

2/4 B.Tech FIRST SEMESTER

**IT3T4 OBJECT ORIENTED PROGRAMMING through C++ Credits: 4
(Common to CSE/IT)**

Lecture: 4 Periods/week

Internal assessment: 30 marks

Tutorial: 1 Period /week

Semester end examination: 70 marks

Objectives:

- To provide different programming paradigms, need for object oriented programming.
- To acquire the knowledge in developing C++ programs.
- To get familiar with the graphics classes & vectors, templates and exception handling.
- To acquire the knowledge in Containers and Iterators.

Outcomes:

Students will be able to

- Understand general principles of computer languages and how these principles are implemented in C++ programming language.
- Distinguish OO programming from declarative and procedural ones.
- Define secure classes, inner classes, objects and message passing.
- Understand about Vectors and Graphics.
- Understand and implement inheritance, Polymorphism, exception handling programs.
- Develop well-structured and documented OO applications with Template Programming, Containers and Iterators.

Syllabus:

UNIT – I

INTRODUCTION:

Programs, The Classic First Program, Compilation, Linking & Programming Environments, Objects, Types, & Values, Computation, Objectives & Tools, Expressions, Statements, Functions, Vector & Language Features. Errors and Exceptions.

UNIT - II

WRITING A PROGRAM:

A Problem, Thinking about the Problem, Calculator Example, Grammars, Turning a Grammar into Code, First Version, Second Version, Token Streams, and Program Structure.

COMPLETING A PROGRAM:

Introduction, Input & Output, Error Handling, Negative Numbers, Remainder, Cleaning of the Code, Recovering from Errors and Variables.

UNIT – III

TECHNICALITIES (FUNCTIONS, Etc.):

Declarations and Definitions, Header Files, Scope, Function Call and Return, Order of Evaluation and Namespaces.

UNIT – IV

CLASSES:

User – Defined types, Classes and Members, Interface and Implementation, Evolving a Class, Enumerations, Operator Overloading, Class Interfaces and the Date class.

UNIT - V

INPUT & OUTPUT STREAMS:

Input and Output, I/O Stream Model, Files, Opening a File, Reading and Writing a File, I/O Error Handling, Reading a Single Value, User-Defined Output Operators, User-Defined Input Operators, A Standard Input Loop and Reading a Structured File.

CUSTOMISING INPUT AND OUTPUT:

Regularity and Irregularity, Output Formatting, File Opening and Positioning, String Streams, Line Oriented Input, Character Classification, Non Standard Separators.

UNIT – VI

DISPLAY MODEL:

Using a GUI Library, Coordinates, Shapes, Using Shape Primitives.

GRAPHICS CLASSES, GRAPHICS CLASS DESIGN:

Design Principles, Shape, Base and Derived Classes (Inheritance & Polymorphism) and Benefits of Object Oriented Programming.

UNIT - VII

VECTOR & FREE STORE:

Introduction, Vector Basics, Memory, Addresses and Pointers, Free Store and Pointers, Destructors, Access to Elements, Pointers to Class Objects, Messing with Types, Pointers and References, The *this* Pointer.

VECTORS AND ARRAYS:

Introduction, Copying, Essential Operations, Access to Vector Elements, Arrays and Examples.

UNIT –VIII

VECTOR, TEMPLATES, AND EXCEPTIONS:

The Problems, Changing Size, Templates, Range Checking and Exceptions, Resources and Exceptions.

CONTAINERS AND ITERATORS:

Storing and Processing Data, STL Ideals, Sequences and Iterators, Linked Lists, Generalizing Vector, A Simple Text Editor, Vector, List and String, Adapting Vector to the STL, Adapting Built-in Arrays to the STL and Container Overview.

Text Book:

Programming: Principles and Practice Using C++, 1st Edition (2009), Bjarne Stroustrup, Addison-Wesley (Pearson Education).

Reference Books:

C++ for Programmers, Paul J. Deitel, Harvey M. Deitel, Pearson Education, 2009