DSC04/DSC01/GE1b: PROGRAMMING USING C++

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre- requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Programming using C++	4	3	0	1	Class XII pass	NIL

Course Objectives:

This course is designed to introduce programming concepts using C++ to students. The course aims to develop structured as well as object-oriented programming skills using C++ programming language. The course also aims to achieve competence amongst its students to develop correct and efficient C++ programs to solve problems spanning multiple domains.

Learning outcomes

On successful completion of the course, students will be able to:

- Write simple programs using built-in data types of C++.
- Implement arrays and user defined functions in C++.
- Write programs using dynamic memory allocation, handling external files, interrupts and exceptions.
- Solve problems spanning multiple domains using suitable programming constructs in C++.
- Solve problems spanning multiple domains using oriented programming concepts in C++.

Syllabus

Unit-1 **(3 hours)**

Introduction to C++: Overview of Procedural and Object-Oriented Programming, Using main() function, Header Files, Compiling and Executing Simple Programs in C++.

Unit-2 **(12 hours)**

Programming Fundamentals: Data types, Variables, Operators, Expressions, Arrays, Keywords, Decision-making constructs, Iteration, Type Casting, Input-output statements, Functions, Command Line Arguments/Parameters

Unit-3 **(15 hours)**

Object Oriented Programming: Concepts of Abstraction, Encapsulation. Creating Classes and objects, Modifiers and Access Control, Constructors, Destructors, Implementation of Inheritance and Polymorphism, Template functions and classes

Unit-4 **(9 hours)**

Pointers and References: Static and dynamic memory allocation, Pointer and Reference Variables, Implementing Runtime polymorphism using pointers and references

Unit-5 **(6 hours)**

Exception and File Handling: Using try, catch, throw, throws and finally; Nested try, creating user defined exceptions, File I/O Basics, File Operations

Practical **(30 hours)**

1. Write a program to compute the sum of the first n terms of the following series: The number of terms n is to be taken from the user through the command line. If the command line argument is not found then prompt the user to enter the value of n.
2. Write a program to remove the duplicates from an array.
3. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
4. Write a menu driven program to perform string manipulation (without using inbuilt string functions):
   - **a.** Show address of each character in string
   - **b.** Concatenate two strings.
   - **c.** Compare two strings
   - **d.** Calculate length of the string (use pointers)
   - **e.** Convert all lowercase characters to uppercase
   - **f.** Reverse the string
   - **g.** Insert a string in another string at a user specified position
5. Write a program to merge two ordered arrays to get a single ordered array.

6. Write a program to search a given element in a set of N numbers using Binary search
   a. with recursion
   b. without recursion.
7. Write a program to calculate GCD of two numbers
   a. with recursion
   b. without recursion.
8. Create a Matrix class. Write a menu-driven program to perform following Matrix operations (exceptions should be thrown by the functions if matrices passed to them are incompatible and handled by the main() function):
   a. Sum
   b. Product
   c. Transpose
9. Define a class Person having name as a data member. Inherit two classes Student and Employee from Person. Student has additional attributes as course, marks and year and Employee has department and salary. Write display() method in all the three classes to display the corresponding attributes. Provide the necessary methods to show runtime polymorphism.
10. Create a Triangle class. Add exception handling statements to ensure the following conditions: all sides are greater than 0 and sum of any two sides are greater than the third side. The class should also have overloaded functions for calculating the area of a right angled triangle as well as using Heron's formula to calculate the area of any type of triangle.
11. Create a class Student containing fields for Roll No., Name, Class, Year and Total Marks. Write a program to store 5 objects of Student class in a file. Retrieve these records from the file and display them.
12. Copy the contents of one text file to another file, after removing all whitespaces.

Essential/recommended readings

1. Stephen Prata, C++ Primer Plus, 6th Edition, Pearson India, 2015.
2. E Balaguruswamy, Object Oriented Programming with C++, 8th edition, McGraw- Hill Education, 2020. 3. D.S. Malik, C++ Programming: From Problem Analysis to Program Design, 6th edition, Cengage Learning, 2013.

Suggestive Readings

1. Schildt, H. C++: The Complete Reference, 4th edition, McGraw Hill, 2003

2. Forouzan, A. B., Gilberg, R. F. Computer Science: A Structured Approach using C++, 2nd edition, Cengage Learning, 2010

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DSC02/DSC03/GE2c: COMPUTER SYSTEM ARCHITECTURE

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Computer System Architecture	4	3	0	1	Class XII pass	NIL

Course Objectives

The objectives of this course are as follows:

• Introduces the students to the fundamental concepts of digital computer organization, design and architecture.
• Develop a basic understanding of the building blocks of the computer system and highlight how these blocks are organized together to architect a digital computer system.

Learning Outcomes

On successful completion of the course, students will be able to:

• Design Combinational Circuits using basic building blocks. Simplify these circuits using Boolean algebra and Karnaugh maps. Differentiate between combinational circuits and sequential circuits.
• Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
• Determine various stages of instruction cycle, pipelining and describe interrupts and their handling.
• Describe how CPU communicates with memory and I/O devices.