Object Oriented Programming

Course Title: Object Oriented Programming
Course No: BIT153
Nature of the Course: Theory + Lab
Semester: II

Full Marks: 60 + 20 + 20
Pass Marks: 24 + 8 + 8
Credit Hrs: 3

Course Description:
The course familiarizes students with the concepts of object oriented programming using C++.

Course Objective:
The main objective of this course is to understand the basics of object oriented programming. This course covers the C++ concepts such as objects, class, operator overloading, inheritance and polymorphism, file I/O, exception handling and templates.

Detailed Syllabus

Unit	Teaching Methodology	Teaching Hours
Unit 1: Introduction to Object Oriented Programming (3 Hrs.)		
1.1 Overview of structured and object oriented programming approach, Characteristics of object oriented languages: Object, Class, Data Abstraction, Encapsulation, Inheritance, Polymorphism, Dynamic Binding, and Message Passing.	Lecture	3
Unit 2: Basics of C++ programming (5 Hrs.)		
2.1 C++ Program Structure, Character Set and Tokens, Data Types, Type Conversion, Preprocessor Directives, Namespace, Input/Output Streams (cin and cout object, Cascading of IO Operators) Manipulators (endl, setw, and setprecision), Dynamic Memory Allocation with new and delete, Review of Control Statements. 2.2 Functions: Function Overloading, Inline Functions, Default Argument, Pass by Reference, Return by Reference, Scope and Storage Class. 2.3 Pointers: Review of Pointer variables declaration & initialization, Operators in pointers/Pointer Arithmetic, Pointers and Arrays, Pointer and Function.	Lecture + Laboratory Work	2 2 1
Unit 3: Class and Objects (10 Hrs.)		
3.1 Structures and Classes, Class and Object, Accessing members of class, Memory allocation	Lecture + Laboratory Work	3

for Objects, Defining Member functions of the class (inside and outside the class). 3.2 Initialization of class object (Constructor), Destructor, Types of Constructor (Default Constructor, Parameterized Constructor, Copy Constructor, The Default Copy Constructor). 3.3 Objects as Function Arguments, Returning Objects from Functions, Static members (Static data members and static member functions). 3.4 Friend Function and Friend Class, Copy Initialization, This Pointer.		3 2 2
Unit 4: Operator Overloading (7 Hrs.)		
4.1 Fundamental of operator overloading, Restriction on operator overloading (operators that cannot be overloaded), Rules for Overloading Operators. Operator functions as a class members. 4.2 Overloading unary Operators: Overloading Pre-increment, Post-increment, and Negation Operator. 4.3 Overloading Binary Operators: Overloading Plus Operator, Overloading Comparison Operator, String Concatenation using operator overloading. 4.4 Data Conversion: Basic to User-defined/class, User-defined/class to Basic, User-defined/class to User-defined/class.	Lecture + Laboratory Work	1 2 2 2
Unit 5: Inheritance (7 Hrs.)		
5.1 Introduction to inheritance, Derived Class and Base Class, Access Specifiers (private, protected, and public). 5.2 Types/Forms of inheritance (Single, Multiple, Hierarchical, Multilevel, Hybrid), Public, Protected and Private Inheritance. 5.3 Constructor and Destructor in derived classes (Order of execution of Constructors and Destructors). 5.4 Aggregation, Aggregation vs. Inheritance, Ambiguity (Ambiguity in Multiple and Multipath Inheritance), Virtual Base Class.	Lecture + Laboratory Work	1.5 2.5 1 2
Unit 6: Virtual Function, Polymorphism, and other C++ Features (3 Hrs.)		
6.1 Concept of Static and Dynamic Polymorphism, Pointers to Base Class, Concept of Virtual functions, Implementation of Dynamic Polymorphism, Abstract class,	Lecture + Laboratory Work	3

Concrete Class, Pure Virtual Functions, Virtual Destructors, Roles of Polymorphism.		
Unit 7: Function Templates and Exception Handling (4 Hrs.)		
7.1 Templates: Introduction, Function Templates, Function templates with multiple arguments, Class Templates, Templates and Inheritance, Rules for Using Templates. 7.2 Exception Handling: Introduction, Mechanism of Exception Handling, Try, catch and throw statements, Multiple Catch Statement, Catching All Exceptions, Uses of exception handling.	Lecture + Laboratory Work	2 2
Unit 8: File and Streams (6 Hrs.)		
8.1 Stream Class Hierarchy, File pointers (Read and Write Pointer), Reading and Writing Text Files, Detecting end of file, Reading and Writing Binary Files 8.2 Reading and Writing Objects, Random File Access.	Lecture + Laboratory Work	4 2

Text Book

1. Robert Lafore, Object Oriented Programming in C++, Fourth Edition, SAMS publications.

Reference Books

2. Deitel and Deitel, C++ How to Program, Third Edition, Pearson Publication.
3. Joyce Farrell, Object-oriented programming using C++, Fourth Edition, Cengage Learning.
4. Herbert Schildt, C++ The Complete Reference, Fourth Edition, Tata McGraw Hill
5. Publication.
6. E. Balagurusamy, "Object Oriented Programming With C++", 7th Edition, Mc Graw Hill India.
7. A. S. Saud, "C++ Programming with Object Oriented Approach", 2nd Edition, KEC Publication and Distributor.

Model Question

Course Title: Object Oriented Programming
Course No: BIT153
Time: 3 Hrs.

Full Marks: 60
Pass Marks: 24
Semester: II

Section A

Attempt ANY TWO {10 ×2=20}

1. Write a program to create a class Customer with data members customer Id, Name, and age. Put member functions to read and display values of data members. Also define constructors to initialize data members with default value. Finally create object to read and display data of 10 customers.
2. Create a class Employee with private data members Eid, Ename, and Salary. Include public member functions to read and display value of data members. Derive a class names Typist from above class. The class should contain a private data member typing_speed and public member functions to read and display values of data members. Finally create two objects of typist class and read and display their values.
3. What are the different ways of achieving static polymorphism? Discuss overloading of pre-increment and post-increment operators with suitable examples.

Section B

Attempt ANY EIGHT {5 ×8=40}

4. What is meant by default argument? Explain with suitable example.
5. What are the uses of this pointer? Discuss its concept with suitable example.
6. What is destructor? Discuss its characteristics with suitable example.
7. Write a program to convert floating point value to object of distance class with data members feet and inch.
8. How late binding differs from early binding? How can you achieve dynamic polymorphism? Explain with example.
9. What is meant by ambiguity? Explain ambiguity in multiple inheritance and its solution with example.
10. Write a program to handle “array index out of range” exception.
11. Why concept of templates is important? Explain function templates with example.
12. Write a program to read text from keyboard and write it to a file named “test.txt”. Again read the content of file and display it in the monitor.

Lab Manual

Lab Sheet#	Chapter
Lab Sheet#1 1. Write programs to demonstrate insertion & extraction operators, manipulators, and dynamic memory management. 2. Write a programs to demonstrate default arguments, inline function, and function overloading.	Chapter 2
Lab Sheet#2 1. Write programs to demonstrate class, object, array of objects, and pointer objects. 2. Write programs to demonstrate constructor and destructor. 3. Write programs to demonstrate static members, this pointer, friend function, and friend class.	Chapter 3
Lab Sheet#3 1. Write programs to demonstrate unary operator overloading. 2. Write programs to demonstrate binary operator overloading. 3. Write programs to demonstrate type conversions.	Chapter 4
Lab Sheet#4 1. Write programs to demonstrate access specifiers and different types of inheritances. 2. Write programs to demonstrate constructor and destructors in inheritance. 3. Write programs to demonstrate public and private derivation. 4. Write programs to demonstrate ambiguities and inheritance and their solution.	Chapter 5
Lab Sheet#5 1. Write programs to demonstrate run time polymorphism.	Chapter 6
Lab Sheet#6 1. Write programs to demonstrate function and class templates. 2. Write programs to demonstrate try, catch, and throw statements. 3. Write programs to demonstrate multiple catch blocks.	Chapter 7
Lab Sheet#7 1. Write programs to demonstrate reading and writing text and binary files. 2. Write programs to demonstrate reading and writing objects. 3. Write programs to demonstrate random access in files.	Chapter 8