniques.....	50
Robot Technology and Applications.....	55
English for Computer Science.....	60

## Fundamentals of Data Mining

Module designation	Fundamentals of Data Mining
Semester(s) in which the module is taught	5 <sup>th</sup> semester
Person responsible for the module	Song Zhenyu
Language	Chinese
Relation to curriculum	<p>Data Mining is a core elective course for Computer Science and Technology majors. As an interdisciplinary field that integrates machine learning, statistics, pattern recognition, database technologies, artificial intelligence, statistical methodologies, visualization techniques, and parallel computing, this course equips students with the ability to automatically analyze and extract valuable insights from data. It aims to help learners identify latent information, knowledge, patterns, correlations, and trends within datasets, thereby supporting the interpretation of current behaviors and prediction of future outcomes. Through this program, students will gain a comprehensive understanding of China's national big data strategy. This program equips students with strategic frameworks and technical expertise to build a solid foundation for creating intelligent societies and smart lifestyles. Through hands-on instruction, learners master fundamental concepts, principles, and technologies in data mining, enabling practical application. The curriculum focuses on data-centric topics and technical challenges in knowledge discovery, incorporating extensive case studies and experimental teaching. Students are trained to apply computational thinking in data mining, which stimulates innovative thinking and enhances their practical problem-solving skills during the process. This approach ensures balanced development of both theoretical knowledge and practical competencies.</p>

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: 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</p>
Workload (incl. contact hours, self-study hours)	<p>Total workload = 90</p> <p>Contact hours = 48</p> <p>Self-study hours = 42</p>
Credit points	3
Required and recommended prerequisites for joining the module	Linear algebra, probability and statistics, introduction to artificial intelligence
Module objectives/intended learning outcomes	<p><b>Learning outcomes:</b></p> <ul style="list-style-type: none"> <li>• <b>Knowledge:</b></li> </ul> <ol style="list-style-type: none"> <li>1. Understand the technical definition and business definition, function and application field of data mining, understand the relationship between data mining and knowledge discovery, data query and expert system.</li> <li>2. Familiar with data mining and knowledge discovery processing processes, understand data mining evaluation techniques, including data evaluation and model evaluation methods.</li> <li>3. Master the basic techniques and methods of data mining, including supervised learning techniques-decision tree technology, production rules, artificial neural network technology and statistical analysis methods, as well as unsupervised learning techniques-clustering technology and association analysis methods, understand the basic concepts,</li> </ol>

	<p>methods and techniques of time series.</p> <ul style="list-style-type: none"> <li>• <b>Skill:</b> <ol style="list-style-type: none"> <li>1. Be able to use Weka software tools, apply various data mining algorithms, build classification and clustering models and conduct correlation analysis.</li> <li>2. Use data mining tools to try to solve real problems.</li> </ol> </li> <li>• <b>Competence:</b> <ol style="list-style-type: none"> <li>1. Make students deeply understand the national big data strategy: promote the innovative development of big data technology.</li> <li>2. We should build a digital economy with data as the key element, use big data to improve the modernization of national governance, use big data to promote the protection and improvement of people's livelihood, effectively safeguard national data security, and be good at acquiring, analyzing and using data.</li> <li>3. Be able to propose data mining methods to independently or collaboratively complete new tasks when practical application problems can be solved.</li> </ol> </li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Entering the class</b> (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> <li>● The emergence and development of data mining</li> <li>● Application fields of data mining</li> <li>● The concept and process of data mining</li> <li>● Basic data mining techniques: classification, clustering, association rule analysis, etc.</li> <li>● Basic use of Weka data mining software.</li> </ul> <p><b>Chapter 2 Data preprocessing</b> (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> <li>● Definition of data</li> <li>● Data set definition and type</li> <li>● Data preprocessing (task, method)</li> <li>● Data type conversion and sampling</li> <li>● Principal component analysis and linear</li> </ul>

	<p>discriminant analysis</p> <ul style="list-style-type: none"> <li>● Data visualization</li> </ul> <p><b>Chapter 3 Classification</b> (7 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Definition of classification</li> <li>● The process of classification and evaluation criteria</li> <li>● Decision tree algorithm ID3 and C4.5</li> <li>● Bayesian theory analysis</li> <li>● Naive Bayes classification</li> <li>● Case analysis of real situations</li> </ul> <p><b>Chapter 4 Association rule analysis</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Association rule definition</li> <li>● Support and confidence</li> <li>● Frequent item set</li> <li>● Misconceptions about association rules</li> <li>● Apriori algorithm</li> </ul> <p><b>Chapter 5 Clustering</b> (7 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Definition and typical application of clustering</li> <li>● similarity measurement</li> <li>● Evaluation and verification of clustering</li> <li>● Classification of clustering algorithms</li> <li>● k-means algorithm</li> <li>● The difference between classification and clustering</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <p>In order to help students better understand the principles of data mining, master data mining or machine learning methods, and improve practical skills, we will arrange the following four typical experimental courses:</p> <p>Experiment 1: Data preprocessing (comprehensive, 4 contact hours; 3 self-study hours)</p> <p>Experiment 2: Data classification and analysis (design, 4 contact hours; 3 self-study hours)</p>
--	---

	<p>Experiment 3: Data association rules and cluster analysis (design, 4 contact hours; 3 self-study hours)</p> <p>Experiment 4: Application of artificial neural network technology (comprehensive, 4 contact hours; 5 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 performance accounts for 40%, including homework (10%), regular performance (10%) and experiments (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 Zhenwu. Principles and Implementation of Data Mining Algorithms [M]. 3rd Edition. Beijing: Tsinghua University Press, 2023.</p> <p>[2] Liu Peng, Wang Xiaoxia. Data Mining [M]. 2nd Edition. Beijing: Electronic Industry Press, 2023.</p> <p><b>2. Reference books</b></p> <p>[1] Jiawei Han, Micheline Kamber, Jian Pei [M]. Data Mining: Concepts and Technologies. Beijing: China Machine Press, 2022.</p> <p>[2] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining [M]. Beijing: People's Posts and Telecommunications Press, 2023.</p> <p>[3] Yuan Meiyu, Data Mining and Machine Learning-Weka Application Technology and Practice [M]. 2nd Edition. Beijing: Tsinghua University Press, 2022.</p>
Data of last amendment	June 9, 2025

## Software Engineering

Module designation	Software Engineering
Semester(s) in which the module is taught	6th semester
Person responsible for the module	Professor Ren Xiangmin
Language	Chinese
Relation to curriculum	This course is a required course for computer science and technology majors. Through this course, students will gain a systematic understanding of the principles, methods, and techniques of software engineering and be able to directly apply this knowledge to guide software development.
Teaching methods	<p>Target students: students of Computer Science</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 hour</p> <p>Size of class: 40-60 students</p>
Workload (incl. contact hours, self-study)	<p>Total workload = 90 hours</p> <p>Contact hours = 48 hours</p> <p>Self-study hours = 42 hours</p>



hours)	
Credit points	3
Required and recommended prerequisites for joining the module	Advanced Programming Language, Data Structure, Operating System
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 concepts, principles, technologies, typical methodologies, and management of software projects.</li> <li>2. Be familiar with relevant technical standards, intellectual property rights, industrial policies, and laws and regulations, and apply software engineering theory to software development and other work.</li> </ol> </li> <li>• <b>Skill:</b> <ol style="list-style-type: none"> <li>1. Be able to read and write relevant software engineering technical documents, and possess problem-solving and engineering capabilities.</li> <li>2. Be able to apply engineering management principles and economic decision-making methods to the design, operation, and management of computer application systems.</li> </ol> </li> <li>• <b>Competence:</b> <ol style="list-style-type: none"> <li>1. Craftsmanship, social responsibility and good work ethics.</li> <li>2. Ability to be a continuous learner and a team player.</li> </ol> </li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 32 self-study hours)</p> <p><b>Chapter 1 Software and Software Engineering</b> (2 contact hours, 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Concepts, characteristics and classification of</li> </ul>

	<p>software</p> <ul style="list-style-type: none"> <li>● Software crisis</li> <li>● Definition of software engineering</li> <li>● Basic principles of software engineering</li> <li>● Software development methods</li> <li>● Software engineering tools</li> </ul> <p><b>Chapter 2 Software Process</b> (2 contact hours, 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Software life cycle</li> <li>● Software process model</li> <li>● Application of software process model</li> </ul> <p><b>Chapter 3 Feasibility study and project development plan</b> (2 hours, 2 hours self-study)</p> <ul style="list-style-type: none"> <li>● Project approval overview</li> <li>● The task of feasibility study</li> <li>● Technical feasibility</li> <li>● Operational feasibility</li> <li>● Economic feasibility</li> <li>● Steps of feasibility study</li> <li>● Develop project development plan</li> <li>● Application examples of feasibility study</li> </ul> <p><b>Chapter 4 Structured analysis</b> (2 contact hours, 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● The task, principles, steps and management of requirements analysis</li> <li>● Structured analysis</li> <li>● Methods of structured analysis</li> <li>● Graphical tools for structured analysis</li> <li>● Application examples of structured analysis</li> </ul> <p><b>Chapter 5 Structured Design (2 contact hours)</b> (2 contact hours, 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of software design</li> <li>● Database structure design</li> <li>● User interface design</li> <li>● Structured software design</li> <li>● Be familiar with the relationship between structured</li> </ul>
--	---

	<p>design and structured analysis</p> <ul style="list-style-type: none"> <li>● Architecture design</li> <li>● Interface design</li> <li>● Data design</li> <li>● Process design</li> <li>● Application examples of de-structured design</li> </ul> <p><b>Chapter 6 Object-Oriented Methods and UML</b> (4 contact hours, 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Object-oriented software engineering approach</li> <li>● Unified modeling language UML</li> <li>● Static modeling mechanism</li> <li>● Dynamic modeling mechanism</li> <li>● Describe the mechanism of physical architecture</li> </ul> <p><b>Chapter 7 Object-Oriented Analysis</b> (4 contact hours, 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Know object-oriented analysis methods</li> <li>● Object-oriented modeling</li> <li>● Object-oriented analysis application example</li> </ul> <p><b>Chapter 8 Software Architecture and Design Patterns</b> (2 contact hours, 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Concepts of software architecture</li> <li>● Typical software architecture styles</li> <li>● Software quality attributes</li> <li>● Distributed system architecture</li> <li>● Architecture framework</li> <li>● Design patterns of software systems</li> </ul> <p><b>Chapter 9 Object-Oriented Design (4 contact hours)</b> (4 contact hours, 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Object-oriented design and structured design</li> <li>● The relationship between object-oriented design and object-oriented analysis</li> <li>● The process and rules of object-oriented design</li> <li>● Inspiration rules for object-oriented design</li> <li>● System design</li> <li>● Object design</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>● Application examples of object-oriented design</li> </ul> <p><b>Chapter 10 Software implementation</b> (2 contact hours, 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Programming languages</li> <li>● Programming style</li> <li>● Object-oriented implementation</li> <li>● Software implementation application example</li> </ul> <p><b>Chapter 11 Software Testing and Maintenance</b> (4 contact hours, 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of software testing</li> <li>● Classification of software testing</li> <li>● Test cases</li> <li>● Software testing methods</li> <li>● Black box testing</li> <li>● White box test</li> <li>● General steps of software testing</li> <li>● unit test</li> <li>● Integration testing</li> <li>● System test 10</li> <li>● Acceptance test</li> <li>● Regression testing</li> <li>● Object-oriented software testing</li> <li>● Software debugging</li> <li>● Software test application example</li> </ul> <p><b>Chapter 12 Software Maintenance and Software Engineering Management</b> (2 contact hours, 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic software maintenance techniques</li> <li>● Software estimation method</li> <li>● Software development schedule</li> <li>● Software developer organization</li> <li>● Software development risk management</li> <li>● Software quality assurance</li> <li>● Software configuration management</li> <li>● Software engineering standards and software documentation</li> <li>● Software Process Capability Maturity Model</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>● 10 Software projects management</li> <li>● Know software reuse</li> </ul> <p><b>Part B. Experimental teaching</b> (16 contact hours; 10 self-study hours)</p> <ul style="list-style-type: none"> <li>● Draw system data flow diagram with small online bookstore system and other cases (2 contact hours, 2 self-study hours)</li> <li>● Draw structure diagrams for small online bookstore system and other cases (2 contact hours, 2 self-study hours)</li> <li>● Create use case models based on small online bookstore system and other cases, and draw class diagram, object diagram, state diagram and sequence diagram (4 contact hours, 2 self-study hours)</li> <li>● Draw activity diagram, collaboration diagram, component diagram and deployment diagram for small online bookstore system and other cases (4 contact hours, 2 self-study hours)</li> <li>● Case study of small online bookstore system and Project management (4 contact hours, 2 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>Project performance 10%, course work 10%, experiment 20%.</p> <p>The final written test accounts for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Lü Yunxiang. Software Engineering-Theory and Practice [M]. People's Posts and Telecommunications Press. 2nd edition, 2024</p> <p>[2] Qi Zhichang. Tan Qingping and Ning Hong.</p>

	<p>Software Engineering [M]. Higher Education Press. 4th edition, March 2019</p> <p>[3] Ian Sommerville. translated by Cheng Cheng [M]. Software Engineering. China Machine Press. 9th edition, 2011</p>
Data of last amendment	June 10, 2025

### Intelligent Application System Development

Module designation	Intelligent Application System Development
Semester(s) in which the module is taught	5 <sup>th</sup> semester
Person responsible for the module	Lecturer Xie Danyan
Language	Chinese
Relation to curriculum	<p>This course serves as a specialized elective in the undergraduate curriculum of Computer Science and Technology, functioning to expand and extend the program's educational framework. Designed to meet the requirements of innovative talent cultivation under the New Engineering Education Initiative (NEEI), the course introduces various intelligent application systems using Raspberry Pi as its platform. Through comprehensive instruction on Raspberry Pi fundamentals, GPIO applications, and practical implementations in smart systems, students will master the essential workflow for designing and developing intelligent applications while establishing a holistic understanding of smart system architecture.</p> <p>This course is designed to equip students with the skills to analyze, apply, and design intelligent application systems, laying a solid foundation for their future careers and development. It cultivates hardware design thinking and dialectical reasoning methods, nurturing innovative engineering professionals who possess research capabilities, practical problem-solving skills, humanistic values, and strong social responsibility. The program helps students develop a sense of duty to serve the nation and society through scientific and technological advancement, while fostering correct worldviews, values, and life philosophies. Furthermore, it promotes the spirit of craftsmanship as an essential professional ethic.</p>
Teaching methods	Target students: students of Computer Science and Technology.

	<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)	<p>Total workload = 90 hours</p> <p>Contact hours = 48 hours</p> <p>Self-study hours = 42 hours</p>
Credit points	3
Required and recommended prerequisites for joining the module	Computer Composition Principles, C Language Programming, Python Programming
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> <li>● <b>Knowledge:</b> Understand the design process of intelligent innovation projects, and master the application methods of Raspberry PI and common sensors.</li> <li>● <b>Skill:</b> Through literature research and experiments, I applied raspberry PI for design and development, learned hardware design specifications, got familiar with common hardware design techniques, and got familiar with project cultivation and roadshow process.</li> <li>● <b>Competence:</b> Have the ability of cooperation, coordination and organization management, be able to pay attention to the development trend and application prospect of the discipline and have the spirit of exploration of new technology.</li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Intelligent innovation project design process and common intelligent application system</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Innovative talent training under the background of</li> </ul>



	<p>new engineering education and intelligent application development</p> <p>(1) Background of new engineering education reform</p> <p>(2) The third quality of innovative talents in new engineering</p> <p>(3) Innovation literacy of innovative talents in new engineering</p> <p>(4) Artificial intelligence literacy of innovative talents in new engineering</p> <ul style="list-style-type: none"> <li>● Project learning method</li> </ul> <p>(1) Definition and process of project learning method</p> <p>(2) Characteristics and examples of project learning</p> <ul style="list-style-type: none"> <li>● Introduction of intelligent products</li> </ul> <p>(1) Intelligent product experience</p> <p>(2) Product evaluation</p> <p>(3) Industry analysis</p> <p>(4) Related learning resources</p> <ul style="list-style-type: none"> <li>● Intelligent product design</li> </ul> <p>(1) Intelligent product design process</p> <p>(2) Embedded intelligent product design</p> <p>(3) Intelligent product case</p> <p><b>Chapter 2 Enter the World of Raspberry PI</b></p> <p>(6 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Raspberry PI development platform</li> </ul> <p>(1) Raspberry PI basics</p> <p>(2) Raspberry PI model guide</p> <p>(3) Hardware required for raspberry PI development</p> <p>(4) Master the working scenario of Raspberry PI</p> <ul style="list-style-type: none"> <li>● Learn to select and install a system on raspberry PI (practice)</li> </ul> <p>(1) The available operating system for Raspberry PI</p> <p>(2) Installation of the Raspberry PI system burn tool and configuration of WIFI and SSH</p> <ul style="list-style-type: none"> <li>● Master the method of configuring raspberry PI (operational practice)</li> </ul> <p>(1) Obtain the WIFI IP of Raspberry PI, SSH remote connection, VNC remote desktop setting and resolution setting</p> <p>(2) Installation of the Raspberry PI C compilation environment wiringPi and the Python environment</p> <p><b>Chapter 3 Basic and Application of Raspberry PI Smart Switch</b></p>
--	---

	<p>(4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Exercise 1: Application of dual-color LED, RGB LED, seven-color LED flashing, active buzzer and passive buzzer</li> <li>● Exercise 2: Application of touch button switch, relay module, laser module, reed switch sensor and U-shaped switch sensor</li> </ul> <p><b>Chapter 4 Raspberry PI Application Examples</b> (16 contact hours; 16 self-study hours)</p> <ul style="list-style-type: none"> <li>● Exercise 3: Application of PCF8591, digital temperature sensor, sound sensor, photoresistor sensor, raindrop sensor, flame sensor, smoke sensor and potential sensor</li> <li>● Exercise 4: PS2 joystick, humidity sensor application, LCD display, atmospheric pressure detection application</li> <li>● Exercise 5: The realization of basic movement and trace function</li> <li>● Exercise 6: The implementation of infrared obstacle avoidance and ultrasonic obstacle avoidance functions</li> </ul> <p><b>Chapter 5 Cultivation and Roadshow of Intelligent Application Projects</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Roadshow of intelligent application projects</li> </ul> <p>(1) Project evaluation (2) Business plan (3) Project roadshow</p> <p>2. Cultivation of intelligent projects</p> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <p>Experiment 1 Independent design of intelligent application system (8 contact hours; 7 self-study hours)</p> <ul style="list-style-type: none"> <li>● Master dual-color LEDs, RGB LEDs, seven-color LED flicker effects, active and passive buzzer systems, tactile button switches, relay modules, laser modules, reed switch sensors, U-shaped switch sensors, PCF8591 chips, digital temperature sensors, sound sensors, photoresistor sensors, raindrop detectors, flame sensors, smoke detectors, infrared remote controls, potential sensors, Hall effect sensors, PS2 joysticks, humidity sensors,</li> </ul>
--	--

	<p>touch-sensitive switches, LCD displays, atmospheric pressure detection, ultrasonic distance measurement modules, infrared barrier modules, gyroscope-accelerometer sensors, and timing systems Based on the DS1302 module, track sensor and other components, a set of innovative, practical and popular intelligent application system is designed using raspberry PI.</p> <p>Experiment 2 Debugging and innovative design of Raspberry PI intelligent car (8 contact hours; 7 self-study hours)</p> <ul style="list-style-type: none"> <li>● Complete the installation of raspberry PI smart car chassis, track module, voltmeter, infrared obstacle avoidance module, ultrasonic head (steering gear), and WIFI camera installation (with steering gear pan).</li> <li>● Complete the wiring of the Raspberry PI smart car.</li> <li>● Complete the software installation and debugging of the Raspberry PI intelligent car, and carry out innovative design and modification of the intelligent car.</li> </ul>
Examination forms	Project presentation, business plan, project roadshow
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 accounts for 50%, final assessment accounts for 50%.</p>
Reading list	<p><b>1. Required books</b></p> <p>He Fugui. Maker Robot Practice: Based on Arduino and Raspberry PI [M]. Beijing: China Machine Press, 2018.</p> <p><b>2. Reference books</b></p> <p>[1] Rushi Gajjar. Raspberry PI + Sensor [M]. translated by Hu Xunqiang. Beijing: Machine Industry Press, 2019.</p> <p>[2] Ke Bowen. Raspberry PI Practical Guide [M]. Beijing: Tsinghua University Press, 2015.</p>
Data of last amendment	June 10, 2025

### Advanced Java Programming

Module designation	Advanced Java Programming
Semester(s) in which the module is taught	4 <sup>th</sup> semester
Person responsible for the module	Associate professor Song Zhenyu
Language	Chinese
Relation to curriculum	<p>"Advanced Java Programming" is a core elective course for Computer Science and Technology majors, designed to develop students' capabilities in algorithm design, programming, and debugging. Through systematic coursework, the program enhances problem-solving skills while fostering innovative practices, enabling learners to truly apply knowledge in real-world scenarios. Students will master object-oriented programming principles and learn to develop Java applications using graphical UI components and event handling techniques, laying the foundation for advanced Web development. The curriculum cultivates computational thinking, sparks creative awareness, and elevates students' technical proficiency. In the process of software design, students' practical ability to analyze and solve problems enables their theoretical knowledge and practical skills to develop together, with good scientific literacy and strong social responsibility.</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.	Total workload = 90 hours

contact hours, self-study hours)	Contact hours = 48 hours Self-study hours = 42 hours
Credit points	3
Required and recommended prerequisites for joining the module	Data structure, algorithm design and analysis, Java programming
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 object-oriented programming ideas.</li> <li>2. Master the syntax of Java language, students learn to write standard code, familiar with common programming techniques.</li> <li>3. Master the basic methods of graphical user interface design, and have the ability to analyze, describe and design algorithms of general computer programs.</li> </ol> </li> <li>● <b>Skill:</b> <p>Enable students to initially have the basic ability of an excellent software developer, be able to write a certain scale of applications, and understand and abide by engineering ethics and norms in computer engineering practice.</p> </li> <li>● <b>Competence:</b> <p>Cultivate students' spirit of cooperation, coordination and organization and management ability, pay attention to the development trend and application prospect of the subject, pay attention to cultivate students' spirit of inquiry into new technology, and have good scientific literacy and social responsibility.</p> </li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Exception Handling</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● The concept of anomaly and the causes of anomaly;</li> <li>● Exceptional inheritance relationship diagram;</li> <li>● Distinguish the types of exception handling: mandatory and exempt exceptions;</li> <li>● Exception handling model: declare, throw and catch exceptions;</li> </ul>

	<ul style="list-style-type: none"> <li>● Get information from the exception object;</li> <li>● Master the finally clause;</li> <li>● Learn to create custom exception classes.</li> </ul> <p><b>Chapter 2 Input Output and File Handling of Java Language</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● File byte stream and file character stream;</li> <li>● Buffer flow, random flow, array flow, data flow, object flow;</li> <li>● InputStream and OutputStream;</li> <li>● Basic classes for handling character streams: Reader and Writer;</li> <li>● File management: File;</li> <li>● Classification of JAVA IO streams.</li> </ul> <p><b>Chapter 3 Multithreading</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Introduction to threads;</li> <li>● Create a multithreaded program;</li> <li>● The life cycle of threads;</li> <li>● Thread operation;</li> <li>● Thread synchronization.</li> </ul> <p><b>Chapter 4 Generic and Container Classes</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Type parameters;</li> <li>● Generic classes, generic methods and generic interfaces;</li> <li>● Generic restrictions and generic wildcards;</li> <li>● Container traversal (also called iteration);</li> <li>● List interface List and two implementation classes;</li> <li>● Set interface and two implementation classes;</li> <li>● The mapping interface Map and two implementation classes.</li> </ul> <p><b>Chapter 5 Annotations, Reflections, Internal Classes, and Anonymous Internal Classes Compare to the Lambda expression</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Class loader;</li> <li>● Concept of reflection;</li> <li>● Reflect to obtain object information;</li> <li>● Create and access objects;</li> <li>● Notes;</li> <li>● Internal classes;</li> </ul>
--	---

	<ul style="list-style-type: none"> <li>● Anonymous;</li> <li>● Lambda expressions.</li> </ul> <p><b>Chapter 6 Drawing and Animation Programming</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Shape class;</li> <li>● Transition animation;</li> <li>● Keyframe, key value and interpolator;</li> <li>● Timeline animation.</li> </ul> <p><b>Chapter 7 Java Database Programming</b> (6 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● JDBC principles;</li> <li>● JDBC programming steps;</li> <li>● Exceptions and transactions in JDBC;</li> <li>● Problems caused by JDBC transaction concurrency and transaction isolation levels.</li> </ul> <p><b>Chapter 8 Network Programming</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Network foundation;</li> <li>● Network programming in JAVA;</li> <li>● Network programming based on TCP protocol;</li> <li>● Network programming based on UDP protocol.</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. Experiment 1: Input/output and file processing of Java language. (4 contact hours; 2 self-study hours)</li> <li>2. Experiment 2: Generic and Container Class Application. (4 contact hours; 4 self-study hours)</li> <li>3. Experiment 3: Implementing user table operations using JDBC. (4 contact hours; 4 self-study hours)</li> <li>4. Experimental 4: Application of Network Programming. (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>Comprehensive score = formative evaluation score (40%) + final evaluation score (60%)</p> <p>Formative evaluation score = classroom performance × 20% + course work × 20% + class notes × 20% + experiment × 40%</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 29, 2025



### Mobile Application Development

Module designation	Mobile Application Development
Semester(s) in which the module is taught	4 <sup>th</sup> semester
Person responsible for the module	Lecturer Tan Lixing
Language	Chinese
Relation to curriculum	This course aims to equip Computer Science and Technology students with Android mobile application development skills through mastering fundamental Android UI techniques, cultivating their ability to analyze and solve problems using mobile internet technologies, and establishing a programming foundation for subsequent advanced Android studies. Upon completion, students will understand Android application design principles, configure Eclipse ADT environments, comprehend Android program structures with debugging proficiency, skillfully utilize basic UI controls and five major layout types, perform SQLite database operations, implement XML/JSON data exchange formats, master core Activity components, and facilitate component communication using Intents.
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 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	Introduction to Computer Science, Java Programming
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.Understand fundamental principles of Android mobile application design;</li> <li>2.Comprehend Android program structure and debugging methodologies;</li> <li>3.Recognize core functions of Android's Activity component;</li> <li>4.Identify characteristics of XML/JSON data exchange formats;</li> </ol> </li> <li>● <b>Skill:</b> <ol style="list-style-type: none"> <li>1.Configure Eclipse ADT integrated development environment;</li> <li>2.Debug Android programs proficiently;</li> <li>3.Implement SQLite database operations;</li> <li>4.Utilize Android's five major layout types;</li> <li>5.Apply basic UI controls;</li> <li>6.Employ Intents for component communication.</li> </ol> </li> <li>● <b>Competence:</b> <ol style="list-style-type: none"> <li>1.Integrate XML/JSON data formats in mobile solutions;</li> <li>2.Combine layouts and controls to design functional UIs;</li> <li>3.Architect applications using Activity lifecycle management;</li> <li>4.Develop database-backed mobile functionalities;</li> <li>5.Diagnose and resolve Android runtime issues.</li> </ol> </li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Android Platform Overview</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Introduction to Android Platform</li> <li>● Android Development Environment Setup</li> <li>● Debugging &amp; Deployment of Android Applications</li> </ul> <p><b>Chapter 2 First Look at Android Applications</b> (4 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Structure of Android Applications</li> <li>● Android Application Lifecycle</li> </ul>

	<ul style="list-style-type: none"> <li>● Four Major Components of Android Applications</li> </ul> <p><b>Chapter 3 Working with Basic UI Components in Android</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Android UI Fundamentals</li> <li>● Text Display Components</li> <li>● Button Components</li> <li>● Selection Components</li> <li>● List-Based Components</li> <li>● Image Display Components</li> </ul> <p><b>Chapter 4 Android Layout Managers</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● LinearLayout</li> <li>● AbsoluteLayout</li> <li>● FrameLayout</li> <li>● RelativeLayout</li> <li>● TableLayout</li> <li>● GridLayout</li> </ul> <p><b>Chapter 5 Working with SQLite Databases</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● SQLiteDatabase Class</li> <li>● Core Database Operations</li> <li>● SQLite Management Tools</li> <li>● SQLiteOpenHelper Class</li> </ul> <p><b>Chapter 6 Data Interchange Formats</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● XML</li> <li>● JSON</li> </ul> <p>● Chapter 7 Working with Activities (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Activity States</li> <li>● Activity Lifecycle</li> <li>● Common Activity Operations</li> </ul> <p><b>Chapter 8 Inter-Component Communication with Intent</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Introduction to Intent Objects</li> <li>● Anatomy of an Intent</li> <li>● Intent Resolution</li> <li>● Data Transfer via Intent</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p>
--	---

	<p>1. Android UI Implementation. (4 contact hours; 3 self-study hours)</p> <p>2. Android Layout Development. (4 contact hours; 3 self-study hours)</p> <p>3. SQLite Database Integration. (4 contact hours; 4 self-study hours)</p> <p>4. Inter-Activity Data Transfer via Intent. (4 contact hours; 4 self-study hours)</p>
Examination forms	Comprehensive assignment
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 40%, including assignments (10%), experiment (10%), stage assessment (20%).</p> <p>The final assessment (comprehensive project) constitutes the remaining 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Zhu Fengshan, Zhang jianjun. Advanced Android Mobile Application Development [M]. Beijing: Tsinghua University Press., 2022.</p> <p><b>2. Reference books</b></p> <p>[1] Zhong yuansheng, Gao chengzhen. Classic Android Programming Case Studies [M]. Beijing: Tsinghua University Press, 2021.</p> <p>[2] Deitel P, Deitel H. Android how to Program [M]. London: Pearson, 2016.</p>
Data of last amendment	June 29, 2025

## Software Project Management

Module designation	Software Project Management
Semester(s) in which the module is taught	6 <sup>th</sup> semester
Person responsible for the module	Professor Ren Xiangmin
Language	Chinese
Relation to curriculum	This course employs engineering management concepts, principles, and methodologies to comprehensively manage core elements of software development projects, including cost, schedule, and quality. Through this program, students will acquire fundamental knowledge of software project management while enhancing practical skills. The curriculum covers essential processes for developing and tracking software project plans, including scope planning, cost estimation, schedule scheduling, quality assurance, resource allocation, team coordination, risk mitigation, and contract management. Students will also learn how to implement effective monitoring and control during project execution, ultimately developing the ability to apply acquired knowledge in solving real-world problems .
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	Software Engineering
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. Organizational structure, role allocation, task undertaking, communication ability and interface definition of software team collaboration and development.</li> <li>2. Prepare software project plan and track and control the process of software project plan, including: cost plan, schedule plan and other plans, as well as how to track and control the project during the implementation process.</li> </ol> </li> <li>● <b>Skill:</b> <p>The process of preparing and tracking and controlling software project plans, including: scope plan, quality plan, configuration plan, personnel and communication plan, risk plan, contract plan and other plans, as well as how to track and control the project during the implementation of the project.</p> </li> <li>● <b>Competence:</b> <ol style="list-style-type: none"> <li>1. Software development team management ability, project manager role, and familiarity with the responsibilities of each role. Through the understanding and implementation of teamwork, students will experience the importance of organizational teams to the country and individuals, and inspire their patriotic feelings and mission responsibility for serving the country through science and technology.</li> <li>2. Ability to continue learning and team work spirit.</li> <li>3. Have the spirit of craftsmanship, social responsibility and good professional ethics.</li> </ol> </li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Overview of Project Management</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Software house</li> </ul>

	<ul style="list-style-type: none"> <li>● Software project</li> <li>● Project management</li> <li>● A new form of the Internet era</li> </ul> <p><b>Chapter 2 Establishment of software Project</b> (1 contact hour; 1 self-study hour)</p> <ul style="list-style-type: none"> <li>● Feasibility analysis</li> <li>● Approve and initiate a project</li> <li>● Project bidding</li> <li>● The project contract is signed</li> <li>● Project authorization</li> <li>● Project tax system</li> <li>● Intellectual property right</li> </ul> <p><b>Chapter 3 Task Breakdown of Software Projects</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Task decomposition definition</li> <li>● Task decomposition process</li> <li>● Task decomposition method</li> <li>● Task breakdown results</li> </ul> <p><b>Chapter 4 Project Cost Estimation</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Cost estimate overview</li> <li>● Cost estimation process</li> <li>● Cost estimation method</li> <li>● Cost budgeting</li> </ul> <p><b>Chapter 5 Project Schedule Estimation</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Progress estimate overview</li> <li>● Relationships between tasks</li> <li>● Schematic diagram of schedule</li> <li>● Task resource estimation</li> <li>● Task time estimate</li> <li>● Progress plan arrangement method</li> </ul> <p><b>Chapter 6 Project personnel selection</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Team member type</li> <li>● Project stakeholder identification</li> <li>● Project communication skills</li> </ul> <p><b>Chapter 7 Software Version Management</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Overview of version management</li> <li>● Versioning process</li> <li>● Version management plan</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>● Versioning tools</li> </ul> <p><b>Chapter 8 Project Quality Assurance</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Software quality definition</li> <li>● Software quality model</li> <li>● Software Quality Assurance</li> </ul> <p><b>Chapter 9 Project Risk Control</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Risk identification</li> <li>● Risk analysis</li> <li>● Risk planning</li> <li>● Risk monitoring</li> <li>● Common risks and their treatment</li> </ul> <p><b>Chapter 10 Software Survival Period Execution Monitoring</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Project survival period execution control</li> <li>● Progress cost execution control</li> <li>● Quality plan implementation control</li> </ul> <p><b>Chapter 11 Knowledge Management in Software Enterprises</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Overview of knowledge management</li> <li>● Knowledge management theory</li> <li>● Software enterprise knowledge management model</li> <li>● Knowledge management influencing factors in software enterprises</li> <li>● Knowledge management capability of software enterprises</li> <li>● Software enterprise knowledge management program</li> </ul> <p><b>Chapter 12 Internet Enterprise Financing</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Overview of Internet enterprises</li> <li>● Overview of Internet enterprise financing model</li> <li>● Comparison between Internet enterprises and traditional enterprises financing mode</li> <li>● Chinese Internet enterprises innovate financing mode</li> <li>● Policy suggestions on financing mode selection of Internet enterprises in China</li> </ul> <p><b>Chapter 13 Start-up Internet enterprise value</b></p>
--	---



	<p><b>assessment</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Internet enterprise valuation method</li> <li>● Start-up Internet enterprise valuation model</li> </ul> <p><b>Chapter 14 Internet enterprise mergers and acquisitions</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Overview of Internet enterprise mergers and acquisitions</li> <li>● Mergers and acquisitions business model and innovation development status</li> <li>● Mergers and acquisitions analysis and its latest development trends</li> <li>● Policy and its development considerations</li> </ul> <p><b>Chapter 15 Internet enterprise operation efficiency</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● The connotation of operation efficiency evaluation</li> <li>● Analysis of operational efficiency</li> <li>● Classification and comparison of operational efficiency</li> <li>● Operational decision recommendations</li> </ul> <p><b>Chapter 16 Market dominance of Internet enterprises</b> (2 contact hour; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Abuse of the concept of dominant market position</li> <li>● The legal basis for abuse of dominant market position</li> <li>● Research on regulations of abusive price behavior</li> <li>● Research on the regulation of abuse of non-price behavior</li> <li>● Suggestions on improving the regulations on abuse of dominant market position</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. Installation and familiarity with Project, a progress management tool. (2 contact hours, 2 self-study hours)</li> <li>2. Use of the Project (4 hours, 2 self-study hours)</li> <li>3. Installation and familiarity with version control software SVN (2 contact hours, 2 self-study hours)</li> <li>4. Use of SVN software (4 contact hours, 4 self-study hours)</li> <li>5. Discussion of management mode 1 (2 contact hours, 2 self-study hours)</li> </ol>
--	--

	6.Discussion of management mode 2 (2 contact hours, 2 self-study hours)
Examination forms	Open-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 (10%), experiments (20%) and project performance (10%).</p> <p>The final assessment (open book written test) accounts for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Fang Muyun. Software Project Management and Case - Micro-lecture video edition [M]. Beijing: Tsinghua University Press, 2023.</p> <p><b>2. Reference books</b></p> <p>[1] Han Wanjiang, Jiang Lixin. Software Project Management Case Tutorial [M]. 4th edition. Beijing: China Machine Press, 2018.</p> <p>[2] Xue Sixin, Jia Guojun. Software Project Management [M]. 2nd Edition. Beijing: China Machine Press, 2010.</p> <p>[3] Wu Weihong, Engineering Project Management Theory and Practice [M]. Beijing: China Machine Press, 2016.</p>
Data of last amendment	June 10, 2025

## Human-Computer Interaction Technology

Module designation	Human-Computer Interaction Technology
Semester(s) in which the module is taught	6 <sup>th</sup> semester
Person responsible for the module	Lecturer Tang Ziwei
Language	Chinese
Relation to curriculum	Human-computer interaction (HCI) is a discipline that explores how to facilitate efficient, natural, and comfortable collaboration between humans and computers through information exchange. This course equips students with essential knowledge in HCI-related fields, including user experience design, interdisciplinary studies, interface design, conceptual interaction, socialized interaction, emotional interaction, requirement discovery, interaction design processes, design guidelines and principles, prototype development and agile methodologies, direct manipulation and interface design, command systems, menus and forms, user documentation and online help, as well as quality assessment of HCI implementations. The curriculum prepares students for subsequent engineering applications and scientific research in this field laying foundations .
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,	Total workload = 90 hours Contact hours = 48 hours

self-study hours)	Self-study hours = 42 hours
Credit points	3
Required and recommended prerequisites for joining the module	Python Programming, C Programming, Software Engineering, Data Structures
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. Understand the core theoretical framework of human-computer interaction, including user-centered design, principles of cognitive psychology, and basic principles of interaction design.</li> <li>2. Master the interaction design process, proficient in using design tools to complete low-fidelity/high-fidelity prototype design, and master user testing methods for iterative optimization.</li> <li>3. Be familiar with frontier technologies such as multimodal interaction, virtual reality/augmented reality interaction, affective computing, and understand their typical application scenarios in medical, education, industry and other fields.</li> </ol> </li> <li>● <b>Skill:</b> <ol style="list-style-type: none"> <li>1. Be able to accurately refine user requirements through user research, competitive product analysis and other methods, and design interaction processes and interface layouts based on requirements.</li> <li>2. Proficient in using design tools to quickly make interactive prototypes, verify the effectiveness of the design through user testing, and iteratively optimize the scheme.</li> <li>3. Master the principles of multimodal interaction design, such as voice, gesture, and haptics, and be able to optimize the interaction experience for different devices.</li> </ol> </li> <li>● <b>Competence:</b> <ol style="list-style-type: none"> <li>1. Be able to explore real needs through scientific methods from the perspective of users, and design interaction schemes that meet users' cognitive and behavioral habits.</li> <li>2. Complete the design, deployment, debugging and other tasks of human-computer interaction system in a</li> </ol> </li> </ul>

	<p>team.</p> <p>3. Play individual, member and leadership roles in a team, communicate effectively with team members, and be able to complete work independently or collaboratively.</p>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Human-Computer Interaction and User Experience</b> (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Human-computer interface and human-computer interaction</li> <li>● Development of human-computer interface</li> <li>● Interaction Design</li> <li>● Optimal User Experience</li> <li>● Human-Computer Interaction and Software Engineering</li> </ul> <p><b>Chapter 2 Related subjects in Human-computer Interaction</b> (3 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Fields related to human-computer interaction</li> <li>● Cognitive psychology</li> <li>● Cognitive theory</li> <li>● Ergonomics</li> </ul> <p><b>Chapter 3 Human-Computer Interaction Interfaces</b> (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Classification of interfaces</li> <li>● Practical interfaces</li> <li>● Input/Output interface</li> <li>● Functional Interface</li> <li>● Platform interface</li> </ul> <p><b>Chapter 4 Conceptualizing Interactions</b> (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Conceptualize interactions</li> <li>● Conceptual models</li> <li>● Interaction Types</li> <li>● Guide design and research</li> </ul> <p><b>Chapter 5 Social Interactions</b> (3 contact hours; 3 self-study hours)</p> <p>Socializing</p> <ul style="list-style-type: none"> <li>● Face-to-face conversation</li> </ul>

	<ul style="list-style-type: none"> <li>● Remote conversations</li> <li>● Co-occurrence</li> <li>● Social Participation</li> </ul> <p><b>Chapter 6 Emotional Interaction</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Emotions and User Experience</li> <li>● Emotional design</li> <li>● 6.Emotional ICONS</li> <li>● 6.Annoying interface</li> <li>● Affective Computing techniques</li> <li>● Persuasion Techniques and Behavior Change</li> <li>● Anthropomorphism</li> </ul> <p><b>Chapter 7 Interaction Design Processes</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● The style of the design</li> <li>● Development methodology</li> <li>● Observing Users</li> <li>● Activities in Interaction Design</li> </ul> <p><b>Chapter 8 Design Guidelines and Principles</b> (3 hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● The 4 Pillars of Interface Design</li> <li>● Participatory Design</li> <li>● Design Guidelines</li> <li>● Design Principles</li> </ul> <p><b>Chapter 9 Direct Manipulation and Interface Design</b> (3 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Direct manipulation and WIMP interface</li> <li>● Applications for direct manipulation</li> <li>● Design for direct manipulation</li> <li>● 3D Interface</li> <li>● Teleoperation</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <p>Experiment 1: Kinect recognition of human motion (2 contact hours; 2 self-study hours)</p> <p>Experiment 2: Leapmotion monitors moving objects in the scene using hands (2 contact hours; 2 self-study hours)</p> <p>Experiment 3: Making Chinchillas using 3Dmax (4 contact hours; 2 self-study hours)</p> <p>Experiment 4: Implement Tank Battle with Unity (4</p>
--	---

	<p>contact hours; 4 self-study hours)</p> <p>Lab 5: Creating simple AR Instances using vuforia (4 contact hours; 4 self-study hours)</p>
Examination forms	Project debrief
Study and examination requirements	<p>The homework should be completed by students in each class independently.</p> <p>Late arrivals, early departures, or unapproved absences are not allowed.</p> <p>4. The usual grade counts for 40%, including class performance (10%), homework assignments (10%) and computer LABS (20%).</p> <p>The final assessment (project report) will account for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Yong Lu, Qian Li, Na Sun, Xiuqin Pan. Human-computer Interaction Technology [M]. Beijing: Tsinghua University Press, 2024.</p> <p><b>2. Reference books</b></p> <p>[1] Yu Qiang, Zhou Su. Human-computer Interaction Technology [M]. 2nd Edition. Beijing: Tsinghua University Press, 2022.</p> <p>[2] Wu Yadong, Zhang Xiaorong, Chen Huarong. Human-computer Interaction Technology and Application [M]. Beijing: China Machine Press, 2020.</p>
Data of last amendment	June 29, 2025

## Machine Learning

Module designation	Machine Learning
Semester(s) in which the module is taught	6 <sup>th</sup> semester
Person responsible for the module	Teaching assistant Cai Chengfei
Language	Chinese
Relation to curriculum	Machine Learning is a core course in Computer Science and Technology, combining theoretical rigor with practical application. This essential program equips students with fundamental machine learning methodologies and principles, while exposing them to industry-leading approaches. The curriculum builds both theoretical foundations and hands-on skills for future research careers. It cultivates problem-solving capabilities and practical expertise, enabling students to design effective machine learning models tailored to specific tasks. By integrating theory with real-world applications, the course emphasizes the balance between academic understanding and real-world implementation. Study and computer practice to improve students' ability to analyze and solve problems.
Teaching methods	Target students: Computer Science and Technology major students Type of teaching: Theoretical teaching and experimental 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,	Total workload = 90 hours Contact hours = 48 hours



self-study hours)	Self-study hours = 42 hours
Credit points	3
Required and recommended prerequisites for joining the module	Python programming, data structure, algorithm analysis and design, data mining, etc
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 theory of machine learning and classical machine learning algorithms.</li> <li>2. Master mainstream machine learning development tools and architectures, and be able to design, program, implement and verify machine learning algorithms by themselves according to specific problems.</li> <li>3. Understand the cutting-edge technology of machine learning, understand the latest cutting-edge technology with the knowledge learned, and skillfully build the running environment of machine learning.</li> </ol> </li> <li>• <b>Skill:</b> <ol style="list-style-type: none"> <li>1. Master the classical machine learning algorithm and be able to design the machine learning algorithm independently and write the code to implement it.</li> <li>2. Master the software and algorithm development environment for machine learning development.</li> <li>3. Understand the development of machine learning and future research directions.</li> </ol> </li> <li>• <b>Competence:</b> <ol style="list-style-type: none"> <li>1. Ability to use machine learning third-party libraries and development environment.</li> <li>2. Be able to use machine learning algorithms to solve problems..</li> </ol> </li> </ul>

Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Overview of machine learning</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● The definition of artificial intelligence</li> <li>● Three waves in the history of artificial intelligence</li> <li>● The field of artificial intelligence research.</li> <li>● The main work of machine learning</li> <li>● Machine learning development environment</li> <li>● Use Anaconda to write and execute Python source files</li> </ul> <p><b>Chapter 2 Python Data Processing Basics</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic methods of Python programming</li> <li>● Python file read/write methods</li> <li>● Use NumPy to retrieve the contents of the data file</li> <li>● Use Pandas to access data files</li> </ul> <p><b>Chapter 3 Commonly used machine learning libraries in Python</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Common invocation methods of third-party libraries</li> <li>● How to use the machine learning library</li> <li>● Use of NumPy, Pandas and Matplotlib libraries</li> <li>● OpenCV, Scikit-Learn, Wordcloud, Jieba, PLI library are used</li> </ul> <p><b>Chapter 4 Fundamentals of Machine Learning</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Fundamentals of machine learning</li> <li>● Common machine learning algorithms</li> <li>● Classification of machine learning algorithms</li> <li>● Machine learning collects data, data standards, model selection, and model training</li> <li>● Performance evaluation of machine learning algorithms</li> <li>● Implement machine learning using Sklearn</li> </ul> <p><b>Chapter 5 KNN classification algorithm</b></p>
---------	--

	<p>(4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of KNN classification algorithm</li> <li>● Core elements of KNN algorithm</li> <li>● Distance measurement method</li> <li>● Use KNN to solve the iris classification problem</li> <li>● Use KNN to realize handwritten digit recognition</li> </ul> <p><b>Chapter 6 K-Means clustering algorithm</b></p> <p>(2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of clustering algorithm</li> <li>● The principle of K-Means clustering algorithm</li> <li>● K-Means algorithms</li> <li>● Use K-Means algorithm to solve practical clustering problems</li> <li>● Characteristics and defects of K-means algorithm</li> </ul> <p><b>Chapter 7 Recommended Algorithms</b></p> <p>(4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of recommendation algorithms</li> <li>● Evaluation perspective of recommendation algorithm</li> <li>● The challenge of recommendation algorithms</li> <li>● The category of recommendation algorithms</li> <li>● Recommendation algorithm based on collaborative filtering</li> <li>● Other commonly recommended algorithms</li> </ul> <p><b>Chapter 8 Regression algorithms</b></p> <p>(4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of regression algorithm</li> <li>● Use of single linear regression algorithm</li> <li>● Multiple linear regression algorithm</li> <li>● The basic concept of logical regression</li> <li>● The algorithmic principle of logistic regression</li> <li>● Use logistic regression algorithm to solve the problem</li> </ul> <p><b>Chapter 9 Support Vector Machine SVM</b></p> <p>(4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of support vector machine</li> <li>● Implementation of support vector machine</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>● Use decision perceptron for credit classification</li> <li>● Support vector machine SVM parameter function</li> <li>● Optimization of support vector machine parameters</li> <li>● Implementation of support vector machine</li> <li>● Function and use of support vector machine kernel function</li> </ul> <p><b>Chapter 10 Neural networks</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● The basic principle of neural networks</li> <li>● The architecture of a multi-layer neural network</li> <li>● Activation function and function</li> <li>● A method for building neural networks using Python</li> </ul> <p><b>Chapter 11 Deep Learning</b> (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic concepts of deep learning</li> <li>● The principle of convolutional neural network CNN</li> <li>● Structure of recurrent neural network RNN</li> <li>● Basic application of the TensorFlow framework: Cat and dog classification</li> <li>● Simple use of Keras framework: face recognition</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 14 self-study hours)</p> <p>Experiment 1: Data processing and use of third-party libraries (4 contact hours; 2 self-study hours)</p> <p>Experiment 2: KNN, K-means and recommendation algorithm implementation (4 contact hours; 4 self-study hours)</p> <p>Experiment 3: Implementation of regression algorithm, SVM and neural network (4 contact hours; 4 self-study hours)</p> <p>Experiment 4: Deep Learning Practice (4 contact hours; 4 self-study hours)</p>
Examination forms	Final paper

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 performance accounts for 50%, including classroom performance (10%), in-class test (10%), technical sharing (10%) and computer experiment (20%).</p> <p>The final assessment (essay) accounts for 50%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Liu Yan. Python Machine Learning [M]. Beijing: Tsinghua University Press, 2021.</p> <p><b>2. Reference books</b></p> <p>[1] Li Hang. Statistical Learning Methods [M]. 2nd Edition. Beijing: Tsinghua University Press, 2019.</p> <p>[2] Magnus Lie Hetland. Python Basic Tutorial [M]. translated by Yuan Guozhong. 3rd edition. Beijing: People's Posts and Telecommunications Press, 2018.</p> <p>[3] Zhou Zhihua. Machine Learning [M], Beijing: Tsinghua University Press, 2016.</p>
Data of last amendment	June 9, 2025

## Digital Image Processing

Module designation	Digital Image Processing
Semester(s) in which the module is taught	7th Semester
Person responsible for the module	Lecturer Tang Ziwei
Language	Chinese
Relation to curriculum	Digital image processing is the basis of machine vision, pattern recognition, medical image processing, etc. This course provides the basic knowledge of digital image processing for engineering students. It is a comprehensive course with strong theoretical and practical. The course covers the basic principles of digital image processing, including image sampling and quantization, image arithmetic and logical operations, histogram, image color space, image segmentation, image morphology, image frequency domain processing, image segmentation, image noise reduction and image restoration, feature extraction and recognition, etc., which lays a foundation for subsequent engineering applications and scientific research.
Teaching methods	Target students: Computer Science and Technology students Type of teaching: theoretical teaching, experimental 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,	Total workload = 90 hours Contact hours = 48 hours

self-study hours)	Self-study hours = 42 hours
Credit points	3
Required and recommended prerequisites for joining the module	Data Structures, Design and Analysis of Algorithms, etc
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. Understand the mathematical representation of digital images, and master the geometric transformation and pixel-level operation of images.</li> <li>2. Master spatial and frequency domain image enhancement methods (such as filtering, sharpening), and image restoration techniques (such as denoising, deblurring).</li> <li>3. Master image segmentation algorithms (such as threshold segmentation, edge detection, region growing) and feature extraction methods (such as shape, texture, color features) to lay a foundation for subsequent image analysis.</li> </ol> </li> <li>• <b>Skill:</b> <ol style="list-style-type: none"> <li>1. Select and implement appropriate algorithms according to image processing requirements, and improve algorithm performance through parameter tuning and code optimization.</li> <li>2. Evaluate the image processing effect through objective indicators (such as PSNR, SSIM) and subjective evaluation, and propose targeted improvement plans.</li> <li>3. Design a complete image processing flow (such as preprocessing, feature extraction, analysis and decision), and integrate the algorithm into the practical application system.</li> </ol> </li> </ul>

	<p>• <b>Competence:</b></p> <ol style="list-style-type: none"> <li>1. Master the methods and general procedures of digital image experiments, and be able to carry out experimental operations safely and correctly.</li> <li>2. Complete the design, deployment, debugging and other tasks of digital image processing programs in a team.</li> <li>3. Work in individual, member and leadership roles in a team, communicate effectively with team members and be able to work independently or collaboratively.</li> </ol>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 32 self-study hours)</p> <p><b>Chapter 1 Introduction</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● The components of a digital image processing system</li> <li>● Electromagnetic spectrum and visible light</li> <li>● Electromagnetic wave sensors</li> <li>● Visual system characteristics</li> <li>● Digital image processing applications</li> </ul> <p><b>Chapter 2 Digital Image Basic Concepts and Operations</b> (3 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Image imaging model</li> <li>● Basic concept of image</li> <li>● Basic relations between pixels</li> <li>● Basic image operations</li> </ul> <p><b>Chapter 3 Image Gray-level Transformation and Spatial Filtering</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Common grayscale transformation</li> <li>● Histogram based gray scale transformation</li> <li>● Filtering in spatial domain</li> </ul> <p><b>Chapter 4 Processing Images in the frequency Domain</b> (4 contact hours; 4 self-study hours)</p>



	<ul style="list-style-type: none"> <li>● Two-dimensional Discrete Fourier Transform</li> <li>● Frequency domain filtering basics</li> <li>● Low-pass Filtering in frequency domain</li> <li>● High-pass filtering in the frequency domain</li> <li>● Homomorphic filtering</li> <li>● Frequency selective filter</li> </ul> <p><b>Chapter 5 Image Restoration</b> (3 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Image degradation models and restoration models</li> <li>● Noise Models</li> <li>● Restore degraded images caused only by noise</li> <li>● Estimation of the degradation function</li> <li>● Inverse filtering</li> <li>● Wiener filtering and constrained least squares filtering</li> <li>● Lucy-Richardson Restoration (non-blind L-R restoration)</li> <li>● Blind image restoration</li> </ul> <p><b>Chapter 6 Processing Color Images</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Color Image basics</li> <li>● Color Spaces</li> <li>● Pseudo-color image processing</li> <li>● Color Transformations</li> <li>● Spatial filtering for color images</li> <li>● Color image edge detection</li> </ul> <p><b>Chapter 7 Wavelets and Multiresolution Processing</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Wavelet transform basics</li> <li>● Image Wavelet Transform</li> <li>● Wavelet image denoising</li> <li>● Wavelet image fusion</li> </ul> <p><b>Chapter 8 Image Compression</b> (3 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Image Compression basics</li> <li>● Common Encodings</li> </ul>
--	---

	<ul style="list-style-type: none"> <li>● Bit plane encoding</li> <li>● Transform Coding</li> <li>● Video Compression</li> </ul> <p><b>Chapter 9 Morphological Processing</b> (3 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> <li>● Corrosion and expansion</li> <li>● Opening and closing operations</li> <li>● Hit and Miss transformations</li> <li>● Some Basic morphological algorithms</li> <li>● Morphological Processing of Grayscale Images</li> </ul> <p><b>Part B. Experiment teaching</b> (16 contact hours; 10 self-study hours)</p> <ul style="list-style-type: none"> <li>● Experiment 1: Image Denoising and Image Enhancement (4 contact hours; 2 self-study hours)</li> <li>● Experiment 2: Automatic detection of ruler in images (4 contact hours; 2 self-study hours)</li> <li>● Experiment 3: Deformable matching of two images with partially identical scenes (4 contact hours; 2 self-study hours)</li> <li>● Experiment 4: Moving target tracking (4 contact hours; 2 self-study hours)</li> </ul>
Examination forms	Project debrief
Study and examination requirements	<p>Homework should be done independently by students after each class.</p> <p>Late arrivals, early departures, or unapproved absences are not allowed.</p> <p>4. The usual grade counts for 40%, including class performance (10%), homework assignments (10%) and computer LABS (20%).</p> <p>The final assessment (project report) will account for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Huang Jin, Li Jianbo. Digital Image Processing: Principle and Implementation [M]. Beijing: Tsinghua University Press, 2020. (in Chinese)</p>

	<p><b>2. Reference books</b></p> <p>[1] Zhang Hong. Digital Image Processing and Analysis [M]. 4th Edition. China Machine Press, 2024.</p> <p>[2] Xing Li, Bian Xuefen, Wang Peng. Intelligent Digital Image Processing: Principle and Technology [M]. Beijing: China Machine Press, 2023. (in Chinese)</p>
Data of last amendment	June 29, 2025

## Data Visualization Techniques

Module designation	Data Visualization Techniques
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	"Data Visualization Technology" is an elective course for Computer Science and Technology majors. As a vital tool for unlocking data's value, data visualization transforms abstract numerical data into visual representations like charts and graphs, enabling rapid interpretation of underlying patterns or trends. Through this course, students will: understand core concepts of data visualization; grasp fundamental principles of visual perception and cognition; master foundational theories of visualization techniques; acquire methods for presenting diverse data types; and become proficient in standard workflows and tool usage. The program aims to equip students with technical expertise in data visualization, empowering them to independently complete data-driven projects Visualize the processing.
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, self-study hours)	<p>Total workload = 90 hours</p> <p>Contact hours = 48 hours</p> <p>Self-study hours = 42 hours</p>

Credit points	2
Required and recommended prerequisites for joining the module	Python programming, Javascript, database principles and applications
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> <li>● <b>Knowledge:</b> To understand the basis, function, development history, future challenges and development direction of data visualization; to be able to effectively express the solutions to complex data visualization problems in big data applications based on big data professional knowledge.</li> <li>● <b>Skill:</b> <ol style="list-style-type: none"> <li>1. Master data visualization tools, Excel, Echarts, Tableau and other common tools.</li> <li>2. Master the libraries related to python data visualization, Numpy, Matplotlib, Pandas, Seaborn, Bokeh, pyqtgraph.</li> </ol> </li> <li>● <b>Competence:</b> Master the principles of data visualization, data visualization process, data visualization design principles and techniques, master text visualization, network visualization, spatial information visualization and other methods of data visualization, be able to select technical routes according to the requirements of engineering project tasks, and design big data application solutions that meet the needs.</li> </ul>
Content	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Overview of Big Data Visualization</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Basic big data visualization</li> <li>● The goal and function of big data visualization</li> <li>● Features and applications of data visualization</li> </ul> <p><b>Chapter 2 Principles of Big Data Visualization</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Light and visual characteristics</li> <li>● Color</li> <li>● Visual channels</li> </ul>

	<ul style="list-style-type: none"> <li>● Data visualization process, design principles and techniques</li> </ul> <p><b>Chapter 3 Big Data Visualization Methods</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Introduction to visual diagrams</li> <li>● Text visualization</li> <li>● Network visualization</li> <li>● Spatial information visualization</li> </ul> <p><b>Chapter 4 Data visualization tools</b> (2 contact hours; 1 self-study hours)</p> <ul style="list-style-type: none"> <li>● Excel</li> <li>● Echart</li> <li>● Tableau</li> <li>● The magic mirror</li> <li>● D3.js</li> <li>● Visual development languages</li> </ul> <p><b>Chapter 5 Excel Data Visualization</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Excel Functions and graphs</li> <li>● Excel Data source</li> <li>● Excel Visualization applications</li> </ul> <p><b>Chapter 6 Tableau Data Visualization</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Tableau and the Tableau interface</li> <li>● Use Tableau for visualization</li> <li>● Tableau data analysis</li> </ul> <p><b>Chapter 7 Echarts and pyecharts data visualization</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Download and use of ECharts</li> <li>● ECharts visualization application</li> <li>● Pyecharts visualization application</li> </ul> <p><b>Chapter 8 Python Data Visualization</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Python visualization library</li> <li>● The NumPy library</li> <li>● Data visualization based on matplotlib</li> <li>● Data visualization based on Pandas</li> <li>● Data visualization based on Seaborn</li> <li>● Data visualization based on Bokeh</li> <li>● Data visualization based on pyqtgraph</li> </ul> <p><b>Chapter 9 Comprehensive Training of Data Visualization</b></p>
--	---

	<p>(6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Python vertical bar chart training</li> <li>● Python horizontal bar chart training</li> <li>● Python multi-data parallel bar chart training</li> <li>● Python line chart training</li> <li>● Python scatter plot training</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. Draw the Olympic rings logo (2 contact hours; 1 self-study hours)</li> <li>2. Drawing charts in Jupyter Notebook (2 contact hours; 1 self-study hours)</li> <li>3. Use Matplotlib to draw a bar chart and line chart of the sequential growth of sales volume of a single product (2 contact hours; self-study hours)</li> <li>4. Use Pandas to draw line chart and column chart to analyze sales revenue data (2 contact hours; self-study hours)</li> <li>5. Use the barplot function to draw a bar chart and analyze and compare the math scores of each student using the provided data set (2 contact hours; 2 self-study hours)</li> <li>6. Pyecharts draw a mixture of pie charts and ring charts to show the sales volume of a product in three cities. (2 contact hours; 2 self-study hours)</li> <li>7. Use of Pyqtgraph library (2 contact hours; self-study hours)</li> <li>8. Implement interactive charts by Matplotlib+Pandas+Pyqtgraph (2 contact hours; 2 self-study hours)</li> </ol>
Examination forms	Practical exams
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>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] Huang Yuan. Big Data Visualization Technology and Application [M]. Shanghai: Huada University Press, 2020.</p>

	<b>2. Reference books</b> [1] Chen Wei, Zhang Song, Lu Aidong. Basic Principles and Methods of Data Visualization [M].Beijing: Science Press, 2018.
Data of last amendment	June 10, 2025



## Robot Technology and Applications

Module designation	Robot Technology and Applications
Semester(s) in which the module is taught	7 <sup>th</sup> semester
Person responsible for the module	Wang Yang
Language	Chinese
Relation to curriculum	The course of "Robot Technology and Applications" is an elective course of the programme of Computer Science and Technology and is an important component of the computer application technology. The teaching goal of "Robot Technology and Applications" is to enable students to master the basic concepts, technical principles, applications, and cutting-edge development directions of industrial robots, and to enable students master the overall design of industrial robots and the design of each core component, and understand the innovative design methods. Lay stress on the training of students' ability to comprehensively apply the basic theories and professional knowledge to analyze and solve problems.
Teaching methods	Target students: students of Computer Science and Technology Type of teaching: theoretical teaching, experiment teaching Contact hour: 48 hours Including: Theoretical teaching: 40 hours Experiment teaching: 8 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	3.0

Required and recommended prerequisites for joining the module	Linear Algebra A, Discrete Mathematics, Digital Logic Circuits
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. The definition, characteristics, composition, technical parameters, classification, application fields and development status and other related knowledge of industrial robots.</li> <li>2. The overall design of industrial robots and the design of each core component.</li> <li>3. Kinematic analysis, static calculation and dynamic analysis of industrial robots.</li> <li>4. The sensing system of industrial robots, the basic principles of sensors and the fundamentals of information processing.</li> <li>5. The control methods and Trajectory planning of industrial robots.</li> </ol> </li> <li>● <b>Skill:</b> <ol style="list-style-type: none"> <li>1. Be able to understand the technology characteristics and application fields of industrial robots.</li> <li>2. Be able to master the overall design of industrial robots and the design of each core component, and understand the basic innovative design methods.</li> <li>3. Be able to master the control methods of industrial robots, including position control and force control.</li> <li>3. Be able to master the knowledge of application cases of industrial robots through experiments.</li> </ol> </li> <li>● <b>Competence:</b> <ol style="list-style-type: none"> <li>1. Master the mathematical knowledge required for designing, analyzing and controlling robot systems, able to establish and analyze kinematic and dynamic equation models of robots, and master the commonly used sensors, control system structures and motion trajectory planning methods for robots.</li> <li>2. Apply the scientific principles and mathematical model knowledge of robots to model practical problems, conduct comprehensive analysis, and propose reasonable solutions.</li> <li>3. Master the cutting-edge knowledge and skills in the</li> </ol> </li> </ul>

	field of robot technology, and be able to recognize the diversity of robot solutions for different application scenarios and being able to select reasonable solutions.
Content	<p><b>Part A. Theoretical teaching</b> (40 contact hours; 36 self-study hours)</p> <p><b>Chapter 1 Introduction</b> (2 contact hours; 2 self-study hour)</p> <ul style="list-style-type: none"> <li>● The definition, characteristics, and composition of Industrial robots.</li> <li>● The technical parameters, classification, and application fields of Industrial robots.</li> <li>● The development status and research contents of Industrial robots.</li> </ul> <p><b>Chapter 2 Kinematic Analysis of Industrial Robots</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Homogeneous coordinates and the description of the object.</li> <li>● Homogeneous transformation and operation.</li> <li>● The coordinate system of the industrial robot connecting rod and its homogeneous transformation matrix.</li> <li>● Kinematic equations of Industrial robots and their solutions.</li> </ul> <p><b>Chapter 3 Static Calculation and Dynamic Analysis of Industrial Robots</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Speed Jacobi and speed analysis of industrial robots.</li> <li>● Force Jacobi and static calculation of industrial robots.</li> <li>● Dynamics analysis of industrial robots.</li> </ul> <p><b>Chapter 4 Mechanical System Design of Industrial Robots</b> (6 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Overall design of industrial robots.</li> <li>● Design of transmission components.</li> <li>● Arm design.</li> <li>● Wrist design.</li> <li>● Hand design.</li> <li>● Design of the fuselage and walking mechanism.</li> </ul> <p><b>Chapter 5 Feelings of Industrial Robots</b> (6 contact hours; 4 self-study hours)</p>

	<ul style="list-style-type: none"> <li>● The sensory systems of industrial robots.</li> <li>● Joint sensors.</li> <li>● Hand sensors.</li> <li>● Environmental recognition sensors.</li> <li>● Information processing of sensors.</li> </ul> <p><b>Chapter 6 Control of Industrial Robots</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Characteristics and classification of industrial robot control.</li> <li>● Position control of industrial robots.</li> <li>● Force control of industrial robots.</li> </ul> <p><b>Chapter 7 Trajectory Planning of Industrial Robots</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● General issues of trajectory planning.</li> <li>● Trajectory planning of joint space.</li> <li>● Trajectory planning in rectangular coordinate space.</li> <li>● Real-time generation of trajectories.</li> </ul> <p><b>Chapter 8 Application Examples of Industrial Robots</b> (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> <li>● Wheeled mobile robots and AGV.</li> <li>● Parallel robots and virtual-axis machine tools.</li> <li>● Spherical SCARA robot.</li> </ul> <p><b>Part B. Experiment teaching</b> (8 contact hours; 6 self-study hours)</p> <p>In order to help students better understand the concept, principle, and applications of industrial robots, master the processes of simulation of Kinematics of the robotic arm by Matlab, and improve their practical abilities to work in teams and analyze and simulate practical problems using software tools, the following two typical experimental classes will be arranged:</p> <ol style="list-style-type: none"> <li>1. Research and report on the frontiers of robotics. (4 contact hours; 3 self-study hours)</li> <li>2. Kinematic simulation of the robotic arm based on Matlab. (4 contact hours; 3 self-study hours)</li> </ol>
Examination forms	Closed-book written exam
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 accounts for 40%, including</p>

	<p>assignments (20%) and experiment (20%).</p> <p>Final assessment (closed-book written exam) accounts for 60%.</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Xiang Sitong, Cui Yuguo, Yu Siyi. Fundamentals and Applications of Robot Technology [M]. Beijing: China Machine Press, 2024.</p> <p><b>2. Reference books</b></p> <p>[1] Zhu Shiqiang, Wang Xuanyin. Robot Technology and Applications [M]. Second Edition. Hangzhou: Zhejiang University Press, 2019.</p> <p>[2] Zhao Jianwei. Robot System Design and Application Technology [M]. Second Edition. Beijing: Tsinghua University Press, 2024.</p>
Data of last amendment	June 2025

## English for Computer Science

Module designation	English for Computer Science
Semester(s) in which the module is taught	2nd semester
Person responsible for the module	Teaching assistant Cai Chengfei
Language	English and Chinese
Relation to curriculum	This course is a professional language foundation course for undergraduate students majoring in Computer Science and Technology. It is designed to improve students' ability to read, understand, and apply professional English materials in the field of computer science. The course emphasizes vocabulary building, text comprehension, and translation skills. It also helps lay a solid foundation for later coursework, technical documentation writing, and international communication.
Teaching methods	Target students: Undergraduate students majoring in Computer Science and related disciplines Type of teaching: Theoretical teaching with integrated practice Contact hours: 32 Including: Theoretical teaching: 32 hours Practical computer lab: 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	Required: College English (Level 1) or equivalent
Module	Learning outcomes: ● <b>Knowledge:</b>

<p>objectives/intended learning outcomes</p>	<ol style="list-style-type: none"> <li>1. Master technical terms and common sentence structures in computer English.</li> <li>2. Understand the typical structure and expressions used in computer-related texts.</li> </ol> <ul style="list-style-type: none"> <li>● <b>Skill:</b> <ol style="list-style-type: none"> <li>1. Improve reading comprehension of English texts in the field of computing.</li> <li>2. Translate basic computer science texts between English and Chinese.</li> <li>3. Accurately use professional terminology in reading and writing tasks.</li> </ol> </li> <li>● <b>Competence:</b> <ol style="list-style-type: none"> <li>1. Develop the ability to independently read professional literature in English.</li> <li>2. Build cross-cultural awareness and international communication readiness.</li> <li>3. Apply English reading skills to support further study and research in the discipline.</li> </ol> </li> </ul>
<p>Content</p>	<p><b>Part A. Theoretical teaching</b> (32 contact hours; 28 self-study hours)</p> <p><b>Chapter 1 Overview of Computer English</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Characteristics of English for Specific Purposes (ESP)</li> <li>● Basic sentence structure and logic in technical texts</li> <li>● Reading strategies for technical articles</li> </ul> <p><b>Chapter 2 Computer Systems and Hardware</b> (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> <li>● Computer architecture terminology</li> <li>● Vocabulary related to input/output devices, storage, CPU, etc.</li> <li>● Parsing descriptions of system configurations</li> </ul> <p><b>Chapter 3 Software and Operating Systems</b> (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> <li>● Common terms in software systems</li> <li>● Understanding user manuals and system instructions</li> <li>● Reading system interface descriptions</li> </ul> <p><b>Chapter 4 Programming Languages and Algorithms</b> (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> <li>● Vocabulary in programming environments</li> <li>● Descriptions of procedures, logic, and flowcharts</li> </ul>

	<ul style="list-style-type: none"> <li>● Reading pseudocode and algorithm explanations</li> </ul> <p><b>Chapter 5 Networking and Internet Technology</b> (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> <li>● Terminology in computer networks</li> <li>● Reading network architecture and protocol texts</li> <li>● Understanding network-related diagrams</li> </ul> <p><b>Chapter 6 Translation and Application Practice</b> (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> <li>● Translation strategies and sentence conversion</li> <li>● English-to-Chinese and Chinese-to-English translation drills</li> <li>● Final unit translation test and peer review</li> </ul>
Examination forms	Written examination and translation tasks
Study and examination requirements	<p>Complete all in-class exercises and homework</p> <p>Translate selected professional texts</p> <p>Prepare unit vocabulary lists and reading notes</p> <p>Continuous assessment (participation, quizzes, assignments): 60%</p> <p>Final written exam (vocabulary, reading, translation): 40%</p>
Reading list	<p><b>1. Required books</b></p> <p>[1] Compilation Group. English for Computer Science [M]. Beijing: Tsinghua University Press, latest edition.</p> <p><b>2. Reference books</b></p> <p>[1] Qian Zhongshu (Ed.). Computer English Reading Course [M]. Beijing: Higher Education Press.</p> <p>[2] McGraw-Hill. English for Information Technology [M]. International Edition.</p>
Data of last amendment	March 2023