

**Subject Code: 01CE1303**

**Subject Name: Object Oriented Design and Programming**

**B.Tech. Year – II**

**Objective:** The objective of the course is to enable graduates to identify and practice the object-oriented programming concepts and techniques. This course covers practical usage of C++ classes and class libraries, modify existing C++ classes, and develop C++ classes for simple applications.

**Credits Earned:** 4 Credits

**Course Outcomes:** After completion of this course, students will be able to

- Identify the potential benefits of object-oriented programming features and compare them with structure-oriented programming features. (Understand)
- Apply various object-oriented Features and Concepts to designing programs and to solve various computing problems using C++ language. (Apply)
- Analyze programs based on exception handling and using advanced features like STL for faster development. (Analysis)
- Apply Different concepts of object-oriented programming to develop real-world applications. (Apply)

**Pre-requisite of course:** Programming Fundamentals

**Teaching and Examination Scheme**

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial/ Practical Marks		Total Marks
Theory	Tutorial	Practical		ESE (E)	Mid Sem (M)	Internal (I)	Viva (V)	Term work (TW)	
3	0	2	4	50	30	20	25	25	150

**Contents:**

<b>Unit</b>	<b>Topics</b>	<b>Contact Hours</b>
1	<b>Introduction :</b> Introduction to Object Oriented Programming, Procedural Vs. Object Oriented Programming, Principles of OOP, Access Modifiers, Basics of a Typical C++ Environment, Pre-processors Directives, C++ Program structure. Header Files and Namespaces, library files.	6
2	<b>Programming Basics :</b> Output using cout. Directives. Input with cin. Type bool. The setw manipulator. Type conversions.	7
3	<b>Functions :</b> Call and Return by reference. Overloaded function. Macro Vs. Inline functions. Default arguments, friend functions.	6
4	<b>Object and Classes :</b> Introduction, Structure Definitions, Accessing Members of Structures, Class Scope and Accessing Class Members. Initializing Class Objects: constructors and their types, destructors.	5
5	<b>Operator overloading :</b> Overloading unary operations. Overloading binary operators, data conversion, pitfalls of operators overloading and Type conversion.	8
6	<b>Inheritance :</b> Concept of inheritance. Derived class and based class. Derived class constructors, member function, class hierarchies, public and private inheritance.	2
7	<b>Polymorphism :</b> Pointers in C++, Objects and Pointers, virtual and pure virtual functions, this pointer, Pointers in C++, Objects and Pointers, virtual and pure virtual functions, this pointer, Implementing run time polymorphism.	2
9	<b>Streams and Files :</b> Concept of streams, C++ stream classes, formatted and Unformatted I/O , File stream, manipulators, C++ File stream classes, File modes, File management functions, Binary Files, random Files.	3
10	<b>Templates &amp; Exception Handling :</b> What is template? Function templates and class templates, Overloading Template Functions, Inheritance and Templates, Templates and Friend	3

	Function, Overview and use of Standard Template Library, try-catch-throw, Multiple catch, catch all, Re-throwing Exception.	
	<b>Total Hours</b>	<b>42</b>

**References:**

1. Object Object oriented Programming with C++ by E Balagurusamy, 2001, Tata McGraw Hill
2. C++ Programming, Black Book, Steven Holzner, dreamtech
3. Object Oriented Programming in Turbo C++ by Robert Lafore ,1994, The WAITE Group Press.
4. Compete Reference C++, Herbert Schlitz, TMH

**Suggested Theory distribution:**

The suggested theory distribution as per Bloom's taxonomy is as per follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyse	Evaluate	Create
0%	20%	60%	0%	0%	20%

**List of Experiments:**

1. Learning and implementing basic OOPC concepts related programs like
  - A Program to display WELCOME TO MEFGI and elaborate all the Statements.
  - A program to perform the following:
    - Accept three numbers
    - Calculate average
    - Find smallest and largest numbers
    - Display the results such as average, maximum, and minimum.
  - A program to make a calculator which performs following operations:
    - a. Addition
    - b. Subtraction
    - c. Multiplication
    - d. Division
    - e. Modulo
  - A Write program in C ++ that tells the form of Water whether it is Ice, Water or Steam. Display the menu also as under
    - Temperature Less than 0 = ICE
    - Temperature Greater than 0 & Less than 100 = Water



Temperature Greater than 100 = STEAM

A program to reverse a string.

A program to perform matrix addition.

A program which takes string as an input and counts numbers of occurrences of each vowel and total vowels.

A program to find area of circle, rectangle and triangle using enum datatype.

2. Programs related to practice the usage of functions like

To swap two values (using call by value and call by reference).

Write a Function called zeroSmaller() that is passed two int argument by reference and then set the smaller of the two numbers to 0. Write a main () to call this function.

A program to evaluate following investment equation

$$V = P(1+r)^n$$

and print the tables which would give the value of V for various combination of the following values of P, r, and n:

P: 1000, 2000, 3000, ....., 10000

r: 0.10, 0.12, 0.13, ....., 0.20

n: 1, 2, 3, ....., 10

Write a function power () to raise a number m to a power n without using Math.h header file for following scenario.

The Function takes double value for m and int value for n, and returns the result correctly.

Use a default value of 2 for n to make function to calculate squares when this argument is omitted.

Take an int value for m.

In above all cases function name must be same. Write a main() that calls above 3 cases.

Write a program to obtain the largest of three numbers using inline function.

3. Programs to learn and implement C++ Classes like :

- Create a class that imitates part of the functionality of the base data type int. Call the class Int. The only data in this class is an int variable. Include member functions to initialize an int to 0, to initialize it to an int value, to display it, and to add two Int values. Write a program that exercises this class by creating two initialized Int values, adding these two initialized values, adding these two initialized value and placing the respond in the un-initialized value, and then display this result.
- Define a Class "bank\_account" having following data members and member functions:  
Data members :
  1. Name of the depositor
  2. Account number
  3. Type of account
  4. Balance amount in the account



Member functions :

1. To assign the initial values
  2. To deposit an amount
  3. To withdraw an amount after checking the balance
  4. To display name and balance
- Write a main program to test the program for 10 customers.
  - Create two classes DM and DB which store the value of distances. DM stores distance in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB by keeping following in mind. Use friendly function to carry out the addition. The object that stores the results may be a DM or DB object, Depending on the unit in which the results are required.
  - Write a program to calculate number of object created for particular class. Define a Class “complex” having data members as real and imag and member functions as add\_comp() & show\_comp(). Write a C++ program to get information of 2 complex numbers and add these 2 complex numbers and display this result using Overloaded Constructor Concept.
  - Create a class FLOAT that contains one float data member. Overload all the four arithmetic operators(+, -, \*, /) so that they operate on the objects of FLOAT.
  - Define a class String. Use overloaded ‘==’ operator to compare two strings.
  - Write a program which overload cin and cout of iostream.
  - Define a circle class with radius as data member, necessary constructors and member function to compute area of circle. Class should overload the = = operator to compare two circle objects whether they are equal in radius. Demonstrate its use in main().
4. Programs to implement Inheritance and test with main class.
  5. Demonstrate Runtime Polymorphism by defining media class as Base class and Book and Tape as Sub Class. Keep display () function such that, It provides run time polymorphism.
  6. Create and maintain a File through a C++ program like  
Write a program that reads a text file and create another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space.  
Write a program that reads data from one file and copy it to other file.  
Write a Program which stores name and mobile number in phonebook and performs following tasks:
    - a) Determine the mobile number of the specified person.
    - b) Determine the name of if a mobile number is known.
    - c) Update the mobile number, whenever there is a change.
  7. Experiments related to Standard Template Library
  8. Write a function templates for finding the minimum value contained in an array.
  9. Write a class template to represent a generic vector. Include the member functions to perform the following tasks:



- a. To create a vector
  - b. To modify the value of given element
  - c. To multiply by a scalar value.
10. A table gives a list of car models and the number of units sold in each type in a specified period. Write a program to store this table in suitable container and to display interactively the total value of a particular model sold, given the unit-cost of that model.

**Instructional Method:**

- a) The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.
- b) The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.
- c) Practical examination will be conducted at the end of semester for evaluation of performance of students in laboratory.
- d) Students will use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory.

**Supplementary Resources:**

Students will use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory.

- a) Open source software dev C++
- b) [www.nptel.ac.in](http://www.nptel.ac.in)
- c) [www.learncpp.com](http://www.learncpp.com)