B.A (Prog) with Computer Science as Major

CATEGORY-II

DISCIPLINE SPECIFIC CORE COURSE – 1: INTRODUCTION TO 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		
Introduction to Programming using C++	4	3	0	1	Class XII pass	Nil

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++.

SYLLABUS OF DSC - 1

Theory

Unit – 1 **(3 hours)**

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

Unit – 2 **(9 hours)**

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

Unit – 3 **(12 hours)**

Control Constructs in C++

Decision making using selection constructs, iteration using looping constructs.

Unit – 4 **(6 hours)**

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

Unit – 5 **(15 hours)**

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.

Practical **(30 hours)**

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! + … + 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 Readings

- E. Balaguruswamy, Object Oriented Programming with C++,7th edition, McGraw-Hill Education, 2017.
- 2. Robert Lafore, Object Oriented Programming in C++, 4th edition, SAMS Publishing, 2008.

Suggestive Reading

- D.S. Malik, C++ Programming: From Problem Analysis to Program Design, 6th edition, Cengage Learning, 2013.
- (ii) Herbert Schildt, C++: The Complete Reference, 4th Edition, McGraw Hill, 2003.

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

DISCIPLINE SPECIFIC CORE COURSE – 2: PROGRAMMING FUNDAMENTALS USING PYTHON

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 Fundamentals Using Python	4	3	0	1	Class XII pass	Nil

Learning Objectives

This course is designed to:

- Introduce programming concepts using Python to students.

- Develop structured as well as object-oriented programming skills using Python.
- Achieve competence amongst its students to develop correct and efficient Python programs to solve problems spanning multiple disciplines.

Learning Outcomes

On successful completion of this course, a student will be able to:

- Write simple programs using built-in data types of Python.
- Implement arrays and user defined functions in Python.
- Solve problems spanning multiple disciplines using suitable programming constructs in Python.
- Solve problems spanning multiple disciplines using the concepts of object-oriented programming in Python.

SYLLABUS OF DSC - 2

Theory

Unit – 1 **(6 hours)**

Introduction to Python Programming

Problem solving strategies; Structure of a Python program; Syntax and semantics; Python interpreter/shell, indentation; Executing simple programs in Python.

Unit – 2 **(12 hours)**

Creating Python Programs

Identifiers and keywords; literals, numbers, and strings; Operators and expressions; Input and output statements; control structures (conditional statements, loop control statements, break, continue and pass), Errors and exception handling.

Unit – 3 **(9 hours)**

User Defined Functions

Defining functions, passing arguments and returning values, default arguments

Unit – 4 **(18 hours)**

Built-in Data Structures

Strings, Lists, Tuples, Sets, Dictionaries; their built-in functions, operators and operations

Practical **(30 hours)**

List of Practicals:

1. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of three subjects are to be input by the user. Assign grades according to the following criteria:

    Grade A : if Percentage >=80
    Grade B : if Percentage >=60 and Percentage <80

Grade C : if Percentage >=40 and Percentage <60

Grade D : if Percentage <=40

2. WAP to print factors of a given number.
3. WAP to add N natural numbers and display their sum.
4. WAP to print the following conversion table (use looping constructs):

Height (in Feet)	Height (in inches)
5.0 ft	60 inches
5.1 ft	61.2 inches
5.8 ft	69.6 inches
5.9 ft	70.8 inches
6.0 ft	72 inches

5. WAP that takes a positive integer n and the produce n lines of output as shown:

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

(for n =4)

6. Write a menu driven program using user defined functions to print the area of rectangle, square, circle and triangle by accepting suitable input from user.
7. Write a function that calculates factorial of a number n.
8. WAP to print the series and its sum: (use functions)

   1/1! + 1/2! + 1/3! …….1/n!
9. WAP 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
10. WAP to create two lists: one of even numbers and another of odd numbers. The program should demonstrate the various operations and methods on lists.
11. WAP to create a dictionary where keys are numbers between 1 and 5 and the values are the cubes of the keys.
12. WAP to create a tuple t1 = (1,2,5,7,2,4). The program should perform the following:
    a. Print tuple in two lines, line 1 containing the first half of tuple and second line having the second half.
    b. Concatenate tuple t2 = (10,11) with t1.

Essential Readings

- Kamthane, A. N., & Kamthane, A.A. Programming and Problem Solving with Python, McGraw Hill Education, 2017.
- Balaguruswamy E. “Introduction to Computing and Problem Solving using Python”,2nd edition, McGraw Hill Education, 2018.
- Taneja, S., Kumar, N. Python Programming- A modular Approach, Pearson Education India, 2018.

Suggestive Readings

- Guttag, J. V. Introduction to computation and programming using Python, MIT Press, 2018.
- (ii) Downey, A. B. Think Python–How to think like a Computer Scientist 2nd edition. O’Reilly 2015.

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