

Course Code	18CSC202J	Course Name	OBJECT ORIENTED DESIGN AND PROGRAMMING	Course Category	C	Professional Core	L	T	P	C
							3	0	2	4

Pre-requisite Courses	18CSS101J	Co-requisite Courses	Nil	Progressive Courses	18CSC207J
Course Offering Department	Computer Science and Engineering	Data Book / Codes/Standards	Nil		

Course Learning Rationale (CLR):		The purpose of learning this course is to:		Learning			Program Learning Outcomes (PLO)														
CLR-1 :		Utilize class and build domain model for real-time programs		1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CLR-2 :		Utilize method overloading and operator overloading for real-time application development programs		Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Engineering Knowledge	Problem Analysis	Design & Development	Analysis, Design, Research	Modern Tool Usage	Society & Culture	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning	PSO - 1	PSO - 2	PSO - 3
CLR-3 :		Utilize inline, friend and virtual functions and create application development programs																			
CLR-4 :		Utilize exceptional handling and collections for real-time object oriented programming applications																			
CLR-5 :		Construct UML component diagram and deployment diagram for design of applications																			
CLR-6 :		Create programs using object oriented approach and design methodologies for real-time application development																			
Course Learning Outcomes (CLO):		At the end of this course, learners will be able to:																			
CLO-1 :		Identify the class and build domain model		3	80	70	H	H	M	-	-	-	-	-	H	H	-	-	M	H	H
CLO-2 :		Construct programs using method overloading and operator overloading		3	85	75	H	H	H	H	H	-	M	-	H	H	-	-	M	H	H
CLO-3 :		Create programs using inline, friend and virtual functions, construct programs using standard templates		3	75	70	H	H	M	H	H	-	M	-	H	H	-	-	M	H	H
CLO-4 :		Construct programs using exceptional handling and collections		3	85	80	H	H	H	-	-	-	-	-	H	M	-	-	M	H	H
CLO-5 :		Create UML component diagram and deployment diagram		3	85	75	H	M	M	M	M	M	-	H	H	-	M	M	M	H	H
CLO-6 :		Create programs using object oriented approach and design methodologies		3	80	70	H	H	M	-	-	-	-	-	H	H	-	-	M	H	H

Duration (hour)	15	15	15	15	15
S-1	SLO-1	Comparison of Procedural and Object Oriented Programming	Types of constructor (Default, Parameter)	Feature Inheritance: Single and Multiple	Generic - Templates : Introduction
	SLO-2	OOPS and its features	Static constructor and copy constructor	Inheritance: Multilevel	Function templates
S-2	SLO-1	I/O Operations, Data Types, Variables, static	Feature Polymorphism: Constructor overloading	Inheritance: Hierarchical	Example programs Function templates
	SLO-2	Constants, Pointers, Type Conversions	Method Overloading	Inheritance: Hybrid	Class Templates
S-3	SLO-1	Features: Class and Objects	Example for method overloading	Inheritance: Example Programs	Class Templates
	SLO-2	UML Diagrams Introduction	Method Overloading: Different parameter with different return values		Example programs for Class and Function templates
S-4	SLO-1	Lab 1: I/O operations	Lab 4: Constructor and Method overloading	Lab 7: Inheritance and its types	Lab 10: Templates
	SLO-2	Feature :Class and Objects	Operator Overloading and types	Advanced Functions: Inline, Friend	Exceptional Handling: try and catch
S-6	SLO-1	Examples of Class and Objects	Overloading Assignment Operator	Advanced Functions: Virtual, Overriding	Exceptional Handling: Multilevel exceptional
	SLO-2	UML Class Diagram and its components	Overloading Unary Operators	Advanced Function: Pure Virtual function	Exceptional Handling: throw and throws
S-7	SLO-1	Class Diagram relations and Multiplicity	Example for Unary Operator overloading	Example for Virtual and pure virtual function	Exceptional Handling: finally
	SLO-2	Feature Abstraction and Encapsulation	Overloading Binary Operators	Abstract class and Interface	Exceptional Handling: User defined exceptional
S-8	SLO-1	Application of Abstraction and Encapsulation	Example for Binary Operator overloading	Example Program	Example Programs using C++
	SLO-2				Algorithms: find(), count(), sort()
					Algorithms: search(), merge()

S 9-10	SLO-1	Lab 2: Classes and Objects, Class Diagram	Lab 5: Polymorphism : Operators Overloading	Lab 8: Virtual Function and Abstract class	Lab 11: Exceptional Handling	Lab 15: STL Associative containers and algorithms
	SLO-2					
S-11	SLO-1	Access specifiers – public, private	UML Interaction Diagrams	UML State Chart Diagram	Dynamic Modeling: Package Diagram	Function Object : for_each(), transform()
	SLO-2	Access specifiers - protected, friend, inline	Sequence Diagram	UML State Chart Diagram	UML Component Diagram	Example for Algorithms
S-12	SLO-1	UML use case Diagram, use case, Scenario	Collaboration Diagram	Example State Chart Diagram	UML Component Diagram	Streams and Files: Introduction
	SLO-2	Use case Diagram objects and relations	Example Diagram	UML Activity Diagram	UML Deployment Diagram	Classes and Errors
S-13	SLO-1	Method, Constructor and Destructor	Feature: Inheritance	UML Activity Diagram	UML Deployment Diagram	Disk File Handling Reading Data and Writing Data
	SLO-2	Example program for constructor	Inheritance and its types	Example Activity Diagram	Example Package, Deployment, Package	
S 14-15	SLO-1	Lab 3: Methods and Constructor, Usecase	Lab 6: UML Interaction Diagram	Lab 9: State Chart and Activity Diagram	Lab12 : UML Component, Deployment, Package diagram	Lab15: Streams and File Handling
	SLO-2					

Learning Resources	1. Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Object-Oriented Analysis and Design with Applications, 3 rd ed., Addison-Wesley, May 2007	4. Robert Lafore, Object-Oriented Programming in C++, 4 th ed., SAMS Publishing, 2008 5. Ali Bahrami, Object Oriented Systems Development", McGraw Hill, 2004 6. Craig Larmen, Applying UML and Patterns, 3 rd ed., Prentice Hall, 2004
	2. Reema Thareja, Object Oriented Programming with C++, 1 st ed., Oxford University Press, 2015 3. Sourav Sahay, Object Oriented Programming with C++, 2 nd ed., Oxford University Press, 2017	

Learning Assessment											
	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	20%	20%	15%	15%	15%	15%	15%	15%	15%	15%
	Understand										
Level 2	Apply	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%
	Analyze										
Level 3	Evaluate	10%	10%	15%	15%	15%	15%	15%	15%	15%	15%
	Create										
	Total	100 %		100 %		100 %		100 %		100 %	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc

For the laboratory component the students are advised to take an application and apply the concepts

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
Mr. Girish Raghavan, Senior DMTS Member, Wipro Ltd.	1. Dr. Srinivasa Rao Bakshi, IITM Chennai, sbakshi@iitm.ac.in	1. Ms. C.G.Anupama, SRMIST
Ms. Thamilchelvi, Solutions Architect, Wipro Ltd	2. Dr. Ramesh Babu, N, IITM Chennai, nrbabu@iitm.ac.in	2. Mr. C.Arun, SRMIST
		3. Mr. Geogen George, SRMIST
		4. Mr. Muthukumaran, SRMIST