Course Syllabi: UEI623 Object Oriented Programming and Applications (L : T : P :: 2 : 1 : 2)

1. **Course number and name:** UEI623; Object Oriented Programming and Applications

2. **Credits and contact hours:** Credits: 3.5; Hours: 5

3. **Text book, title, author, and year**

   - *Aho, A.V., Hopcraft, J.E . and Ullman, J.D., Data Structures and Algorithms, Addison Wesley (2004).*
   - *Langsam, Y., Augenstein, M. J. and Tenenbaum, A. M., Data Structures with C and C++, Prentice–Hall of India Private Limited (2000).*
   - *Tremblay, J.P. and Sorenson, P.G. Data StructuresOrganization and Architecture – Designing for Performance , Prentice–Hall of India Private Limited(2004).*
   - *Pattric, N., The C++ Complete Reference, McGraw–Hill (1982).*
   - *Larman, Craig, Applying UML and Patterns, Pearson Education (2005).*

   a. Other supplemental materials
      - Nil

4. **Specific course information**

   a. Brief description of the content of the course (catalog description)

   Introduction: Need of Object Oriented Programming (OOP), Advantages of OOP, Characteristic of Object Oriented Languages, Basic concepts of C, Basic concepts of Turbo C++/ Visual C++.

   C++ Programming: Data Types, Operators and Statements, Writing a Program in C++: Declaration of variables, Statements, Simple C++ programs, Features of iostream.h, Manipulator functions, Input and Output Stream Flags, Control Statements, Functions and Program Structures, Arrays, Pointers, Enumerated data types.

   UML: Basic concepts of Unified Modeling Language, Class diagrams, State machine diagrams and flow diagrams.

   Classes and Objects: Introduction, Structures and Classes, Declaration of Class, Member functions, defining the Object of a Class, Accessing a member of class, Array of class objects, Unions and classes, Nested class.

   More on Classes: Constructors, Destructors, Inline member functions, Static class members, Friend functions, Dynamic memory allocations, this pointer.

   Inheritance: Introduction, Single inheritance, Types of base classes, Types of derivation, Ambiguity in single heritance, Array of class objects and single heritance, Multiple heritance, Container classes, Member access control.

   Overloading: Function overloading, Operator overloading, Overloading of binary operators, overloading of unary operators.

   Polymorphism: Introduction, Early binding, Polymorphism with pointers, Virtual functions, Late binding, Pure virtual functions, Abstract base classes, Constructors under inheritance, Destructors under inheritance, Virtual Destructors, Virtual base classes.

   Templates and Exception Handling: Function Template, Class template, Exception handling.

   Data File Operations: Opening and Closing of Files, Stream state member functions, Reading/Writing a character from a file, Binary file operations, Classes and file operations, Array of class objects and file operations, Nested classes and file operations, Random access file processing.

Application of OOP in Data Structure: Data Structures - Sparse matrices, Stacks, Queues, recursion, Applications of recursion, linked lists (singly linked, doubly linked and circular linked lists). General graph features, Trees, Binary trees and their applications, Traversal algorithms for binary trees, Threaded binary tree, breadth first search, Depth first search and heuristic search algorithms, B trees and Game trees, Memory mapping.

Laboratory Work: Introduction to Turbo C++/Visual C++, Simple class programs, Structured programming based on UML, Accessing function among classes, Time display, Use of constructor and destructor, String concatenation using operator overloading, Inheritance in currency conversion, Software designing based on OOPS, Applications based on various concepts of OOP.

5. **Specific goals for the course**

After the completion of the course, the students will be able to:

- Demonstrate the concept of OOPS.
- Have a practical hand on programming concepts by the use of conditional statements, pointers, arrays classes, polymorphism etc.
- Demonstrate the inheritance concept, use of OOPS in data structures.

6. **Brief list of topics to be covered**

- C++ Programming
- UML
- Classes and Objects
- Inheritance
- Overloading
- Polymorphism
- Templates and Exception Handling