Object Oriented Programming using C++

[Lab Manual]

2015-2016

INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

(IGDTU)

Table of Contents

0. Guidelines

This document titled “OOP using C++ Manual” is based on following guidelines:

1. Each Lab work has some objective to accomplish.
2. It is expected and strongly encouraged that students utilize their Lab hours efficiently.
3. Every effort should be made to complete the lab work in the assigned lab only.
4. Every student work on assigned system only. Any problem in the system must be reported to the lab in charge in advance. Reporting and ensuring the physical condition of the system is the responsibility of the students. No exercise in this regard will be entertained.
5. Any student found indulged in malpractices, disobedience will be debarred from the lab without any warning for a part/whole semester.

SYLLABUS

Paper Title: Object Oriented Programming in C++

UNIT 1 Introduction to Object Oriented Programming: Need for Object Oriented Programming, Comparison of Programming paradigms, Characteristics of Object-Oriented Programming Languages, Structure of a C++ program, Use of cin and cout, Compilation process. C++ Programming Language (Procedural): Tokens, Data Types (Basic, Advanced and Derived), Variables, Reference vs Pointers, Operators (scope resolution, dynamic memory related, type cast), Expressions, Functions (inline function, const arguments, default arguments).

(10Hrs)

UNIT 2 Classes and Objects: Objects, Classes, Encapsulation, Data Abstraction, Role of private and public access specifier, Memory organization of class, Member functions – inline and noninline, static member variables, Friend functions, Class vs Structure, Constructors – default, parameterized, copy and dynamic, Destructors, Assignment operator – deep and shallow copying. Polymorphism: Function overloading, Constructor overloading, Compile time polymorphism, Overloading Rules, Operator Overloading (Unary and Binary) as member function/friend function, Example operators to be overloaded: Arithmetic, Output/Input, Prefix/ Postfix Increment and Decrement, Comparison, Assignment, subscript and function call Operator. **(10Hrs)**

UNIT 3 Inheritance: Inheritance, Types of Inheritance, Use of protected access specifier, Virtual base class, Ambiguity resolution using scope resolution operator and Virtual base class, Overriding inheritance methods, Constructors and Destructor in derived classes. Runtime polymorphism, Pointer to objects, Virtual Functions (concept of virtual table), pure virtual functions, Abstract Class. Managing Input / Output: Concept of streams, console I/O – formatted and unformatted, Manipulators, File I/O – Predefined classes, file opening & closing, file manipulation, read & write operations, sequential and random file access.

(10Hrs)

UNIT 4 Exception Handling: Basic mechanism, Throwing, Catching and Re-throwing. Namespace: Basic concept, role of scope resolution operator and using keyword. Generic Programming: User defined Templates - Class templates with and without multiple parameters and Function templates with and without parameters, Template overloading. Standard Template Library (STL): Introduction, Components – Container, Iterator and Algorithm, Example programs using STL. **(10Hrs)**

Text Books:

1. E. Balaguruswamy, “Object Oriented Programming with C++”, 4th Edition, TMH.

2. Bjarne Stroustrup , “The C++ Programming Language”, Addison Welsley, 3rd Ed.

Reference:

1. D . Parasons, “Object Oriented Programming with C++”, BPB Publication.
2. Steven C. Lawlor, “The Art of Programming Computer Science with C++”, Vikas Publication.
3. Schildt Herbert, “C++: The Complete Reference”, Tata McGraw Hill, 4th Ed., 1999.
4. Behrouz A. Forouan, Richrad F. Gilberg, Computer Science - A Structural Approach Using C++”, Cengage Learning, 2004.
5. Nell Dale, “C++ Plus Data Structure”, Jones and Bartlett, 4th Ed., 2010. 6. Nell Dale, Chips Weens, “Programming and Problem Solving with C++”, Jones and Bartlett , 5th Ed., 2010.

List of Practical

Code: MCA-154 **Subject Name: OOP Using C++ Lab**

Classes, Objects & Functions

1. Write a program to print 'Hello World'.
2. Write a program to calculate factorial of a number entered by user.
3. Write a program to print Fibonacci series up to n.
4. Write a program to read an employee's information from the user and print the same. Employee's information shall include employee ID (*int*), employee name (*string*) and employee salary (*float*).
5. Write a program to take two integer inputs and output their sum, difference, product and division (quotient and remainder) as result based on a third input (operator).
6. Write program(s) to perform following conversions (and vice-versa):-
   a. Temperature in Celsius to Fahrenheit

   b. Height in Centimetres to Feet and Inches

7. A *perfect number* is one whose divisors add up to the number. For example, 6 is a perfect number because 6 = 1 + 2 + 3. Write a program that prints all perfect numbers from 1 till 10000.
8. Write a program to swap two numbers: using call by value, call by reference & pointer
9. Write a program that calculates area and perimeter of the following geometric figures. Your program should use *function overloading* and each function should take as inputs the required arguments (without constraining the user) and return both area and perimeter
   a Square b. Rectangle
10. Write a program with *recursive functions* that perform the following:-
    a. Reverse an input string b. Check if an input string is palindrome or not

11. What are the different ways in which objects of a class can be created? Explain with an Illustrative program for each.
12. Write a program to illustrate the use of a) inline function b) function with default argument
13. Write a program to illustrates :
    a. use of friend function b. Member function as friend function

14. Account database is maintaining details of customer. Design suitable *class* and write member functions which provide the following functions:
    a. Insert details of customer b. Search an existing customer w.r.t. account no.
    c. Modify the details of an existing customer

Operator Overloading

15. Write a program which provides concrete representation for the concept of complex number. Using *operator overloading*, perform the operations of addition and multiplication of two given complex numbers. Implement operator overloading
16. Extend the concept of complex number by providing provisions for unary operators + and – which increments and decrements both real and imaginary part of complex numbers by 1, respectively. Implement operator overloading
17. Write a program to overload the following operators:
    a. Subscript operator
    b. Function call operator
    c. Assignment operator

Inheritance

18. Write a C++ program that represents an Vehicle-Car and Vehicle-Motorcycle relationship C++ program should provide following operations.
    a. Add a vehicle (car/motorcycle).

    b. Display the vehicles (including both cars and motorcycles) with respect to their registration ID (alpha-numeric).

19. Design a single C++ program illustrating the following concept of inheritance:-
    i. Single inheritance
    ii. Multiple Inheritance

    ii. Multilevel Inheritance

20. Design a single C++ program illustrating the following concept of inheritance:-
    i. Public derivation

    ii. Private derivation

    iii. Protected derivation

21. Write a C++ program that illustrates the concept of virtual base class
22. Write a C++ program that illustrates the use of pure virtual function. Create an abstract class *shape* which has *area ( )* and *perimeter ( )* as its member functions. These functions are overridden by inherited classes namely *rectangle, square, triangle* and *circle.*

Input/output

23. Write a C++ program user to reads the contents of a given text file and writes it to the standard output console.
24. Copies the contents of a source file into the destination file
25. Write a C++ program user to take the input from standard console and writes it to text file.
26. Write a program to check a number entered by user is even or not. If number is even then store in a file (even.txt) else store in a file (odd.txt).

Exception handling

27. Write a program to handle exception different try of exceptions in C++
28. Division by zero is an exception. Write a C++ program to detect such an exception and *handle it by displaying an appropriate message and exit from the program.*
29. *Write a C++ program to illustrate the concept of re-throwing an exception.*
30. *Write a program to handle an exception thrown by new.*

Namespace

31. Write a program to illustrate the use of Functions defined within a user defined namespace
32. Write a program to create within a user defined namespace and create the objects of class using scope resolution and using directive
33. Write a program to show the concept of nested namespace

Template

34. Write a C++ program that represents a collection of elements of pre-defined using template class. The template class should provide following features:-
    a. Adding a new element.

    b. Deleting an existing element.

    c. Displaying the collection of elements.

35. Write a C++ program to illustrate the concept of template function by writing template function which performs the following tasks:-
    a. Addition of two given type

    b. Subtraction of one type with another type