

<b>Course Code</b>	21CSC101T	<b>Course Name</b>	OBJECT ORIENTED DESIGN AND PROGRAMMING	<b>Course Category</b>	C	PROFESSIONAL CORE	L	T	P	C
							2	1	0	3

<b>Pre-requisite Courses</b>	Nil	<b>Co-requisite Courses</b>	Nil	<b>Progressive Courses</b>	Nil
<b>Course Offering Department</b>	Computer Science and Engineering		Data Book / Codes / Standards		Nil

Course Learning Rationale (CLR):		The purpose of learning this course is to:		Program Outcomes (PO)											
CLR-1 :	Programs using object-oriented approach and design methodologies for real-time application development			1	2	3	4	5	6	7	8	9	10	11	12
CLR-2 :	Method overloading and operator overloading for real-time application development programs			Engineering Knowledge	Problem Analysis	Design/development of solutions	Conduct investigations of complex problems	Modern Tool Usage	The engineer and society	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning
CLR-3 :	Inline, friend and virtual functions and create application development programs														
CLR-4 :	Exceptional handling and collections for real-time object-oriented programming applications														
CLR-5 :	Model the System using Unified Modelling approach using different diagrams														
Course Outcomes (CO):		At the end of this course, learners will be able to:		-	2	2	-	2	-	-	-	-	-	-	3
CO-1:	Create programs using object-oriented approach and design methodologies			-	2	2	-	2	-	-	-	-	-	-	3
CO-2:	Construct programs using method overloading and operator overloading			-	2	2	-	2	-	-	-	-	-	-	3
CO-3:	Create programs using inline, friend and virtual functions, construct programs using standard templates			-	2	2	-	2	-	-	-	-	-	-	3
CO-4:	Construct programs using exceptional handling and collections			-	2	2	-	2	-	-	-	-	-	-	3
CO-5:	Create Models of the system using UML Diagrams			-	2	2	-	2	-	-	-	-	-	-	3

<b>Unit-1 : Introduction to OOPS</b>	<b>9 Hour</b>
Object-Oriented Programming - Features of C++ - I/O Operations, Data Types, Variables-Static, Constants-Pointers-Type Conversions – Conditional and looping statements – Arrays - C++ 11 features - Class and Objects, Abstraction and Encapsulation, Access Specifiers, Methods- UML Diagrams Introduction – Use Case Diagram - Class Diagram.	
<b>Unit-2 : Methods and Polymorphism</b>	<b>9 Hour</b>
Constructors- Types of constructors - Static constructor and Copy constructor -Destructor - Polymorphism: Constructor overloading - Method Overloading Operator Overloading - UML Interaction Diagrams -Sequence Diagram - Collaboration Diagram - Example Diagram	
<b>Unit-3: Inheritance</b>	<b>9 Hour</b>
Inheritance – Types -Single and Multiple Inheritance - Multilevel Inheritance - Hierarchical Inheritance - Hybrid Inheritance - Advanced Functions - Inline, Friend- Virtual - Pure Virtual function - Abstract class - UML State Chart Diagram - UML Activity Diagram	
<b>Unit-4 : Generic Programming</b>	<b>9 Hour</b>
Generic - Templates - Function templates - Class Templates - Exceptional Handling: try and catch - Multilevel exceptional - throw and throws - finally - User defined exceptional - Dynamic Modeling: Package Diagram - UML Component Diagram - UML Deployment Diagram	
<b>Unit-5: Standard Template Library</b>	<b>9 Hour</b>
STL: Containers: Sequence and Associative Container - Sequence Container: Vector, List, Deque, Array, Stack - Associative Containers: Map, Multimap - Iterator and Specialized iterator - Functions of iterator - Algorithms: find(), count(), sort() - Algorithms: search(), merge(), for_each(), transform()	

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

Learning Assessment							
	Bloom's Level of Thinking	Continuous Learning Assessment (CLA)				Summative Final Examination (40% weightage)	
		Formative CLA-1 Average of unit test (50%)		Life Long Learning CLA-2 – (10%)			
		Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	20%	-	20%	-	20%	-
Level 2	Understand	20%	-	20%	-	20%	-
Level 3	Apply	30%	-	30%	-	30%	-
Level 4	Analyze	30%	-	30%	-	30%	-
Level 5	Evaluate	-	-	-	-	-	-
Level 6	Create	-	-	-	-	-	-
	Total	100 %		100 %		100 %	

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
1. Mr. Sagar Sahani, Amadeus Software Labs, Bangalore, <a href="mailto:hello.sagarsahni@gmail.com">hello.sagarsahni@gmail.com</a>	1.Prof. R. Golda Brunet, GCE,Salem, <a href="mailto:goldabrunet@gcessalem.edu.in">goldabrunet@gcessalem.edu.in</a>	1. Mr.C.Arun, SRMIST
2. Mr. Janmajay Singh, Fuji Xerox R&D, Japan, <a href="mailto:janmajaysingh14@gmail.com">janmajaysingh14@gmail.com</a>		2. Mrs.C.G.Anupama, SRMIST