

Annexure-LXXIII
Item-4-1-33
AC-03.08.2022

UNIVERSITY OF DELHI

Multidisciplinary Courses of Study in Computer Science with Three Core Disciplines

(SEMESTER-I)

based on

Undergraduate Curriculum Framework 2022 (UGCF)

(Effective from Academic Year 2022-23)



University of Delhi

DSC01:- Programming fundamentals Using C++

Course Title	Nature of the Course	Total Credits	Components			Eligibility Criteria/ Prerequisite	Contents of the course and references may be seen at
			L	T	P		
Programming fundamentals Using C++	DSC-01	4	3	0	1	Class XII Pass	Annexure – III

Table of Contents

1. [Table of Core Courses](#)

Table of Core Courses

Semester	DSC -No.	Title	L	T*	P*	Total credits	Prerequisite
I	DSC 01	Programming fundamentals using C++	3	0	1	4	Pass in Class XII
II	DSC 02	Data structures	3	0	1	4	DSC 01 /a course in C/C++ at plus 2 level/**
III	DSC 03	Computer System Architecture	3	0	1	4	Pass in Class XII
IV	DSC 04	Operating systems	3	0	1	4	DSC 01 /a course in C/C++ at plus 2 level/**
V	DSC 05	Database Management systems	3	0	1	4	DSC 01 /a course in C/C++ at plus 2 level/** DSC 04
VI	DSC 06	Computer Networks	3	0	1	4	DSC 01 /a course in C/C++ at plus 2 level/** DSC 02 DSC 04
VII	DSC 07	Design and	3	1	0	4	DSC 01 /a course in C/C++

		Analysis of Algorithms					at plus 2 level/**, DSC 02
VIII	DSC 08	Information Security	3	0	1	4	DSC 01 DSC 02 , DSC 03 DSC 04 DSC 05 DSC 06 DSC 07

Note: Batch size for Practicals will be (8-10) and Tutorials will be (12-15).

Syllabi of Core Courses

This section gives the detailed syllabus of the core courses. Each course describes the course objective, learning outcomes, the units and the reading material. The reading material has 2 -3 components: main resource(/s), additional text material, and online resources. Main resources are kept to a minimum possible and no more than 3. Additional resources and the online material may be used to enhance the knowledge of the subject.

DSC 01: Programming Fundamentals using C++

Course Objective

This course is designed to develop structured as well as object-oriented programming skills using C++ programming language. The course provides a complete understanding of the object-oriented programming features, namely Encapsulation, Abstraction, Inheritance and Polymorphism along with an in-depth knowledge of C++ constructs.

Course Learning Outcomes

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

1. Explain significance of object oriented paradigm.
2. Solve programming problems using C++.
3. Create classes and reuse them.
4. Implement programs using dynamic memory allocation.
5. Handle external files as well as exceptions.

Syllabus

Unit 1 Introduction to C++: Overview of Procedural and Object-Oriented Programming, Header Files, Compiling and Executing Simple Programs in C++.

Unit 2 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 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 Pointers and References: Static and dynamic memory allocation, Pointer and Reference Variables, Implementing Runtime polymorphism using pointers and references

Unit 5 Exception and File Handling: Using try, catch, throw, throws and finally; Nested try, File I/O Basics, File Operations

References

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.

Additional References

- (i) Herbert Schildt, *C++: The Complete Reference*, 4th Edition, McGraw Hill, 2003.
- (ii) A. B. Forouzan, Richard F. Gilberg, *Computer Science: A Structured Approach using C++*, 2nd edition, Cengage Learning, 2010.

Suggested Practical list

1. Write a program to compute the sum of the first n terms of the following series:

$$S = 1 - 2^n + 3^n - 4^n + \dots$$

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 display the following pattern:

A
BA
CBA
DCBA

The number of rows n, is to be taken from the user.

3. Write a program to compute the factors of a given number using the default argument.
4. Write a menu driven program to perform the following operations on an array:
 - a. Find the minimum, maximum and average of the array elements
 - b. Search an element in the array using linear search
 - c. Search an element in the array using binary search (both iterative and recursive versions)
 - d. Display the address of every element of the array
5. Write a menu driven program to perform the following operations on a string:
 - a. Calculate length of the string (use pointers)
 - b. Check whether the first character of every word in the string is in uppercase or not
 - c. Reverse the string
 - d. Display the address of every character in the string
6. Create a class Triangle. Include overloaded functions for calculating the area of a triangle.
7. Create a template class TwoDim which contains x and y coordinates. Define default constructor, parameterized constructor and void print() function to print the coordinates. Now reuse this class in ThreeDim adding a new dimension as z. Define the constructors and void print() in the subclass. Implement main() to show runtime polymorphism.
8. Copy the contents of one text file to another file and display the number of characters copied.