Teaching Plan

Course: B.Sc.(Physical Sciences)

Semester-I

Subject: DSC01: Introduction to Programming using C++

Learning Objectives

This course is designed to:

• Introduce programming concepts using C++ to students.

• Develop structured as well as object-oriented programming skills using C++ programming language.

• Achieve competence amongst its students to develop correct and efficient C++ programs to solve problems spanning multiple disciplines.

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++.
- Solve problems spanning multiple disciplines using suitable programming constructs in C++.
- Solve problems spanning multiple disciplines using the concepts of object oriented programming in C++.

Week	Topic
Week 1	**Unit – 1** **Introduction to C++** Need and characteristics of Object-Oriented Programming, Structure of a C++ Program (main () function, header files, output, input, comments), compile and execute a simple program
Week 2,3	**Unit – 2** **Data types and Expressions** Keywords, built in data types, variables and constants, naming convention, Input-Output statements, operators and their precedence, expressions, typecasting, library functions

Week 4,5,6	**Unit – 3 (12 hours)** **Control Constructs in C++** Decision making using selection constructs, iteration using looping constructs.
Week 7,8,9	**Unit – 4** **Arrays, Pointers and User Defined Functions** Defining and initializing single and multi-dimensional arrays, user defined functions, passing arguments to functions, returning values from functions, inline functions, default arguments, introduction to pointers **(Assignment-1)**
Week 10,11,12,13,14	**Unit – 5** **Classes and Objects** Need and implementation of abstraction, encapsulation, inheritance and polymorphism, creating classes, objects as function arguments, modifiers and access control, constructors and destructors. **(Test-1) (Assignment-2)**

Practical component

List of Practicals:

1. Write a program to find the largest of n natural numbers.
2. Write a program to find whether a given number is prime or not.
3. Write a program that takes a positive integer n and the produce n lines of output as shown:

```
*
* *
* * *
* * * *
```

(for n = 4)

4. Write a menu driven program for following:
   a. to check whether a given number is odd or even.
   b. display a fibonacci series
   c. compute factorial of a number
5. Write a program to accept a number, reverse it and print the sum of its digits.
6. Write a program using functions to print the series and its sum:
   $1 + 1/2! + 1/3! + \ldots + 1/n!$
7. Write a program to perform the following operations on an input string
   a. Print length of the string
   b. Find frequency of a character in the string
   c. Print whether characters are in uppercase or lowercase
   d. to check whether a given string is palindrome or not.
8. Write a program that will prompt the user for a list of 5 prices. Compute the average of the prices and find out all the prices that are higher than the calculated average.
9. Design a class named Vehicle, having registration number and year as its private members. Define a suitable constructor and a method to print the details of a vehicle. Write a C++ program to test the above class.
10. Inherit a class Car from the Vehicle class defined above. Add model to the Car class. Define a suitable constructor and a method to print the details of a car. Write a C++ program to test inheritance of this class.

Essential/recommended readings

E. Balaguruswamy, Object Oriented Programming with C++,7th edition, McGraw-Hill Education, 2017.

Robert Lafore, Object Oriented Programming in C++, 4th edition, SAMS Publishing, 2008.