

Title: Computer Programming

Code Number: CS1202

Credit Hours: 3 (3+1)

Prerequisites: Nil

Semester: 2nd

Course Objectives

The course will enable students to:

1. Apply knowledge of C++ basic features to interpret various kinds of mathematical and engineering problems
2. Design solutions for advanced computing problems, integrating principles of object-oriented programming
3. Demonstrate proficiency in executing various computer programming techniques to solve complex problems, effectively utilizing programming language and tools with precision and accuracy. the solution of searching problems with data structures principles
4. Communicate understanding of computer programming concepts, methodologies, and problem-solving strategies.

Contents

Unit 1: Introduction to Programming in C++

1. Introduction to Programming
2. Introduction to Object Technology
3. Typical C++ Development Environment
4. Compiling and Running an Application in Visual Studio 2015
5. Compiling and Running Using GNU C++

Unit 2: Introduction to C++ Programming

1. Input/Output and Operators: Program in C++
2. Printing
3. Adding Integers
4. Arithmetic
5. Equality and Relational Operators

Unit 3: Algorithm Development and Control Statements

1. Algorithms
2. Pseudocode
3. Control Structures
4. Selection Statements
5. Iteration Statements
6. if Single/ double-Selection Statement
7. Nested if...else Statements
8. Conditional Operator (?:)
9. While Iteration Statement
10. Counter-Controlled Iteration
11. For Iteration Statement
12. Do-While Iteration Statement
13. Switch Multiple-Selection Statement

14. Break and continue Statements
15. Logical AND (&&) Operator
16. Logical OR (||) Operator
17. Logical Negation (!) Operator

Unit 4: Functions and an Introduction to Recursion

1. Math Library Functions
2. Function Prototypes
3. Argument-Coercion
4. C++ Standard Library Headers
5. Scope Rules
6. Inline Functions
7. Function Overloading
8. Function Templates, Recursion

Unit 5: Arrays & Pointers

1. Basic concepts of arrays
2. Declaring arrays, array with loops
3. Sorting and Searching arrays
4. Multidimensional arrays
5. Pointer Variable Declarations and Initialization
6. Pointer Operators
7. Pass-by-Reference with Pointers
8. Pointer Expressions and Pointer Arithmetic

Unit 6: Classes

1. Creating class
2. Objects
3. Member functions
4. Interface of a Class
5. Scope Resolution Operator (::)
6. Member Functions vs. Global Functions
7. Constructors and Destructors
8. Composition: Objects as Members of Classes
9. Friend Functions
10. Friend Classes
11. Using the this Pointer
12. Static Class Members
13. Fundamentals of Operator Overloading
14. Base Classes and Derived Classes
15. Introduction to Polymorphism

Unit 7: Templates and Header Files

1. Introduction to Templates
2. Generic Programming with Templates
3. Function Templates
4. Class Templates
5. Template Specialization
6. Header Files and their Role

Lab Work Outline

The lab course for Introduction to Programming in C++ begins with setting up the development environment, where students will install and configure Visual Studio 2015 and the GNU C++ compiler. They will create, compile, and run a simple "Hello, World!" program in both

environments to familiarize themselves with these tools. Following this, the focus shifts to basic input/output operations and the use of operators in C++, where students will write programs to read user input, perform arithmetic operations, and use relational and equality operators to compare values and display results. Next, students will delve into control structures and algorithm development. They will write pseudocode for simple problems and implement it using if, if-else, and switch statements, as well as develop iteration statements such as while, for, and do-while loops, experimenting with break and continue statements within these loops. The course then moves to functions and recursion, where students will learn to define and use various functions, including function overloading and recursion, by creating functions with different return types and parameters, and observing the impact of inline functions. In the arrays and pointers section, students will work with arrays by declaring, initializing, sorting, and searching them, as well as implement multi-dimensional arrays. They will also explore pointer variables, including declaration, initialization, and arithmetic, and learn how to pass arrays and pointers to functions. The course continues with an in-depth look at classes, where students will create classes with member variables and functions, instantiate objects, use member functions, and implement constructors and destructors. They will also experiment with friend functions, friend classes, the this pointer, and static members. The final part of the lab course covers advanced class features, including operator overloading, creating base and derived classes to understand inheritance, and developing polymorphic behavior using virtual functions. Students will also learn about composition by using objects as members of other classes. Throughout the course, students are expected to document their work in lab reports, which should include the objective, procedure, results, conclusion, and answers to any additional questions provided by the instructor. This comprehensive approach ensures that students gain practical experience and reinforce their theoretical knowledge in C++ programming.

Teaching-Learning Strategies:

The pedagogical approach to this course relies on face-to-face teaching in a university classroom environment. The lectures are delivered using multimedia support and on white board. Students are engaged and encouraged to solve real world problems using computer-aided tools.

Assignments/Types and Number with calendar:

A minimum of four assignments to be submitted before the written exams for each term.

Assessment and Examinations:

Sr. No.	Elements	Weightage	Details
1.	Midterm Assessment	35%	It takes place at the mid-point of the semester.
2.	Sessional Assessment	25%	It is continuous assessment. It includes classroom participation, attendance, assignments and presentations, homework, attitude and behavior, hands-on-activities, short tests, quizzes etc.
3.	Final Assessment	40%	It takes place at the end of the semester. It is mostly in the form of a test, but owing to the nature of the course the teacher may assess their students based on term paper, research proposal development, field work and report writing etc.

Recommended Books:

1. Dietel and Dietel, "C++ How to Program", Prentice Hall; 9th edition (2013).
2. Patt and Patel, "Introduction to Computing Systems from Bits and Gates to C and Beyond", McGraw-Hill, 2nd edition (2004).
3. Kernighan and Ritchie, "The C programming language", Latest Edition.