25UCST303	OBJECT ORIENTED PROGRAMMING CONCEPTS IN C++	Category	L	T	P	R	Credit	Year of Introduction
		PBL	3	0	0	1	4	2025

Preamble: This course provides a foundation in object-oriented programming using C++. It covers key OOP principles such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction, along with important C++ features. The course helps students develop structured, reusable programs and strengthens their problem-solving and software design skills through practical applications.

Prerequisite: PIC204 Programming in C

Course Outcomes: After the completion of the course, the student will be able to:

CO	Course Outcomes	Knowledge Level
CO1	Explain basic concepts & structure of Object-oriented programming language using C++	K2 Understand
CO2	Demonstrate the concepts of C++ with various programming problems.	K3 Apply
CO3	Implement various OOP concepts using of C++	K3 Apply
CO4	Implement file handling functions, IO operations and Exception handling mechanism using C++	K3 Apply

Mapping of course outcomes with program outcomes and program specific outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	1	-	2	-	-	-	-	-	2	2	1
CO2	3	3	2	1	3	-	-	1	1	-	2	3	2
CO3	3	3	3	1	3	-	-	1	1	-	2	3	3
CO4	3	3	3	2	3	-	-	1	1	-	2	3	3
Avg	3	2.75	2.25	1.33	2.75	-	-	1	1	-	2	2.75	2.25

Mapping of course outcomes with Knowledge and Attitude Profile (WK)

	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9

CO1	*	*	*	*	*	*		*	
CO2	*	*	*	*	*	*		*	
CO3	*	*	*	*	*	*		*	
CO4	*	*	*	*	*	*		*	

WKs Mapped with the course

WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
*	*	*	*	*	*		*	

Mark Distribution:

Total Marks	CIE Marks	ESE Marks	ESE Duration
100	**60**	**40**	**2.5 hrs**

Continuous Internal Evaluation Pattern:

Attendance : 05 marks
Continuous Assessment Test : 25 marks
Project : 30 marks

SYLLABUS

Module 1 (14 hrs)

Object Oriented Programming Principles: Procedural Oriented Programming (POP), Object Oriented Programming, Difference between POP & OOP, OOPs features, Introduction to C++, Structure of C++ program, C++ Tokens, Datatypes (Basic, Derived, User defined), Console I/O statements, Operators.

Control statements in C++: Conditional statements, Looping statements (for, while, do..while), Arrays in C++, Definition, Array types, array declaration, accessing array data, implementation of array operations.

Module 2 (14 hrs)

Functions in C++: call and return by reference, inline functions, Classes and objects, defining a class, creating objects, accessing data members, calling member functions, Object as parameters, this operators, Scope resolution operator, Access specifiers, Friend Function- Definition, usage, Friend class- Definition, usage.

Constructors and Destructors: Definition, Rules for Defining Constructors, uses, Types (Default Constructor, Parameter Constructor, Copy Constructor), Destructors.

Module 3 (14 hrs)

Polymorphism: Function overloading, implementing overloading on various functions, operator overloading: rules of operator overloading, overloading various operators, Virtual Functions - Definition, implementation of virtual function. Abstract Classes - Implementing abstract classes

Inheritance: Subclass – Super class concepts, Types of Inheritances (Single, Hierarchical, Multilevel, Multiple, Hybrid), Implementing various inheritances, Function overriding, Inheritance levels, Public and Private inheritance, Constructors in derived classes.

Module 4 (14 hrs)

Templates: Definition, generic class, function template, Template function overloading, Exception handling: Try- Catch block, finally, throw, throws, I/O streams- Streams hierarchy

Working with files: Basic file operations, file modes, random access files

Text Books:

1. E. Balagurusamy. (2020). *Object-oriented programming with C++ (8th ed.)*. McGraw Hill Education.
2. B. Trivedi. (2012). *Programming with ANSI C++ (2nd ed.)*. Oxford University Press.
3. D. Ravichandran. (2017). *Programming with C++(1st ed.)*. McGraw Hill Education.

Reference Books:

1. B. Stroustrup. (2013). *The C++ programming language (4th ed.).* Pearson Education.
2. R. Lafore. (2001). *Object-oriented programming in C++ (4th ed.)*. Pearson Education India.
3. Y. Kanetkar. (2020). *Let us C++ (1st ed.)*. BPB Publications.

Suggestion on Project Topics

Students must identify and implement a project using **C++ with Object Oriented Programming concepts**. The selected project topic should satisfy the following requirements:

Project Nature Requirements

i. **User Data Input Requirement**

The project must accept a considerable amount of input data from the user through a menu-driven interface using C++ classes and objects for data handling and validation.

ii. **Permanent Data Storage Requirement**

The project must store a considerable amount of data permanently in the computer using:

- File handling in C++ (text/binary files), and/or
- Simple database connectivity (if implemented additionally).

Data should be organized using appropriate class structures.

iii. **Data Processing Requirement**

The project must process both:

- User-provided data, and
- Stored file/database data to generate meaningful outputs such as:

- Reports
- Summaries
- Search results
- Calculations
- Analytical views

Outputs must be displayed clearly to the user.

Examples: -

- **Develop Student Grading System.** Accept student data and marks for 5 subjects for 5 students. Calculate the percentage and finalize the grade awarded to the student. Write the records into a file.
- **Develop a Quiz Management System**. Quiz should accept student credentials and contain 10 MCQ type questions. Determine the final result. Save the result in a file along with student credentials.
- **Develop advanced calculator** for the following function: Binary to Decimal, Decimal to Binary etc.
- **Develop Hotel Management Application**. It should accept room reservation for 10 rooms. Find the number of empty rooms. Display relevant information and write a maximum of 5 records into file.
- **Develop Employee Management System** using Inheritance. Collect following information from the user:Employee_ID ,Employee_Name, Basic_Salary, Leave taken in the month Calculate Net Salary assuming applicable deductions and display. Write a maximum of 5 records into a file.
- **Any other micro project as suggested by subject faculty**.

PBL Course Elements

L: Lecture (3 Hrs.)	**R: Project (1 Hr.), 2 Faculty Members**		
	Tutorial	**Practical**	**Presentation**
Lecture delivery on OOP concepts (classes, objects, inheritance, polymorphism, file handling)	Project identification based on real-world OOP problems	Simulation / C++ Lab Work / OOP Workshops	Progress Presentation and Final Project Presentation
Group discussion on problem modelling using classes and objects	Project Analysis and class design planning	Data collection (requirements, use cases, sample data sets)	Periodic Evaluation and Review
Question–Answer sessions / Brainstorming on OOP design approaches	Analytical thinking and self-learning of C++ OOP features	Program testing, debugging, and validation	Project Milestone Reviews, Feedback, and Refactoring (if required)
Expert talk / Industry use-case discussion on OOP applications	Case Study / Mini Project Design Report	Prototype development using C++ with OOP features	Poster / Demo / Video Presentation — Students present results in a 2–5-minute demo

Assessment and Evaluation for Project Activity

Sl. No	**Evaluation Rubrics**	**Allotted Marks**
1	Problem Definition & Design	5
2	Implementation Using OOP	5
3	Use of C++ Features	6

4	Code Quality	4
5	Documentation	5
6	Demo & Viva	5
Total		**30**

1. Problem Definition & Design (5 marks)

- Clarity of the problem statement and objectives
- Proper requirement analysis
- Suitable class design (classes, objects, relationships)
- Use of design tools like class diagrams / flowcharts (if applicable)

2. Implementation Using OOP (5 marks)

- Correct use of OOP principles
- Proper creation of classes and objects
- Meaningful data members and member functions
- Logical program structure based on object-oriented approach

3. Use of C++ Features (6 marks)

- Effective use of C++ concepts such as constructors/destructors
- Inheritance, polymorphism, encapsulation, abstraction
- Function/operator overloading where relevant
- Use of STL, templates, exception handling, file handling (if needed)

4. Code Quality (4 marks)

- Readable and well-structured code
- Proper naming conventions
- Indentation and formatting
- Efficient logic and minimal redundancy

5. Documentation (5 marks)

- Clear project report with introduction and objectives
- Design details and class descriptions
- Algorithm / module explanation
- Sample inputs/outputs and screenshots

6. Demo & Viva (5 marks)

- Successful program execution during demo
- Ability to explain code and design choices
- Answers to OOP and C++ concept questions
- Understanding of project workflow

END SEMESTER EXAMINATION – MODEL QUESTION PAPER

Course Code	**25UCST303**
Course Name	**OBJECT ORIENTED PROGRAMMING CONCEPTS IN C++**
Max. Marks: 40	**Duration: 2.5 Hours**

Part A *(Answer all questions. Each question carries 2 marks)*	

Qn No.	Question	Marks	CO	BL
1	Explain the basic difference between C and C++.	2	1	L2
2	Explain with suitable example, the syntax of for loop in C++.	2	1	L2
3	Construct a C++ program that applies the concept of inline functions.	2	2	L3
4	Describe the rules for defining a constructor.	2	2	L2
5	Develop a C++ program that demonstrates the execution of a virtual function.	2	3	L3
6	Explain multiple inheritance and its purpose in C++.	2	3	L2
7	Develop a generic class template to store and display data.	2	4	L3
8	Explain how a file is opened and closed in C++.	2	4	L2
Part B (*Answer any one full question from each Module. Each carry 6 marks*)				
Module I				
Qn No.	**Question**	**Marks**	**CO**	**BL**
9	Compare the looping statements for, while, and do-while in C++ with suitable examples.	6	1	L2
OR				
10	Describe the concept of arrays in C++ and explain how array elements are declared and accessed.	6	1	L2
Module II				
Qn No.	**Question**	**Marks**	**CO**	**BL**
11	Apply object-oriented concepts by developing a class and invoking its member functions through objects.	6	2	L3
OR				
12	Develop suitable C++ programs that utilize (i) constructors with default arguments (ii) copy constructors.	6	2	L3
Module III				
Qn No.	**Question**	**Marks**	**CO**	**BL**

13	Write a C++ program to demonstrate function overloading for the following prototypes.
	add(int a, int b);
	add(double a, double b);
		6	3	L3

OR				
14	Construct a C++ program using hierarchical inheritance to represent different categories of vehicles.	6	3	L3
Module IV				
Qn No.	**Question**	**Marks**	**CO**	**BL**
15	Demonstrate the use of exception handling by implementing a function that throws a division-by-zero exception and catches it using an appropriate catch block.	6	4	L3
OR				
16	Develop a C++ program that applies file handling operations to create a text file, write text into it, and retrieve the contents from the file.	6	4	L3

******