|  |  |
| --- | --- |
| **Introduction to C++ Programming** | |
| **Course Code:PLC144**  **Pre-requisites: Nil**  **Course Coordinator:** | **Credits: 2:0:1**  **Contact Hours:28+14** |

**Course Contents**

**Unit I**

**Introduction to Object Oriented Programming**: Software Evolution, A Look at procedure-oriented programming, Object-oriented programming paradigm, Basic Concepts of Object-oriented programming: Objects, Classes, Data Abstraction and Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message passing, Benefits of OOP, Application of OOP, C++ overview, A simple C++ Program, Structure of C++ Program (Textbook 1)

* Pedagogy/Course delivery tools: Chalk and talk, Power point presentation, Videos
* Link: Basics of C++ - https://www.youtube.com/watch?v=BClS40yzssA

Classes and Objects - <https://www.youtube.com/watch?v=p8ehAjZWjPw>

**Unit II**

**Tokens, Expressions and Control Structures, Array in C++:** Tokens, Keywords, Identifiers and constants, Operators in C++, Scope resolution operator Expressions and their types, Special assignment expressions, Decision making statements: if, if-else, switch, Loops: while loop, do-while loop, for loop, Array: Introduction, initializing single dimension array, Linear search operation on array elements.

**Functions in C++:** Function prototyping, Call by Value, Call by reference, Return by reference, Inline functions, Default arguments, Function overloading (Textbook 1)

* Pedagogy/Course delivery tools: Chalk and talk, Power point presentation, Videos
* Link : Basics of C++ - https://www.youtube.com/watch?v=BClS40yzssA

Functions of C++ - <https://www.youtube.com/watch?v=p8ehAjZWjPw>

**Unit III**

**Classes and Objects:** Specifying a class, Defining member function, Static data members, static member functions, constructors and types- Parameterized and copy, Destructors

**Inheritance & Polymorphism:** Types of Inheritance Defining Derived classes, Single Inheritance, Multiple, Hierarchical Inheritance, Hybrid Inheritance, polymorphism (Textbook 1)

* Pedagogy/Course delivery tools: Chalk and talk, Power point presentation, Videos
* Links: Basics of C++ - https://www.youtube.com/watch?v=BClS40yzssA

Inheritance and polymorphism- <https://www.youtube.com/watch?v=p8ehAjZWjPw>

**Unit IV**

**Streams and formatted input/output:** C++ Stream Classes Hierarchy, Formatted and Unformatted I/O, Process of Input/output, File I/O Programming, Text and Binary Streams, Opening and Closing Files, Text Files, Binary Files, End of File. (Textbook 2)

* Pedagogy/Course delivery tools: Chalk and talk, Power point presentation, Videos
* Links: Basics of C++ - https://www.youtube.com/watch?v=BClS40yzssA

File handling in C++- <https://www.youtube.com/watch?v=p8ehAjZWjPw>

**Unit V**

**Exception Handling:** Introduction, Traditional Error Handling, Need for Exception Handling, Components of Exception, Handling Mechanism, Drawbacks of Exception Handling. (Textbook 2)

* Pedagogy/Course delivery tools: Chalk and talk, Power point presentation, Videos
* Links: Basics of C++ - https://www.youtube.com/watch?v=BClS40yzssA

Exception Handling - <https://www.youtube.com/watch?v=p8ehAjZWjPw>

**Lab Component:**

|  |  |
| --- | --- |
| **SL. No.** | **QUESTIONS** |
|  | 1. Write a C++ program to find the area and circumference of a circle 2. Write a C++ program to find the simple interest 3. Write a C++ program to find the area of a triangle given its sides 4. Write a C++ program to get the name, age and salary of a person and display the same. |
|  | 1. Write a C++ program to find the factorial of a number 2. Write a C++ program to find whether the entered number is palindrome or not. 3. Write a C++ program to find the sum of all the natural numbers from 1 to n. 4. Write a C++ program to find sum of all the elements, maximum and minimum element in an array |
|  | 1. Write a C++ program to search an element in an array using linear search 2. Write a C++ program to find whether an entered number is prime or not using a function (with value, with return type) 3. Write a C++ program to swap 2 values by writing a function that uses call by reference technique. |
|  | 1. Write a C++ program to overload function for computing the area triangle, circle and square 2. Write a C++ program to overload a function to add two numbers of different data types (int, float, double) |
|  | 1. Write a C++ program to perform square of a number using inline function 2. Write a C++ program to create a class called bank\_acct with following data member (cust\_name, cust\_accno, balance) and member functions (read\_details, deposit, withdraw, display balance). Read and display details using array of objects and implement deposit and withdraw using inline. |
|  | 1. Write and execute a C++ Program to display names, roll no’s, and grades of 3 students who have appeared in the examination. Create a class with data members as Name, Roll no and Marks for 3 subjects. Write a method to calculate the grade of a student. 2. Create a C++ class that includes constructors to do the following.  * Create an uninitialized string. * Initialize an object with a string constant at the time of creation. * Create an object and initialize with another object. Also write a function to concatenate two strings. |
|  | 1. Write a C++ program to implement the following inheritance.   Person  Student  Teacher    Marks   * Assume suitable data members and member functions for all the classes. * Display the number of publications for a teacher and percentage marks for a student. |
|  | Write a C++ program to demonstrate multilevel inheritance for the following:  Suppose we have three classes Vehicle, FourWheeler, and Car. The class Vehicle is the base class, the class FourWheeler is derived from it and the class Car is derived from the class FourWheeler. Class Vehicle has a method 'vehicle' that prints 'I am a vehicle', class FourWheeler has a method 'fourWheeler' that prints 'I have four wheels', and class Car has a method 'car' that prints 'I am a car'. So, as this is a multi-level inheritance; we can have access to all the other classes methods from the object of the class Car. We invoke all the methods from a Car object and print the corresponding outputs of the methods.  So, if we invoke the methods in this order, car(), fourWheeler(), and vehicle(), then the output will be  I am a car  I have four wheels  I am a vehicle |
|  | 1. Write a C++ program to create a text file, check file created or not, if created it will write some text into the file and then read the text from the file. 2. Write a C++ program to read the contents from a text file, count and display the number of alphabets present in it. |
|  | 1. Write a program that creates a Calculator class. The class contains two variables of integer type. Design a constructor that accepts two values as parameter and set those values.  * Design four methods named Add (), Subtract (), multiply (), Division ( ) for performing addition, subtraction, multiplication and division of two numbers. * For addition and subtraction, two numbers should be positive. If any negative number is entered then throw an exception in respective methods. So design an exception handler (ArithmeticException) in Add () and Subtract () methods respectively to check whether any number is negative or not. * For division and multiplication two numbers should not be zero. If zero is entered for any number then throw an exception in respective methods. So design an exception handler (ArithmeticException) in multiply () and Division () methods respectively to check whether any number is zero or not. |

**Course outcome:**

At the end of the course the student will be able to:

1. Explain the characteristics of Object oriented programming approach.

2. Develop programs based on decision making statements and arrays.

3. Achieve code reusability and extensibility by means of Inheritance and Polymorphism.

4. Demonstrate C++ functions to perform operations on a file.

5. Illustrate the use of Exception handling feature in C++ for handling errors at runtime.

Suggested Learning Resources

**Text Books:**

1. E Balagurusamy, Object Oriented Programming with C++, Tata McGraw Hill Education Pvt.Ltd, 8th Edition.
2. Bhushan Trivedi, “Programming with ANSI C++”, Oxford Press, Second Edition, 2012.

**Course Assessment and Evaluation:**

|  |  |  |
| --- | --- | --- |
| **Continuous Internal Evaluation (CIE):** 50 Marks | | |
| **Assessment Tool** | **Marks** | **Course outcomes addressed** |
| Internal test-I | 30 | CO1, CO2 |
| Internal test-II | 30 | CO3, CO4, CO5 |
| Average of the two internal tests shall be taken for 30 marks. | | |
| **Other components** | | |
| Lab Component Evaluation | 20 | CO1, CO2, CO3, CO5, CO5 |
| **Semester-End Examination (SEE)** | 100 | CO1, CO2, CO3, CO4, CO5 |