Code No.: MCA 104	**L T C**
Paper: Object Oriented Programming in C++	**3 1 4**

INSTRUCTIONS TO PAPER SETTERS:

1. Question No. 1 should be compulsory and cover the entire syllabus. There should be 10 questions of short answer type of 2 marks each, having at least 2 questions from each unit.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions to evaluate analytical/technical skills of candidate. However, student may be asked to attempt only 1 question from each unit. Each question should be 10 marks including subparts, if any.

OBJECTIVES: *After covering the core C++ in about 25 lectures the course shall aim to aquaint the students about advanced features of the language the following features are as suggested guideline for the teacher.*

- *Copy construtor, Deep and shallow coping, assignment operator and destrutors, when the programmer must implement these*
- *Static and late binding. Run time and comile time polymorphism, virtual functions and VTABLE*
- *Implementing ADT with C++ classes. Stacks Queues and Linked Lists as cases*
- *Implementing Trees and Graph and all comparison based sorting algorithms*
- *Function objects and call backs*
- *Templates and Generics Stack Queses should be implemented in the practicals*
- *Extensive coverage of all the three components of STL namely containers, iterators and algorithms throgh suitable pratical caseletts*
- *Final case study could be an application making extensive handling files.streams classes*

PRE-REQUISITES:

- Data Structure Concept
- Real Programming Experience with C Language
- UNIT-III of MCA-102 should be finished before start of Unit-IV of this paper

UNIT – I

OOP Paradigm: Comparison of Programming paradigms, Characteristics of Object-Oriented Programming Languages, Object-based programming languages **C++**: Brief History of C++, Structure of a C++ program, Difference between C and C++ - cin, cout, new, delete operators, ANSI/ISO Standard C++, Comments, Working with Variables and const Qualifiers. Enumeration, Arrays and Pointer.

Implementing oops concepts in c++ Objects, Classes, Encapsulation, Data Abstraction, Inheritance, Polymorphism, Dynamic Binding, Message Passing, Default Parameter Value, Using Reference variables with Functions. **[No. of Hrs: 09]**

UNIT – II

Abstract data types, Class Component, Object & Class, Constructors Default and Copy Constructor, Assignment operator deep and shallow coping, Access modifiers – private, public and protected. Implementing Class Functions within Class declaration or outside the Class declaration. instantiation of objects, Scope resolution operator, Working with Friend Functions, Using Static Class members. Understanding Compile Time Polymorphism function overloading Rules of Operator Overloading (Unary and Binary) as member function/friend function, Implementation of operator overloading of Arithmetic Operators, Overloading Output/Input, Prefix/ Postfix Increment and decrement Operators, Overloading comparison operators, Assignment, subscript and function call Operator , concepts of namespaces. **[No. of Hrs. 10]**

UNIT – III

Inheritance: Inheritance, Types of Inheritance, Abstract Classes, Ambiguity resolution using scope resolution operator and Virtual base class, Aggregation, composition vs classification hierarchies, Overriding inheritance methods, Constructors and Destructor in derived classes. Multiple Inheritance.

Polymorphism: Polymorphism, Type of Polymorphism – compile time and runtime, Understanding Dynamic polymorphism: Pointer to objects, Virtual Functions (concept of VTABLE) , pure virtual functions, Abstract Class.

Advanced Input/Output, Exception Handling and Manipulating strings, Using istream / ostream member functions, Using Manipulators, Creating Manipulator Functions, Understanding Implementation of Files, Writing and Reading Objects. Understanding of working and implementation of Exception Handling. **[No. of Hrs: 11]**

UNIT – IV

Generic Programming: and mastering STL Understanding Generic Functions with implementation of searching sorting algorithm. Overloading of Function Templates. Understanding Class Templates using Implementation of Generic stack, linked lists: singly and doubly linked lists, Binary Search Tree basic operations. Understanding Inheritance with Generic Class.

Standard Template Library:– Understanding Components of Standard Template Library, Working of Containers, Algorithms, Iteraters and Other STL Elements. Implementation of Sequence and Associative containers for different Algorithms using their Iterator. Understanding of Algorithms Requiring Operations on the element using function objects. Implementing graph algorithm dfs, bfs, minimum spanning tree , dijkastra etc using STL **[No. of Hrs: 12]**

TEXT BOOKS:

1. A. R. Venugopal, Rajkumar, and T. Ravishanker "Mastering C++", TMH, 1997.
2. S. B. Lippman and J. Lajoie, "C++ Primer", 3rd Edition, Addison Wesley, 2000.
3. Bruce Eckel, "Thinking in C++", President, Mindview Inc., Prentice Hall, 2nd Ed.