	L	T	P	C

I Year - II Semester

L	T	P	C
4	0	0	3

OBJECT-ORIENTED PROGRAMMING THROUGH C++

OBJECTIVES:
- This course is designed to provide a comprehensive study of the C programming language. It stresses the strengths of C, which provide students with the means of writing efficient, maintainable and portable code. The nature of C language is emphasized in the wide variety of examples and applications. To learn and acquire art of computer programming. To know about some popular programming languages and how to choose
- Programming language for solving a problem.

UNIT-I: Introduction to C++
Difference between C and C++- Evolution of C++- The Object Oriented Technology- Disadvantage of Conventional Programming- Key Concepts of Object Oriented Programming- Advantage of OOP- Object Oriented Language.

UNIT-II: Classes and Objects &Constructors and Destructor
Classes in C++-Declaring Objects- Access Specifiers and their Scope- Defining Member Function- Overloading Member Function- Nested class, Constructors and Destructors, Introduction- Constructors and Destructor- Characteristics of Constructor and Destructor-Application with Constructor- Constructor with Arguments (parameterized Constructor-Destructors- Anonymous Objects.

UNIT-III: Operator Overloading and Type Conversion & Inheritance
The Keyword Operator- Overloading Unary Operator- Operator Return Type- Overloading Assignment Operator (=)- Rules for Overloading Operators, Inheritance, Reusability- Types of Inheritance- Virtual Base Classes- Object as a Class Member- Abstract Classes- Advantages of Inheritance-Disadvantages of Inheritance,

UNIT-IV: Pointers & Binding Polymorphisms and Virtual Functions
Pointer, Features of Pointers- Pointer Declaration- Pointer to Class- Pointer Object- The this Pointer- Pointer to Derived Classes and Base Class, Binding Polymorphisms and Virtual Functions, Introduction- Binding in C++- Virtual Functions- Rules for Virtual Function- Virtual Destructor.

UNIT-V: Generic Programming with Templates & Exception Handling
Generic Programming with Templates, Need for Templates- Definition of class Templates- Normal Function Templates- Over Loading of Template Function-Bubble Sort Using Function Templates- Difference Between Templates and Macros- Linked Lists with Templates, Exception Handling- Principles of Exception Handling- The Keywords try throw and catch- Multiple Catch Statements – Specifying Exceptions.

UNIT-VI: Overview of Standard Template Library
Overview of Standard Template Library- STL Programming Model- Containers- Sequence Containers- Associative Containers- Algorithms- Iterators- Vectors- Lists- Maps.

OUTCOMES:
- Understand the basic terminology used in computer programming
- Write, compile and debug programs in C language. Use different data types in a computer program.
- Design programs involving decision structures, loops and functions.
- Explain the difference between call by value and call by reference

Text Books:
1. A First Book of C++, Gary Bronson, Cengage Learing.
2. The Complete Reference C++, Herbert Schildt, TMH.
3. Programming in C++, Ashok N Kamathane, Pearson 2nd Edition.

Reference Books:
1. Object Oriented Programming C++, Joyce Farrell, Cengage.
2. C++ Programming: from problem analysis to program design, DS Malik, Cengage Learing.

	L	T	P	C
I Year - II Semester	**0**	**0**	**3**	**2**

OBJECT-ORIENTED PROGRAMMING LAB

OBJECTIVE:

- To strengthen their problem solving ability by applying the characteristics of an object-oriented approach.

- To introduce object oriented concepts in C++ and Java.

Programming:

Exercise – 1 (Basics)
Write a Simple Program on printing "Hello World" and "Hello Name" where name is the input from the user
  a) Convert any two programs that are written in C into C++
  b) Write a description of using g++ (150 Words)

Exercise – 2 (Expressions Control Flow)
 a) Write a Program that computes the simple interest and compound interest payable on principalamount(inRs.)ofloanborrowedbythecustomerfromabankforagiverperiodof time (in years) at specific rate of interest. Further determine whether the bank will benefit by charging simple interest or compound interest.

 b) WriteaProgramtocalculatethefareforthepassengerstravelinginabus. WhenaPassenger enters the bus, the conductor asks "What distance will you travel?" On knowing distance from passenger (as an approximate integer), the conductor mentions the fare to the passenger according to following criteria.

Exercise – 3 (Variables, Scope, Allocation)

  a) Write a program to implement call by value and call by reference using reference variable.

  b) Write a program to illustrate scope resolution, new and delete Operators. (Dyanamic Memory Allocation)

  c) Write a program to illustrate Storage classes

  d) Write a program to illustrate Enumerations

Exercises –4 (Functions)

Write a program illustrating Inline Functions

    a) Write a program illustrate function overloading. Write 2 overloading functions for power.

    b) Write a program illustrate the use of default arguments for simple interest function.

Exercise -5 (Functions –Exercise   Continued)

a) Write a program to illustrate function overloading. Write 2 overloading functions for adding two numbers

b) Write a program illustrate function template for power of a number.

c) Write a program to illustrate function template for swapping of two numbers.

Exercise -6 (Classes Objects)

Create a Distance class with:

- feet and inches as data members
- member function to input distance
- member function to output distance
- member function to add two distance objects

  a). Write a main function to create objects of DISTANCE class. Input two distances and output the sum.

  b). Write a C++ Program to illustrate the use of Constructors and Destructors (use the above program.)

c) Write a program for illustrating function overloading in adding the distance between objects (use the above problem)

d). Write a C++ program demonstrating a BankAccount with necessary methods and variables

  Exercise – 7 (Access)

  Write a program for illustratingAccess Specifiers public, private, protected

a) Write a program implementing Friend Function

b) Write a program to illustrate this pointer

c) Write a Program to illustrate pointer to a class

d)

  Exercise -8 (Operator Overloading)

  a). Write a program to Overload Unary, and Binary Operators as Member Function, and Non Member Function.

  i. Unary operator as member function

ii. Binary operator as nonmember function

b). Write a c ++ program to implement the overloading assignment = operator

c).Write a case study on Overloading Operators and Overloading Functions (150 Words)

Exercise -9 (Inheritance)
 a) Write C++ Programs and incorporating various forms of Inheritance
   i) Single Inheritance
   ii) Hierarchical Inheritance
   iii) Multiple Inheritances
   iv) Multi-level inheritance
   v) Hybrid inheritance
b) Write a program to show Virtual Base Class

c) Write a case study on using virtual classes (150 Words)

Exercise-10 (Inheritance –Continued)
a) Write a Program in C++ to illustrate the order of execution of constructors and destructors in inheritance
b) Write a Program to *show* how *constructors* are invoked in *derived class*

Exercise -11 (Polymorphism)
   a) Write a program to illustrate runtime polymorphism
   b) Write a program to illustrate this pointer
   c) Write a program illustrates pure virtual function and calculate the area of different shapes by using abstract class.
   d) Write a case study on virtual functions (150 Words)

Exercise -12(Templates)
   a) Write a C++ Program to illustrate template class

   b) Write a Program to illustrate class templates with multiple parameters

   c) Write a Program to illustrate member function templates

Exercise -13 (Exception Handling)
a).Write a Program for Exception Handling Divide by zero

 b). Write a Program to rethrow an Exception

Exercise -14 (STL)
a) Write a Program to implement List and List Operations

b) Write a Program to implementVector andVector Operations

Exercise -15 (STLContinued)
a) Write a Program to implement Deque and Deque Operations

b) Write a Program to implement Map and Map Operations

OUTCOMES:

- Explain what constitutes an object-oriented approach to programming and identify potential benefits of object-oriented programming over other approaches.

- Apply an object-oriented approach to developing applications of varying complexities