BSc. Physical Sciences/ Mathematical Sciences with Computer Science as one of the Core disciplines

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 C++	4	3	0	1	Class XII pass	Nil

Learning Objectives

The Learning Objectives of this course are as follows:

- 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 domains.

Learning outcomes

This course will enable the students 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 the concepts of object oriented programming in C++.

SYLLABUS OF DSC

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 **(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 **(9 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 **(12 hours)**

Exception and File Handling

Using try, catch, throw, throws and finally; Nested try, File I/O Basics, File Operations

Practical **(30 hours)**

List of Practicals:

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

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

   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.

Essential Readings

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

Suggestive Readings

- Herbert Schildt, C++: The Complete Reference, 4th Edition, McGraw Hill, 2003.
- A. B. Forouzan, Richard F. Gilberg, 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.

    b. Check whether the first character of every word in the string is in uppercase or not
    c. Reverse 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 co-ordinates. 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.

Essential Readings

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

Suggestive Reading

- Herbert Schildt, C++: The Complete Reference, 4th edition, McGraw Hill, 2003.
- A. B. Forouzan, Richard F. Gilberg, 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.

GENERIC ELECTIVES: PROGRAMMING WITH 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 with Python Code:	4	3	0	1	Class XII pass	NIL

Learning Objectives

The Learning Objectives of this course are as follows:

- 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 in their respective domains.

Learning outcomes

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

- Write simple programs using built-in data structures in Python.
- Implement arrays and user defined functions in Python.
- Solve problems in the respective domain using suitable programming constructs in Python.
- Solve problems in the respective domain using the concepts of object oriented programming in Python.

SYLLABUS OF GE

Theory

Unit - 1 **(6 hours)**

Introduction to Programming

Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs in Python.

Unit - 2 **(15 hours)**

Creating Python Programs

Identifiers and keywords; Literals, numbers, and strings; Operators; Expressions; Input/output statements; Defining functions; Control structures (conditional statements, loop control statements, break, continue and pass, exit function), default arguments.

Unit - 3 **(15 hours)**

Built-in Data Structures

Mutable and immutable objects; Strings, built-in functions for string, string traversal, string operators and operations; Lists creation, traversal, slicing and splitting operations, passing list to a function; Tuples, sets, dictionaries and their operations.

Unit - 4 **(9 hours)**

File and Exception Handling

File handling through libraries; Errors and exception handling.

Practical (30 hours)

List of Practicals:

1. WAP to find the roots of a quadratic equation.
2. WAP to accept a number ‘n’ and
   a. Check if ’n’ is prime
   b. Generate all prime numbers till ‘n’

   c. Generate first ‘n’ prime numbers
   d. This program may be done using functions.
3. WAP to create a pyramid of the character ‘*’ and a reverse pyramid

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

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

4. WAP that accepts a character and performs the following:
   a. print whether the character is a letter or numeric digit or a special character
   b. if the character is a letter, print whether the letter is uppercase or lowercase
   c. if the character is a numeric digit, prints its name in text (e.g., if input is 9, output is NINE)
5. WAP to perform the following operations on a string
   a. Find the frequency of a character in a string.
   b. Replace a character by another character in a string.
   c. Remove the first occurrence of a character from a string.
   d. Remove all occurrences of a character from a string.
6. WAP to swap the first n characters of two strings.
7. Write a function that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string then it should return -1.
8. WAP to create a list of the cubes of only the even integers appearing in the input list (may have elements of other types also) using the following:
   a. 'for' loop
   b. list comprehension
9. WAP to read a file and

  a. Print the total number of characters, words and lines in the file.
  b. Calculate the frequency of each character in the file. Use a variable of dictionary type to maintain the count.
  c. Print the words in reverse order.
  d. Copy even lines of the file to a file named ‘File1’ and odd lines to another file named ‘File2’.

10. Write a function that prints a dictionary where the keys are numbers between 1 and 5 and the values are cubes of the keys.
11. Consider a tuple t1=(1, 2, 5, 7, 9, 2, 4, 6, 8, 10). WAP to perform following operations:
    a. Print half the values of the tuple in one line and the other half in the next line.
    b. Print another tuple whose values are even numbers in the given tuple.
    c. Concatenate a tuple t2=(11,13,15) with t1.
    d. Return maximum and minimum value from this tuple
12. WAP to accept a name from a user. Raise and handle appropriate exception(s) if the text entered by the user contains digits and/or special characters.

Essential Readings

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

Suggestive Reading

- Brown, Martin C., Python: The Complete Reference, 2nd edition, McGraw Hill Education, 2018.
- Guttag, J.V. Introduction to computation and programming using Python, 2nd edition, MIT Press, 2016.

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