## **Paper IV -Object Oriented Programming**

**Course Description:** This course discusses the object-oriented model in programming. Students taking this course would have already taken a course on programming. This will help build good quality software using object-oriented techniques

**Aims and Objectives:** When students complete this course, they will be able to:

* Understand the importance of OOP as compared to Structured Programming.
* Understand the basic constructs of a C++ program.
* Define a problem and implement its solution using classes and objects.
* Effectively use and implement OOP analysis and design.
* Understand object-oriented programming features in C++.
* Understand object-oriented concepts and how they are supported by C++.
* Gain some practical experience of C++,

**Course Contents:** Object Oriented Programming (OOP) and its Significance as a Modeling Technique. Comparison of Structured Programming and OOP, Classes and Objects in C++, Abstraction, New User Defined Data Types, Creating Objects from Classes, Accessing Member, Access Specifiers, Member Functions, Defining Member Functions, Constructors and Properties, Default Constructor, Constructor Overloading, Copy Constructor, Deep Copy, Shallow Copy, Destructors, “this” Pointer, Constant Member Function, Static Variables, Accessing Static Data Member, Static Member Function, Comparison of Global Variables and Static Variable, Arrays of Objects, Pointer to Objects Templates, Function Templates, Class Templates, Data Encapsulation and Abstraction, Importance of Data Encapsulation and Abstraction, Correctly Using the Access Modifiers, Friend Functions, Composition, Aggregation, Operator Overloading, Overloading Assignment Operator, Friend Function and Operator Overloading, Unary Operators Overloading, Inheritance and Importance, Inheritance in C++, Comparison of Overloading and Overriding, Hierarchy of Inheritance, Types of Inheritance, Private Inheritance, Protected Inheritance Multiple Inheritance Problem in Multiple Inheritance, Polymorphism and Importance, Virtual Functions, Static Binding, Dynamic Binding, Abstract Classes and Concrete Classes, Virtual Destructors, Virtual Functions and Pure Virtual Functions, Virtual Functions Usage, Dynamic Dispatch, Namespaces and Using Namespaces, Memory Management and Importance, Memory Areas(Heap, Stack), Use of new Operator, malloc() and calloc() Functions Calls.

**Recommended Books**

1. Deitel, H. M., Deitel, P. J., & Nieto, T. R. (2016). *C++ How to Program* (10th ed.).Prentice Hall.
2. Laurence, P. (2017). *C++: The Ultimate Crash Course to Learning the Basics of C++* (Latest ed.). Prentice Hall.

**Bibliography**

1. Lafore, R. (2001). *Object-Oriented Programming in C++* (4th ed.).Pearson Education.
2. Kanetkar, Y. (2003). *Let us C++* (2nd ed.).PBP Publication.