# Course Description

*Computer Science and Technology*

*College of Computer Science*

*– University*

# Contents

# 1 CORE COURSE

## 1.1 High Level Language Program Design 2-1

| | |
|---|---|
| **Course Name** | High Level Language Program Design 2-1 |
| **Credits** | 3.5 credits = 5.25 ECTS |
| **Credit Hours** | 80 |
| **Semester** | 2019-2020/1 |
| **Description** | This course lays a crucial foundation for computer science and related majors, focusing on practical programming skills and computational thinking. It covers C++ fundamentals such as data types, operators, control structures, and functions, enabling students to solve engineering problems. Key topics include arrays, pointers, and the evolution of C++ standards across platforms. With project-based learning, students gain hands-on experience in software design, development methods, and modern engineering tools. By working on comprehensive projects, they will develop the skills necessary for the full software development cycle, providing a solid base for advanced studies and graduation projects. |
| **Education Aims** | 1. Understand and master basic concepts in C++ such as data types, operators, and expressions, and use structured programming methods to implement complex computer engineering solutions.<br>2. Grasp key concepts related to C++ control statements and functions, and identify critical aspects of complex engineering problems.<br>3. Learn the differences between C++ compilation platforms, understand the evolution of C++ standards, and use advanced debugging/testing tools. Gain proficiency in C++ arrays, pointers, and the associated programming techniques.<br>4. Through project-based learning, acquire the methods and technologies for full-cycle software design and development, enhancing programming and software engineering skills. |
| **Main Contents** | **Chapter 1:** Introduction to C++ Programming (3 hours)<br>• **Topics:** Overview of programming languages, numeral systems, data storage, basic concepts of C++, structure of C++ programs.<br>• **Key Concepts:** Binary data storage, numeral system conversions, literal constants.<br>• **Difficulties:** Converting between numeral systems and understanding different types of literals.<br>**Chapter 2:** Data Types (3 hours)<br>• **Topics:** Basic data types, compound data types, enumeration, and cv-qualified data types.<br>• **Key Concepts:** Data representation in memory, equivalence, and type conversions.<br>• **Difficulties:** Initialization methods, memory storage forms of different data types.<br>**Chapter 3:** Operators and Expressions (3 hours)<br>• **Topics:** Assignment, arithmetic, relational, logical, bitwise, conditional operators.<br>• **Key Concepts:** Expression evaluation.<br>• **Difficulties:** Operator precedence and associativity.<br>**Chapter 4:** Control Structures, Arrays, Structures, and Unions (15 hours)<br>• **Topics:** Branch, loop, and jump statements, arrays, structures, and unions.<br>• **Key Concepts:** Programming with control structures and arrays.<br>• **Difficulties:** Nested loops, array-pointer relationships. |

| | |
|---|---|
| | **Chapter 5:** Functions (12 hours)<br>• **Topics:** Function declarations, recursion, operator overloading.<br>• **Key Concepts:** Function calls, recursion, parameter passing, scope, and lifetime of variables.<br>• **Difficulties:** Recursive function calls and static variables.<br>**Chapter 6:** Pointers, References, and Dynamic Memory Allocation (12 hours)<br>• **Topics:** Pointers, dynamic memory, linked lists, references.<br>• **Key Concepts:** Dynamic memory allocation, pointer arithmetic.<br>• **Difficulties:** Implementing and applying linked lists.<br>**Laboratory Sessions:**<br>• Practical exercises including C++ setup, basic data types, operators, control structures, arrays, functions, and pointers. Students will use online evaluation systems to complete and submit their assignments. |
| **Evaluation** | The course assessment is a blend of continuous assessment and a final exam, with the following breakdown:<br>• **Self-study assessment (5%):** Based on video completion and pre-class quizzes, supporting course objective 2.<br>• **In-class quizzes (5%):** Evaluated based on the accuracy of in-class quizzes, supporting course objective 2.<br>• **Lab work assessment (10%):** Evaluated through the completion of programming assignments, supporting course objectives 2 and 3 (each contributing 50%).<br>• **Comprehensive project development (10%):** Assessment based on the completion of a project, supporting course objective 4.<br>• **Comprehensive lab skills (20%):** Assessed via an on-computer exam, supporting course objectives 1, 2, and 3 (contributing 15%, 35%, and 50%, respectively).<br>• **Final Exam (50%):** A closed-book exam covering various C++ topics, supporting course objectives 1, 2, 3, and 4. It covers multiple-choice questions, code correction, code analysis, and programming tasks, each designed to reinforce specific course objectives. |
| **Study Materials** | • *C++ Programming (2nd Edition)*, by Liu Jing, Higher Education Press, ISBN 9787040354560, 2013<br>• *The C++ Programming Language*, by Bjarne Stroustrup, Mechanical Industry Press, ISBN 9787111539414, 2016<br>• *C++ Primer (Chinese Edition)*, by S.B. Lippman and J. Lajoie, Electronics Industry Press, ISBN 9787121155352, 2014 |
| **Language** | Chinese |
| **Final Grade** | 4.0 |

# 2 NON-MAJOR COURSE

## 2.1 Course B

| | |
|---|---|
| **Course Name** | Course B |
| **Credits** | 4.0 |
| **Semester** | Fall |