

Taizhou University



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

Appendix D

Syllabus - Practice Courses

Department of Computer Science and Technology
School of Information Engineering

2025

Contents

Professional Practice.....	1
C Programming Project.....	4
Design and Analysis of Algorithms Project.....	7
Software Engineering Project.....	11
Intelligent Application System Development Project.....	14
Professional Internship.....	18
Extracurricular Quality Development Program.....	24

Professional Practice

Module designation	Professional Practice
Semester(s) in which the module is taught	3-6 th semesters
Person responsible for the module	Associate professor Xun Yan
Language	Chinese
Relation to curriculum	Professional internships are arranged after the completion of basic professional courses and some professional courses, aiming to strengthen theoretical knowledge through practice. It is intended to deepen the understanding of technical processes, cultivate engineering ethics, teamwork and problem solving ability for the subsequent major.
Teaching methods	Target students: students of Computer Science and Technology. Type of teaching: practical teaching Contact hour: 30 hours Including: Theoretical teaching: 0 hours Experiment teaching: 30 hours Computer practice: 0 hours Size of class: 40-60 students
Workload (incl. contact hours, self-study hours)	Total workload = 60 hours Contact hours = 30 hours Self-study hours = 30 hours
Credit points	1
Required and recommended prerequisites for joining the module	Database principles, Java programming, C programming, Python programming
Module objectives/intended learning outcomes	Learning outcomes: ● Knowledge: 1. Understand the current situation of technology application and job responsibilities in IT industry, and enhance professional identity. 2. Familiar with enterprise project development process

	<p>and team division mode.</p> <ul style="list-style-type: none"> ● Skill: <p>1.To solve practical problems with professional knowledge and improve the ability to combine theory with practice.</p> <p>2.Develop team work, communication and project execution skills.</p> <ul style="list-style-type: none"> ● Competence: <p>Establish professional norms, strengthen safety responsibility and professional ethics.</p>
Content	<p>Part A. Theoretical teaching</p> <p>(0 contact hours; 0 self-study hours)</p> <p>Part B. Experiment teaching</p> <p>(30 contact hours; 30 self-study hours)</p> <ul style="list-style-type: none"> ● Technical knowledge: Participate in equipment debugging, network maintenance, product research and development and other production activities; ● Management learning: understand the organizational structure, production management and safety standards of the enterprise; ● Industry research: master industry trends and technology trends through lectures and visits.
Examination forms	<p>The college arranges the base internship;</p> <p>Double mentor system: enterprise technicians guide the practice, and school teachers are responsible for theoretical connection and process supervision.</p>
Study and examination requirements	<p>Summary report: Submit a report of more than 3000 words, which should analyze the technical logic and relate to the course knowledge;</p> <p>Five-point grading: The results are divided into excellent, good, medium, pass and fail. Those who fail cannot enter the professional internship</p>
Reading list	<p>1. Required books</p> <p>1. C Programming Language, book number: 9787111617945, China Machine Press;</p> <p>2. Python Programming, book number: 9787115428028, People's Posts and Telecommunications Press;</p> <p>3. Data Structures, ISBN: 9787302147510, Tsinghua University Press;</p> <p>4. Computer Composition Principles, book number: 9787040545180, Higher Education Press;</p>

	<p>5. Computer Network, ISBN: 9789121411748, Electronic Industry Press.</p> <p>2. Reference books</p> <p>None.</p>
Data of last amendment	May 20, 2025

C Programming Project

Module designation	C Programming Project
Semester(s) in which the module is taught	2 nd semester
Person responsible for the module	Associate professor Jiang Min
Language	Chinese
Relation to curriculum	"Course Design for C Programming" serves as a core practical component in the Computer Science and Technology program, functioning as a comprehensive practice following the completion of the "C Programming" course. This initiative aims to deepen students' understanding of fundamental programming concepts, master the process and steps of software design using C language, acquire debugging techniques and methods, enhance their ability to develop programs for solving real-world problems, cultivate skills in integrating theory with practice, logical reasoning, and collaborative innovation, while improving teamwork, communication, and language expression abilities. It prepares students for future careers in computer-related fields and subsequent programming courses Basic knowledge and skills.
Teaching methods	Target students: students of Computer Science and Technology. Type of teaching: theoretical teaching, experiment teaching Contact hour: 30 hours Including: Theoretical teaching: 0 hours Experiment teaching: 30 hours Computer practice: 0 hours Size of class: 40-60 students
Workload (incl. contact hours, self-study hours)	Total workload = 60 hours Contact hours = 30 hours Self-study hours = 30 hours
Credit points	1

Required and recommended prerequisites for joining the module	C programming
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Knowledge: <ol style="list-style-type: none"> 1. Master the ability to write programs to solve practical problems, including the overall process from understanding practical application requirements, problem analysis and model determination, exploration of appropriate solutions, coding implementation, debugging feedback, test analysis, and writing research reports; 2. Master debugging techniques and methods; 3. Understand and master the concepts and theories of C programming. ● Skill: <ol style="list-style-type: none"> 1. Can use C programming technology to solve practical problems; 2. To enable students to have the basic ability of an excellent software developer: to be able to write simple software and to use software development technology to solve practical application problems; 3. Lay the foundation for students to further research and develop software systems. ● Competence: <ol style="list-style-type: none"> 1. Cultivate the sense of standard, students learn to write standard code, familiar with common programming skills. 2. Cultivate students' spirit of cooperation, coordination and organizational management ability. 3. Focus on the development trend and application prospect of the subject, and pay attention to cultivating students' spirit of inquiry into new technologies.
Content	<p>Part A. Theoretical teaching (0 contact hours; 0 self-study hours)</p> <p>Part B. Experiment teaching (30 contact hours; 30 self-study hours)</p> <p>Task 1: Problem formulation and analysis (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> ● Determine the requirements document and conduct problem design and analysis according to the

	<p>requirements of the topic.</p> <ul style="list-style-type: none"> ● After the requirement analysis, the task is assigned in groups. <p>Task 2: Determination of the solution framework (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> ● Determine the framework of the solution according to the function required by the problem; ● Determine the functions of each module in the framework and describe them with functions; ● Analyze the parameters of each subfunction and the invocation relationship between functions. <p>Task 3 Detailed design and coding (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> ● Determine the algorithm process of each function, and draw a flow chart or N-S diagram; ● Write the function code. <p>Task 4 Program debugging and testing (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> ● Debugging methods for different environments (VC6.0, DEVCCP, vc2010) ● Function module debugging; ● Selection of test data and analysis of test results. <p>Task 5: Course Design Report writing (6 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> ● Excel Methods of data statistical analysis. ● Format and requirements for writing scientific reports.
Examination forms	Design report
Study and examination requirements	This course is an examination course, and the overall evaluation score adopts the grade system. The design process (40%), design report (40%) and work attitude and ability (20%) are comprehensively evaluated.
Reading list	<p>1. Required books Zhang Peng, Liu Xin. C Language Course Design Tutorial [M]. Beijing: Tsinghua University Press, 2021.</p> <p>2. Reference books [1] Tan Haoqiang. C Programming [M]. 5th Edition. Beijing: Tsinghua University Press, 2017. [2] Tan Haoqiang. C Programming study guide [M]. Fifth Edition. Beijing: Tsinghua University Press, 2017.</p>
Data of last amendment	Dec 31, 2024

Design and Analysis of Algorithms Project

Module designation	Design and Analysis of Algorithms Project
Semester(s) in which the module is taught	6th semester
Person responsible for the module	Associate professor Jiang Min
Language	Chinese
Relation to curriculum	This course is a key practical module for computer science undergraduates, designed to consolidate the theoretical knowledge learned in the “Algorithm Design and Analysis ” course and cultivate students' problem-solving skills, algorithm design capabilities, programming proficiency, and teamwork. Through independent or group-based project implementation, students gain hands-on experience in algorithm application and software development, and improve their ability to analyze, abstract, and solve practical computational problems.
Teaching methods	Target students: Students majoring in Computer Science and Technology Type of teaching: project-based teaching Contact hours: 32 Including: Theoretical instruction: 4 hours Experimental/project-based instruction: 28 hours Computer practice: included Class size: 30–40 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: Data Structures and Algorithms Programming Fundamentals Recommended: Algorithm Design and Analysis

<p>Module objectives/intended learning outcomes</p>	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Knowledge: <ol style="list-style-type: none"> 1. Understand the full process of algorithm design and project implementation. 2. Master the application of common algorithm paradigms such as divide-and-conquer, greedy, dynamic programming, and graph algorithms in practical scenarios. 3. Understand the evaluation metrics for algorithm correctness and complexity. ● Skill: <ol style="list-style-type: none"> 1. Use appropriate data structures and algorithm strategies to solve computational problems. 2. Implement algorithm-based software solutions using a programming language. 3. Analyze algorithm performance and perform optimization and debugging. ● Competence: <ol style="list-style-type: none"> 1. Work collaboratively to complete algorithmic projects, demonstrating project management and communication abilities. 2. Conduct independent problem analysis and solution abstraction for real-world challenges. 3. Produce high-quality project documentation and deliver professional-level presentations.
<p>Content</p>	<p>Part A. Theoretical teaching (4 contact hours; 4 self-study hours)</p> <p>Chapter 1: Introduction and Project Guidelines (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Overview of course objectives, methodology, and assessment ● Project topic selection and requirement interpretation <p>Chapter 2: Design Strategies and Performance Evaluation (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Algorithm strategy review and evaluation criteria ● Introduction to project design documentation standards <p>Part B. Project-based teaching (28 contact hours; 24 self-study hours)</p> <p>Topic 1: Problem Analysis and Modeling</p>

	<p>(4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Task decomposition and input/output specification ● Modeling real-world problems into algorithmic form <p>Topic 2: Algorithm Design and Optimization (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> ● Algorithmic strategy selection and correctness proof ● Time and space complexity analysis <p>Topic 3: Program Implementation (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> ● Code development using standard programming practices ● Debugging and modular design <p>Topic 4: Testing and Performance Analysis (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Test case generation and boundary testing ● Benchmarking and performance profiling <p>Topic 5: Project Documentation and Presentation (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Writing reports including problem statement, design, implementation, testing ● Oral defense and project demonstration <p>Topic 6: Peer Review and Improvement (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Group feedback and peer evaluation ● Code review and project refinement
Examination forms	Project implementation, written report, and oral presentation
Study and examination requirements	<p>Students must complete their assigned project independently or in teams.</p> <p>Attendance and participation in group meetings and tutorials are required.</p> <p>Final assessment is composed of:</p> <ul style="list-style-type: none"> ● Project deliverables and documentation: 60% ● Project presentation and defense: 30% ● Peer review and participation: 10%
Reading list	<p>1. Required books</p> <p>[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms [M]. MIT Press, 3rd Edition, 2009.</p> <p>2. Reference books</p> <p>[1] Jon Kleinberg, Éva Tardos. Algorithm Design [M]. Pearson, 2005.</p>

	[2] Steven S. Skiena. The Algorithm Design Manual [M]. Springer, 2020.
Data of last amendment	December 15, 2024

Software Engineering Project

Module designation	Software Engineering Project
Semester(s) in which the module is taught	6th semester
Person responsible for the module	Professor Ren Xiangmin
Language	Chinese
Relation to curriculum	This is a core practical module in the Computer Science and Technology undergraduate program. It serves as a bridge between software engineering theory and real-world software development. The course enhances students' ability to apply the software engineering development lifecycle, including requirements analysis, system design, implementation, and testing. It aims to consolidate theoretical knowledge and cultivate teamwork, engineering practice, and comprehensive software development competence.
Teaching methods	Target students: Students majoring in Computer Science and Technology Type of teaching: project-based practical training Contact hours: 32 Including: Theoretical instruction: 4 hours Experimental/project-based instruction: 28 hours Computer practice: included Class size: 30–40 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: Fundamentals of Programming Software Engineering Recommended: Database Principles and Applications Java Programming Basics

<p>Module objectives/intended learning outcomes</p>	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Knowledge: <ol style="list-style-type: none"> 1. Understand the principles, processes, and tools of software engineering. 2. Grasp standard development models and collaborative methods for team software development. 3. Understand the organization and management of software projects, as well as the principles of testing and deployment. ● Skill: <ol style="list-style-type: none"> 1. Apply software engineering methods to complete team-based software development tasks. 2. Perform requirements analysis, system design, implementation, testing, and documentation. 3. Use common development tools (such as version control, modeling tools, IDEs). ● Competence: <ol style="list-style-type: none"> 1. Demonstrate teamwork, division of labor, and collaboration in software project development. 2. Integrate engineering thinking and practice in real development scenarios. 3. Communicate effectively in project documentation and presentations.
<p>Content</p>	<p>Part A. Theoretical teaching (4 contact hours; 4 self-study hours)</p> <p>Chapter 1: Overview of Software Engineering Projects (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Introduction to practical course objectives and methods ● Team organization and task allocation <p>Chapter 2: Software Project Planning and Design (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Requirements analysis and system modeling ● Software design documentation standards <p>Part B. Experimental/project-based teaching (28 contact hours; 24 self-study hours)</p> <p>Topic 1: Requirements Analysis and Use Case Design (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Interview simulation, use case diagram drafting

	<p>Topic 2: Software System Design (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● UML modeling (class diagram, sequence diagram) <p>Topic 3: Programming and Implementation (8 contact hours; 6 self-study hours)</p> <ul style="list-style-type: none"> ● Modular development, code implementation <p>Topic 4: Software Testing and Debugging (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Unit testing, integration testing <p>Topic 5: Project Summary and Presentation (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Report writing, team presentation <p>Topic 6: Peer Evaluation and Reflection (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Peer review and group feedback
Examination forms	Group project deliverables and oral presentation
Study and examination requirements	<p>Each student must complete tasks within their team independently and responsibly.</p> <p>Regular attendance and participation in project meetings are mandatory.</p> <p>Assessment includes:</p> <ul style="list-style-type: none"> ● Project implementation and documentation: 60% ● Final presentation and defense: 30% ● Participation and peer evaluation: 10%
Reading list	<p>1. Required books</p> <p>[1] Ian Sommerville. Software Engineering (10th Edition) [M]. Pearson Education, 2016.</p> <p>2. Reference books</p> <p>[1] Roger S. Pressman. Software Engineering: A Practitioner's Approach [M]. McGraw-Hill Education, latest edition.</p> <p>[2] Craig Larman. Applying UML and Patterns [M]. China Machine Press, 2020.</p>
Data of last amendment	June 2022

Intelligent Application System Development Project

Module designation	Intelligent Application System Development Project
Semester(s) in which the module is taught	6th semester
Person responsible for the module	Lecturer Xu Zilong
Language	Chinese
Relation to curriculum	This is a core practical course in the Computer Science and Technology program. It aims to strengthen students' engineering design and implementation capabilities through a comprehensive system development project. The course helps consolidate knowledge from previous theoretical modules such as databases, software engineering, front-end/back-end development, and artificial intelligence applications. It promotes teamwork, innovation, and problem-solving skills necessary for real-world intelligent system construction.
Teaching methods	<p>Target students: Undergraduate students majoring in Computer Science and Technology</p> <p>Type of teaching: Project-based experimental teaching</p> <p>Contact hours: 32</p> <p>Including:</p> <p style="padding-left: 20px;">Theoretical instruction: 4 hours</p> <p style="padding-left: 20px;">Project design and implementation: 28 hours</p> <p style="padding-left: 20px;">Computer practice: 0 hours</p> <p style="padding-left: 20px;">Class size: 4–6 students per project team</p>
Workload (incl. contact hours, self-study hours)	<p>Total workload: 60 hours</p> <p>Contact hours: 32</p> <p>Self-study hours: 28</p>
Credit points	2
Required and recommended prerequisites for joining the module	<p>Required:</p> <p style="padding-left: 20px;">Database Systems</p> <p style="padding-left: 20px;">Software Engineering</p>
Module objectives/intended	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Knowledge: <ol style="list-style-type: none"> 1. Understand the full life cycle of intelligent system

learning outcomes	<p>development.</p> <p>2. Apply architectural design principles and project management methods.</p> <p>● Skill:</p> <p>1. Design, develop, test, and deploy intelligent applications in team settings.</p> <p>2. Integrate front-end and back-end technologies, databases, and intelligent modules.</p> <p>3. Document the system, write technical reports, and conduct presentations.</p> <p>● Competence:</p> <p>1. Analyze user needs and transform them into system functions.</p> <p>2. Collaborate effectively in teams and manage project progress.</p> <p>3. Demonstrate engineering literacy and innovation in system design.</p>
Content	<p>Part A. Theoretical teaching (4 contact hours; 4 self-study hours)</p> <p>Chapter 1 Course Introduction and Team Setup (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● Introduction to intelligent application systems ● Review of relevant technologies and prerequisites ● Formation of student project teams ● Project scope and requirements interpretation <p>Chapter 2 System Design and Engineering Planning (2 contact hours; 2 self-study hours)</p> <ul style="list-style-type: none"> ● System architecture design ● Function module decomposition ● Development scheduling and team task division <p>Part B. Experiment teaching (28 contact hours; 24 self-study hours)</p> <p>Experiment 1 Requirement Analysis and System Specification (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Create use case diagrams and system requirement documents ● Conduct feasibility analysis and user modeling <p>Experiment 2 Database and Backend Module Implementation (6 contact hours; 5 self-study hours)</p>

	<ul style="list-style-type: none"> ● Build and test core backend functions ● Implement database structure and data logic <p>Experiment 3 Front-end Interface and Interaction Design (6 contact hours; 5 self-study hours)</p> <ul style="list-style-type: none"> ● UI/UX design and front-end code development ● Real-time interaction and page layout optimization <p>Experiment 4 Intelligent Component Integration (4 contact hours; 4 self-study hours)</p> <ul style="list-style-type: none"> ● Embed AI modules (e.g., NLP, image processing, recommendation) ● API integration and model invocation <p>Experiment 5 Testing, Deployment and Team Presentation (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Conduct functional testing and system debugging ● Prepare team report, poster, and final presentation <p>Experiment 6 Project Evaluation and Reflection (4 contact hours; 3 self-study hours)</p> <ul style="list-style-type: none"> ● Project demonstration and peer review ● Instructor feedback and course summary discussion
Examination forms	Project deliverables, documentation, and team presentation
Study and examination requirements	<p>Complete each development phase within team deadlines</p> <p>Submit system documentation and development report</p> <p>Attend and present during final project defense</p> <p>Continuous assessment (teamwork, process, outcome): 60%</p> <p>Final project presentation and report: 40%</p>
Reading list	<p>1. Required books</p> <p>[1] Pressman, R. S. Software Engineering: A Practitioner's Approach [M]. New York: McGraw-Hill, latest edition.</p> <p>[2] Wenguang Chen. Design and Practice of Intelligent Application Systems [M]. Beijing: Tsinghua University Press.</p> <p>2. Reference books</p> <p>[1] Ian Sommerville. Software Engineering [M]. Pearson Education, latest edition.</p> <p>[2] Liang Jie. Full Stack Development Practice [M]. Beijing: Posts and Telecom Press.</p>

Data of last amendment	May 2022
------------------------	----------

Professional Internship

Module designation	Professional Internship
Semester(s) in which the module is taught	7 th semester
Person responsible for the module	Associate professor Xu Yan
Language	Chinese
Relation to curriculum	<p>This course is a concentrated practical teaching link of computer science and technology, Internet of Things engineering and other majors.</p> <p>As a crucial component of the curriculum, this course serves as a vital practical training phase before graduation, bridging theoretical knowledge with real-world applications. Through hands-on practice, students gain insights into developing computer technologies for specific fields, deepen their understanding of foundational theories, and develop practical skills. The program also cultivates professional ethics, social responsibility, teamwork capabilities, communication proficiency, and self-directed learning abilities essential for career success in technical roles.</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: 90 hours</p> <p>Including:</p> <p>Theoretical teaching: 90 hours</p> <p>Experiment teaching: 90 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 = 180 hours</p> <p>Contact hours = 90 hours</p> <p>Self-study hours = 90 hours</p>
Credit points	3
Required and recommended	Database principles, Java programming, C programming, Python programming

prerequisites for joining the module	
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Knowledge: <ol style="list-style-type: none"> 1. Understand the current situation of information construction and practical application scenarios of the internship unit. 2. Master the process, operation technology and basic decision-making methods of computer development/application positions. 3. Deepen the understanding and application of professional theoretical knowledge through post practice. ● Skill: <ol style="list-style-type: none"> 1. Proficient in computer technology to perform development, application or operation and maintenance tasks. 2. Apply professional knowledge to analyze practical problems in the position, propose feasible solutions and implement them. 3. Identify defects in existing systems, technologies or processes and propose improvement suggestions. ● Competence: <ol style="list-style-type: none"> 1. Understand the impact of information application on society, security, law and environment, abide by professional ethics and establish a sense of social responsibility. 2. Take the initiative to collaborate with team members and other personnel to clearly express opinions. 3. Take the initiative to learn new technologies and procedures, and continuously improve professional competence. 4. Discover systematic defects through observation and put forward innovative solutions.
Content	<p>Part A. Theoretical teaching (0 contact hours; 0 self-study hours)</p> <p>Part B. Experiment teaching (90 contact hours; 90 self-study hours)</p> <p>(1) Internship content Students should complete 1-2 of the following internship tasks according to their specific conditions.</p> <ol style="list-style-type: none"> 1. Participate in the installation, debugging and testing of

	<p>computer-related systems;</p> <p>2. Participate in the development of computer-related systems, such as: requirement analysis, program design, programming, debugging and testing, documentation and collation;</p> <p>3. Participate in the management, maintenance and optimization of computer-related systems;</p> <p>4. Participate in the practical application of computer-related systems;</p> <p>The above computer-related systems include computer software system, hardware system, network system, website system, embedded system, Internet of things system and other single system or multiple system integrated system.</p> <p>(2) Internship requirements</p> <p>1. Strictly comply with the school's regulations. Students are required to maintain punctuality during internships, avoiding tardiness, early departures, or unexcused absences, and must obtain proper leave authorization for any absence. According to Article 15 of the Academic Records Management Regulations: The number of class hours missed for professional training activities is calculated as 8 hours/day during regular classes and 4 hours/day for internship/practical sessions (excluding statutory holidays). Article 17 stipulates that students who miss more than one-third of a course's teaching hours may be disqualified from the course assessment and required to retake it. Article 42 specifies that students who fail compulsory courses or restricted electives within a single academic year will face disciplinary consequences Up to 30 credits will be treated as withdrawal.</p> <p>2. Carefully implement the relevant principles and policies of the Party and the state, strictly implement the rules and regulations of the internship unit, strictly guard the business secrets of the enterprise, treat all aspects of the internship seriously with an owner's attitude, and complete the tasks required by the internship unit on time.</p> <p>3. Under the guidance of the instructor and relevant personnel of the internship unit, conscientiously complete the internship tasks assigned by the instructor and the</p>
--	---

	<p>internship unit. Go deep into the practice, and complete the internship tasks in time, with good quality and in full quantity.</p> <p>4. Keep the internship records at any time, fill in the internship report, internship appraisal form and other relevant contents carefully. At the end of the internship, complete the internship summary.</p> <p>5. Learn from the staff of the internship unit with an open mind, learn from practice seriously, and strive to cultivate the ability to find, analyze and solve practical problems.</p> <p>6. The trainees should focus on the internship content, conduct analysis and research based on the information collected during the internship, put forward their own opinions, and prepare for the writing of the graduation thesis.</p>
Examination forms	On-the-job internship
Study and examination requirements	<p>1. Requirements for assessment materials</p> <p>(1) Internship Evaluation Form. The internship evaluation form serves as an assessment document for students 'on-the-job training, comprising three sections: Self-Assessment, Institutional Evaluation, and Supervisor's Evaluation. The Self-Assessment requires a concise summary of the student's main achievements, gains, and existing issues during the internship. The Institutional Evaluation involves a comprehensive assessment and scoring by the internship unit's leadership. The Supervisor's Evaluation is conducted by the school's internship supervisors based on the student's performance, with the final evaluation score being calculated through a combination of Institutional Evaluation and Supervisor's Evaluation "Teacher appraisal" and "unit appraisal" each account for 50%.</p> <p>(2) Internship log. The internship log records the completion of students' daily internship tasks.</p> <p>(3) Internship weekly report. The internship weekly report records the main internship tasks completed each week, existing problems and personal reflection.</p> <p>(4) Internship report. The internship report is a summary material of students' on-the-job internship. Its content mainly includes personal understanding and performance of the main work of the internship, main</p>

	<p>content and completion of the internship work, main harvest and experience of the internship, and the report should be no less than 3000 words.</p> <p>2. Assessment of achievements</p> <p>The internship results are graded on a five-level scale, consisting of two parts: the score of the internship unit supervisor and the score of the school supervisor, each accounting for 50%. The calculation result is converted into grades according to the following standards: excellent (90~100 points), good (80~89.5 points), medium (70~79.5 points), pass (60~69.5 points) and fail (below 60 points).</p> <p>The specific indicators are as follows:</p> <p>(1) Enterprise mentor scoring standard</p> <p>① Attitude towards internship (30 points)</p> <p>Consciously abide by all rules and regulations of both the internship organization and the school. Maintain punctuality with no tardiness or early departures, take leave only under special circumstances, and refrain from any illegal or disciplinary violations. Comply with assignments from instructors and supervisors, demonstrating proactive work ethic, strong sense of responsibility, and dedication to professional duties. Cultivate humility, diligence in learning, steady decision-making, and warm interpersonal relationships. Strictly uphold professional ethics to safeguard the reputation of both the internship organization and the school.</p> <p>② Business ability (50 points)</p> <p>Complete the internship work, master the necessary basic skills; good at applying the theoretical knowledge to practice, have unique opinions on problems, strong analytical ability, quick thinking.</p> <p>③ Internship performance (20 points)</p> <p>My independent working ability has been greatly improved; I have demonstrated good social adaptability, psychological endurance, organization and coordination ability and communication skills during the internship; I have been praised by all sides.</p> <p>(2) School tutor scoring criteria</p> <p>① Attendance (20 points)</p>
--	--

	<p>Sign in at Alumni Bank every day: no less than 10 weeks, a total of no less than 40 times, each absence will be deducted 1 point; if the attendance is less than 35 times, the attendance score is 0 points; if the attendance is less than 30 times, the internship is not qualified.</p> <p>② Internship log (20 points)</p> <p>Record the internship situation every day: the ratio of the number of internship logs to the number of check-ins should not be lower than 90%, and 1 point will be deducted for every 2 percentage points below; if the number of internship logs is less than 70%, the internship log score will be 0 points; if the number of internship logs is less than 50%, the internship will be unqualified.</p> <p>③ Internship report (40 points)</p> <p>Each weekly report is one week's worth, and the basic score of the internship weekly report is 0.4 times the average score of all weekly reports. If there is one less weekly report, 5 points will be deducted; if there are fewer than 8 internship weekly reports, the score of the internship weekly report will be 0 points; if there are fewer than 6 internship weekly reports, the internship will fail.</p> <p>④ Internship report (20 points)</p> <p>Requirements: The content should be true, comprehensive and general, consistent with the internship weekly report, with standard format, concise language, clear regulations and reasonable logic. Points will be awarded according to the actual writing of the requirements.</p>
Reading list	<p>Required books</p> <ol style="list-style-type: none"> 1. C Programming Language, book number: 9787111617945, China Machine Press, 2019. 2. Python Programming, book number: 9787115428028, People's Posts and Telecommunications Press, 2020 3. Data Structures, ISBN: 9787302147510, Tsinghua University Press, 2015. 4. Computer Composition Principles, book number: 9787040545180, Higher Education Press; 5. Computer Network, book number: 9789121411748, Electronic Industry Press, 2012.
Data of last amendment	May 20, 2025

Extracurricular Quality Development Program

Module designation	Extracurricular Quality Development Program
Semester(s) in which the module is taught	1-8 th semesters
Person responsible for the module	Lecturer Tan Lixing
Language	Chinese
Relation to curriculum	<p>This program enhances students' comprehensive qualities and practical skills, fostering holistic development beyond academics. Through systematic module design covering communication, teamwork, leadership, innovative thinking, career exploration, and mental health, it employs diverse teaching methods to unlock students' potential while strengthening social adaptability and lifelong learning capabilities.</p> <p>The curriculum emphasizes experiential and project-based learning, encouraging active participation and reflection in real or simulated scenarios. It promotes interdisciplinary integration and hands-on exploration. Through this program, students will develop sound values, enhance soft skills, and build a solid foundation for future academic pursuits, career development, and social engagement.</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: 120 hours</p> <p>Including:</p> <p>Theoretical teaching: 0 hours</p> <p>Experiment teaching: 120 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 = 240 hours</p> <p>Contact hours = 120 hours</p> <p>Self-study hours = 120 hours</p>
Credit points	8

Required and recommended prerequisites for joining the module	None.
Module objectives/intended learning outcomes	<p>Learning outcomes:</p> <ul style="list-style-type: none"> ● Knowledge: Develop students' comprehensive quality, including communication, expression, collaboration, leadership, innovation and other abilities ● Skill: Broaden students' knowledge horizon, enhance social responsibility and practical ability ● Competence: Enrich students' campus cultural life and promote the all-round development of morality, intelligence, physical fitness, beauty and labor
Content	<p>Part A. Theoretical teaching (0 contact hours; 0 self-study hours)</p> <p>Part B. Experiment teaching (120 contact hours; 120 self-study hours)</p> <p>Chapter 1 Self-knowledge and personality development (20 contact hours; 20 self-study hours)</p> <ul style="list-style-type: none"> ● Models of self-perception (Johari Window, Iceberg model) ● Personality analysis tools: MBTI, DISC or Big Five ● Values clarification exercise and exploration of the meaning of life ● Goal setting methods (SMART principles) and time management techniques <p>Chapter 2: Communication and Expression Skills (20 contact hours; 20 self-study hours)</p> <ul style="list-style-type: none"> ● The three elements of communication: content, way and emotion ● Oral expression skills: structured expression, persuasive and infectious ● Listen and feedback: Listen actively and respond with empathy ● Nonverbal communication: body language, facial expressions, tone of voice

	<p>Public speaking and presentation skills</p> <p>Chapter 3: Teamwork and Leadership (20 contact hours; 20 self-study hours)</p> <ul style="list-style-type: none"> ● Team development (Tuckman model) and effective collaboration ● Team role cognition and division of labor (Belbin theory) ● Situation leadership model and leadership style identification ● Conflict identification and mediation strategies <p>Chapter 4: Innovation and Problem Solving (20 contact hours; 20 self-study hours)</p> <ul style="list-style-type: none"> ● Innovation thinking tools: SCAMPER, six Thinking hats ● Creative generation and selection: brainstorming, divergent and convergent thinking ● Design thinking process: same-definition-conception-prototype-test ● Problem analysis methods: fishbone diagram, 5Why rule <p>Chapter 5: Public Service and Social Responsibility (20 contact hours; 20 self-study hours)</p> <ul style="list-style-type: none"> ● The definition and classification of public services: education, environmental protection, community, etc ● Citizen responsibility and youth volunteerism ● Community research and public welfare project planning process ● Social issue identification and introduction of Sustainable Development Goals (SDGs) <p>Chapter 6: Professional quality and career exploration (20 contact hours; 20 self-study hours)</p> <ul style="list-style-type: none"> ● Career personality and interest assessment (Holland Type 6, career anchor) ● Basic format and key points of resume and cover letter ● Behavioral interview skills (STAR method) and practical simulation ● Industry exploration and future trend analysis (such as AI, green economy, freelance, etc.)
Examination forms	Overall merit

Study and examination requirements	<p>1. Classroom performance and attendance (15%) Attendance, classroom engagement, group participation Observation and recording of behaviors such as questioning, expressing, and collaborating</p> <p>2. Progress of phased operations and completion of activities (25%) For example, MBTI test report, speech practice, group plan, public welfare project design draft, etc Layout and submission by module, focusing on process and quality of completion</p> <p>3. Results of the Team Practice Project (30 per cent) Including project idea, solution design, teamwork, final result presentation (such as video, display board, report PPT, etc.)</p> <p>4. Personal growth portfolio or reflection report (15%) Including self-cognition, learning gains, growth plans, ability improvement reflection, etc It can be presented in the form of learning logs, visual files and written summaries Peer review and self-assessment (10%) Each student evaluates himself or herself and the other members of the group to promote fairness and self-knowledge</p>
Reading list	<p>Reference books</p> <p>[1] Covey S R. The 7 habits of highly effective people [M]. 3rd ed. Beijing: China Youth Press, 2012.</p> <p>[2] Goleman D. Emotional intelligence: Why it can matter more than IQ [M]. Beijing: CITIC Publishing House, 2009.</p> <p>[3] Hanh T N. The miracle of mindfulness [M]. Haikou: Hainan Publishing House, 2017</p>
Data of last amendment	July 2025