

Branch Name:	IMCA
Program Code:	CS301
Course Title:	Object Oriented Programming with C++ Practical
Course Code:	1CS3010301P
Pre-requisite Course:	Basic Knowledge of Procedural Programming

Course Objectives:

1. This course is intended to make the students aware of Object Oriented Programming (OOP) concepts.
2. Students are expected to learn various terminologies related to OOP such as class, objects, inheritance, polymorphism, encapsulation, abstraction etc.
3. These objectives provide a comprehensive overview of what students should learn in an Object-Oriented Programming with C++ course, covering both theoretical concepts and practical implementation.

Teaching and Examination Scheme:

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture (L)	Tutoria l (T)	Practic al (P)	Credit	Theory (Marks)		Practical (Marks)		Total (Marks)
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
3	-	-	3	-	-	25	25	50

Inductive Practical List

1. Write a cpp program that display "Hello Friends" message on the screen..
2. Write a cpp program the use of a scope resolution operator
3. Write a cpp program the use of a manipulators operator.
4. Write a cpp program the use of reference variable.
5. Write a cpp program the feature of a inline function.
6. Write a cpp program the concept of default arguments
7. Write a program to implement static data member and static member function.
8. Write a cpp program for function overloading
9. Write a cpp program for static class member. (Class member should be a static variable)
10. Write a cpp program concept of an "array of object".
11. Write a cpp program concept of "object as a arguments".
12. Write a cpp program for a friend function.
13. Write a cpp program of a swapping private data of classes
14. Write a cpp program for class with constructors.
15. Write a cpp program for overloaded constructors.
16. Write a cpp program of copy constructors.
17. Write a cpp program of implementation of destructors.
18. Write a cpp program for implementation of unary minus operator.
19. Write a cpp program for implementation of binary plus (+) operator.
20. Write a cpp program for implementation of overloading operators.
21. Write a cpp program for implementation of mathematical operations on strings. {Overloads two operators + and <=}
22. Write a cpp program for implementation of a single inheritance of public data member.
23. Write a cpp program for implementation of a single inheritance of private data member

24. Write a cpp program of multilevel inheritance.
25. Write a cpp program of multiple inheritances.
26. Write a cpp program of hybrid inheritance.
27. Write a cpp program of virtual base class
28. Write a cpp program in which use constructors in derived class.
29. Write a cpp program of initialization list in constructors
30. Write a cpp program for implementation of pointers to objects.
31. Write a cpp program for implementation of array of pointer to objects.
32. Write a cpp program for implementation of this pointer.
33. Write a cpp program for implementation of virtual function.

Text Books:

1. ObjectOrientedProgrammingWithC++,ByEBalagurusamy,TataMcGraw-Hill.
2. ObjectOrientedProgrammingandC++,ByRajaram,NewAgeInternationalPublishers.

Reference Books:

1. TeachYourselfC++,ByHerbertSchildt,TataMcGraw-Hill.
2. LetUsC++,ByYashavantKanetkar,BPBPublications.

List of Open-Source Software/learning website:

- GeeksforGeeks C++ Programming
- W3schools.com /cpp
- tutorialspoint.com/cpp
- javatpoint.com/cpp
- programiz.com/cpp

Course Learning Outcomes (CLO): On completion of this course, the students will be able to:

CLO	Description	Bloom's Taxonomy Level
CLO1	Understanding OOP Concepts: <ul style="list-style-type: none"> • Define and explain the fundamental concepts of Object-Oriented Programming, including encapsulation, inheritance, and polymorphism etc. • Identify and understand the importance of objects, classes, and methods in OOP. 	<ol style="list-style-type: none"> 1. Remember 2. understand
CLO2	C++ Syntax and Language Features: <ul style="list-style-type: none"> • Demonstrate proficiency in C++ syntax and language features relevant to OOP. • Use data types, operators, control structures, and functions in the context of object-oriented programming. 	<ol style="list-style-type: none"> 1. Remember 2. Understand 3. Apply
CLO3	Classes and Objects: <ul style="list-style-type: none"> • Design and implement classes and objects in C++. • Understand the concept of constructors and destructors, and their role in object initialization and cleanup. 	<ol style="list-style-type: none"> 1. Remember 2. Understand 3. Apply 4. create
CLO4	Operator Overloading: <ul style="list-style-type: none"> • Overload operators to provide custom behavior for user-defined types. • Recognize situations where operator overloading is beneficial. 	<ol style="list-style-type: none"> 1. Remember 2. Understand 3. Apply 4. Create 5. Evaluate

CLO5	Inheritance: <ul style="list-style-type: none"> • Create and use inheritance in C++ to establish relationships between classes. • Understand the concepts of base classes and derived classes 	<ol style="list-style-type: none"> 1. Understand 2. Apply 3. Create 4. Evaluate
CLO6	Pointer and Object Relationships: <ul style="list-style-type: none"> • Apply pointers in the context of object-oriented programming. • Understand how pointers can be used to access and modify class members. • Understand and apply virtual functions and pure virtual functions. 	<ol style="list-style-type: none"> 1. Understand 2. Apply 3. Analyze 4. Create 5. Evaluate

Mapping of CLOs with POs & PSOs

Course Learning Outcomes	Program Outcomes (POs)												Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CLO1	H	-	-	L	-	-	L-	-	L	M	-	-	M	L
CLO2	M	M	-	L	-	M	-	L	-	L	-	L	H	L
CLO3	M	-	M	-	L	-	H	-	M	-	H	-	M	L
CLO4	M	M	-	M		L	-	M	-	M	-	M	-	M
CLO5	H	-	L	-	M	-	M	-	H	-	L	-	L	M
CLO6	M	M	-	-	L-	H	-	M	-	L-	M-	L	-	M

H: High, M: Medium, L: Low