

## **GE1a: Programming 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.

### **Syllabus**

**Unit 1 Introduction to C++:** Overview of Procedural and Object-Oriented Programming, Using main() function, 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

### References

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

### Additional References

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

### Suggested Practical list

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

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

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:

1

22

333

4444

55555

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

3. Write a program to compute the factors of a given number.

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 and binary search
5. Write a menu driven program to perform the following operations on a string:
  - a. Calculate length of the string
  - 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.

### **GE1b: Programming with Python**

#### **Course Objective**

This course is designed as the first course that introduces programming concepts using Python to students. The course focuses on the development of Python programming to solve problems of different domains. It also introduces the concept of object-oriented programming.

#### **Course Learning Outcomes**

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

1. Understand the basics of programming language
2. Develop, document, and debug modular Python programs.
3. Apply suitable programming constructs and built-in data structures to solve a

- problem.
4. Use and apply various data objects in Python.
  5. Handle files

## **Syllabus**

**Unit 1 Introduction to Programming:** Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs in Python.

**Unit 2 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 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 File and exception handling:** File handling through libraries; Errors and exception handling.

## **References**

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

## **Additional References**

- (i) Brown, Martin C., *Python: The Complete Reference*, 2<sup>nd</sup> edition, McGraw Hill Education, 2018.
- (ii) Guttag, J.V. *Introduction to computation and programming using Python*, 2<sup>nd</sup> edition, MIT Press, 2016.

## Suggested Practical List

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.