



FACULTY OF ENGINEERING & TECHNOLOGY

Effective from Academic Batch: 2025-26

Programme: Bachelor of Technology (Computer Engineering)

Semester: II

Course Code: 102001226

Course Title: Object Oriented Programming with C++

Course Group: Engineering Science Courses

Course Objectives: To equip graduate students with fundamental and advanced concepts of Object Oriented Programming using C++, fostering skills in design, development, and problem-solving through real-world applications.

Teaching & Examination Scheme:

Contact hours per week			Course Credits	Examination Marks (Maximum / Passing)				
Lecture	Tutorial	Practical		Theory		J/V/P*		Total
				Internal	External	Internal	External	
3	0	2	4	50 / 18	50 / 18	25 / 9	25 / 9	150 / 54

* J: Jury; V: Viva; P: Practical

Detailed Syllabus:

Sr.	Contents	Hours
1	Concepts of OOP: Introduction OOP, Procedural Vs. Object Oriented Programming, Principles of OOP, Benefits and applications of OOP, Program structure, namespace, identifiers, variables, constants, enum, operators, typecasting, control structures	04
2	Objects and Classes: Basics of Object and Class, Private and Public members, Static Data Member and Member Function, Constructors and their types, Destructors, Type Conversion, New and Delete operators, Arrays of Objects, Reference variables.	08
3	Functions and Classes: Simple functions, Call and Return by Reference, Inline functions, Macro Vs. Inline functions, Overloading of functions, Operator Overloading, Default arguments, Friend functions.	05
4	Inheritance: Concept of Inheritance, types of inheritance: single, multiple, multilevel, hierarchical, hybrid, protected members, overriding, virtual base class	06
5	Dynamic Polymorphism: Pointers and Objects, this pointer, virtual and pure virtual functions, Implementing polymorphism	05



6	Exception Handling: Review of Traditional Error Handling, Basics of Exception Handling, Exception Handling Mechanism, Throwing and Catching Mechanisms, Rethrowing an Exception, Specifying Exceptions.	06
7	I/O, File Management and Templates: C++ stream classes, Unformatted and formatted I/O, manipulators, File management functions, File modes, Need and Use of Templates; Function templates and Class templates; STL: Introduction, Containers	06
TOTAL		40

List of Practicals / Tutorials:

1	- Write a program to input a single character and print a message "It is a vowel" if it is a vowel otherwise print "It is a consonant" using if-else structure and OR () operator. - Write a program that prints the tables from 1 to 12. - Write a program to display the following output using a single cout statement. Maths = 90, Physics = 77, Chemistry = 69
2	- Write a program to find the largest of three integers using a function. The function accepts integer arguments by reference. - Write a program that creates an array of user given size using new operator. Initialise and display the array elements.
3	- Design a simple class to perform basic arithmetic functions. Use them in main() function. - Create a C++ program to convert temperature from Fahrenheit to Celsius and display. Use class. - Design classes named Triangle, Square and Circle. Create functions in each class to find areas of particular shape.
4	- Create a function called reverse () that takes two parameters. The first parameter, called str is a pointer to a string that will be reversed upon return from the function. The second parameter is called count, and it specifies how many characters of str to reverse. Give count a default value that, when present, tells reverse () to reverse the entire string. - Write a C++ program to implement function overloading in order to compute power(m,n) where 1) m is double and n is int 2) m and n are int.
5	- Create a program to understand and use static members and static member functions. - Create a class time with members hours and minutes. Write a member function 'add' which takes 2 arguments of type class time and demonstrate the use with a main program. - Create a class 'DISTANCE' with feet and inches as data members. Create member function to input distance, member function to output distance and member function to add two distance objects. Write a main function to create objects of DISTANCE class. Input two distances and output the sum.



6	• Create a class sample with members a and b of type integer. Write a friend function that takes an object as argument and calculates the mean of the two members. • Create a class complex that has two members of type float. Write a friend function that calculate the sum of the two complex objects and returns the result as an object. Demonstrate the working using a main function.															
7	• Create a class with string pointer as data member and member functions: Constructor to allocate memory dynamically and read value, Display() function to display the string and Destructor() function to free allocated memory. Demonstrate the working using a main function. • Write a program to demonstrate the use of copy constructor.															
8	• Write a program to overload the + and – operators for the complex class. (as per the experiment 6) • Write a program to overload the unary – operator for a suitable class. • Write a program to overload the [] operator.															
9	• Write a program to convert a basic type to a class type and vice versa. • Write a program to convert an object of one class to another class.															
10	• Write a program to implement single inheritance. Show the consequences of deriving a class in public, protected and private manner with a simple example. • Consider a class student that stores the roll-number and a class test that stores the marks in two subjects. Inherit a class result from class student and class test, which contains the total marks obtained in the test. Write a program to demonstrate the same.															
11	• Write a program to demonstrate how parameters are passed to the base class constructor via the derived class constructor. • Write a program to understand the use of this pointer.															
12	• Write program to use a base class pointer to point to the derived class object. • Write a program to understand the use of virtual functions. Class media-the base class, two derived classes, tape and book. • Write a program to understand the use of pure virtual functions.															
13	• Write a program to use the following functions: Put(), Get(), Getline(), Write(). • Write a program to produce formatted output using the following functions: Width(), Precision(), Fill(), Setf(), Unsetf(). • Write a program to use manipulators setw, setiosflags and setprecision for formatted output. • Write a program to read a list containing item name, item code, and cost interactively and produce a three column output as shown below. <table><tr><td>NAME</td><td>CODE</td><td>COST</td></tr><tr><td>Turbo C++</td><td>1001</td><td>250.95</td></tr><tr><td>C Primer</td><td>905</td><td>95.70</td></tr><tr><td>.....</td><td>...</td><td>.....</td></tr><tr><td>.....</td><td>...</td><td>.....</td></tr></table>	NAME	CODE	COST	Turbo C++	1001	250.95	C Primer	905	95.70		...			...	
NAME	CODE	COST														
Turbo C++	1001	250.95														
C Primer	905	95.70														
.....	...															
.....	...															



14	- Write a program to create files with constructor function, open function, and using various file mode parameters. - Write a program to use the following functions: Seekg(), Tellg(), Seekp(), Tellp(), Put(), Get(), Write(), Read()
15	- Write a program to implement single inheritance. Show the consequences of deriving a class in public, protected and private manner with a simple example. - Consider a class student that stores the roll-number and a class test that stores the marks in two subjects. Inherit a class result from class student and class test, which contains the total marks obtained in the test. Write a program to demonstrate the same.

Reference Books:

1	Object Oriented Programming with C++, by E Balagurusamy, TMH
2	Object Oriented Programming in Turbo C++, by Robert Lafore, Galgotia
3	The Complete Reference C++, by Herbert Schlitz, TMH
4	C++ : How to Program, by Deitel and Deitel, PHI
5	C++ Programming, by Steven Holzner, Dreamtech

Supplementary learning Material:

1	NPTEL courses: 1. nptel.ac.in/courses/106101208 2. nptel.ac.in/courses/106105151
---	--

Pedagogy:

- Direct classroom teaching
- Assignments/Quiz
- Continuous assessment
- Seminar/Poster Presentation
- Course Projects

Suggested Specification table with Marks (Theory) (Revised Bloom's Taxonomy):

Distribution of Theory Marks in %						R: Remembering; U: Understanding; A: Applying; N: Analyzing; E: Evaluating; C: Creating
R	U	A	N	E	C	
05%	5%	30%	30%	20%	10%	

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes (CO):

Sr.	Course Outcome Statements	%weightage
CO-1	Understand the concept of object oriented programming and differentiate it with procedure oriented programming.	25
CO-2	Apply C++ features such as class, objects, constructors, destructors, inheritance and templates in program design and implementation.	25
CO-3	Assess the OOPs features like Virtual Functions, Polymorphism and Exception Handling.	30
CO-4	Analyze the characteristics of OOPS and build object oriented software using C++.	20



CVM
UNIVERSITY

Aegis: Charutar Vidya Mandal (Estd.1945)

Curriculum Revision:	
Version:	3.0
Drafted on (Month-Year):	April - 2025
Last Reviewed on (Month-Year):	-
Next Review on (Month-Year):	April - 2029