Course Code	Course Title						
116U04L301	Programming Laboratory I@						
	TH			P		TUT	Total
Teaching Scheme(Hrs.)	0			2		1	3
Credits Assigned	0			1		1	2
Examination Scheme	Marks						
	CA		ESE	TW	O	P&O	Total
	ISE	IA					
	-	-	-	50	--	25	75

@ C++ Programming

Course prerequisites: **Programming in C**

Course Objectives:

The major objective of the course is to introduce fundamental concept of Object Oriented Programming (OOP) using C++. Students will be able to develop the skills with the comprehensive capabilities that are required for the efficient programming. Develop applications for a range of problems using object-oriented programming techniques.

Course Outcomes

At the end of successful completion of the course the student will be able to

CO1. Inculcate fundamental concepts of Object Oriented Programming.
CO2. Implement the principles of Data Abstraction Inheritance & Polymorphism.
CO3. Understand the concepts of streams and templates.
CO4. Implement exception handling and test driven development using C++.

Module No.	Unit No.	Details	Hrs.	CO
0	**Introduction to Test Case Driven Development**		**01**	
	0.1	Testing a code, Manual and Automated Testing, Basic structure of Test Case, Writing Test Cases		
1	**Introduction C++ fundamentals**		**02**	**CO1**
	1.1	C Vs. C++, C++ Basics: I/O in C++, Object-Oriented Thinking: Different paradigms for problem solving, need for OOP paradigm, differences between OOP and Procedure oriented programming, Overview of OOP concepts Abstraction, Encapsulation, Inheritance and Polymorphism.		
	1.2	Pointers, Dynamic memory allocation and de-allocation using calloc, malloc and realloc and free, Pointers and functions		
2	**Classes, Data Abstraction & Operator Overloading**		**03**	**CO2**
	2.1	Introduction, Class Scope and accessing Class Members, Separating Interface from Implementation, Controlling Access Function And Utility Functions		
	2.2	Initializing Class Objects: Constructors, Using Default Arguments With Constructors, Using Destructors, Classes : Const(Constant) Object And Const Member Functions, Object as Member of Classes, Friend Function and Friend Classes, Using This Pointer, Dynamic Memory Allocation with New and Delete, Pointers to objects, Static Class Members, Container Classes And Integrators, Proxy Classes, Function overloading.		
	2.3	Fundamentals of Operator Overloading, Restrictions On Operators Overloading, Operator Functions as Class Members vs. as Friend Functions, Overloading, <<, >> Overloading Unary Operators, Overloading Binary Operators.		
3	**Inheritance, Virtual Functions and Polymorphism**		**03**	**CO2**
	3.1	Introduction to Inheritance, Base Classes And Derived Classes, Protected Members, Casting Base- Class Pointers to Derived- Class, Using Member Functions, Overriding Base – Class Members in a Derived Class, Public, Protected and Private Inheritance, Using Constructors and Destructors		

		in derived Classes, Implicit Derived –Class Object To Base-Class Object Conversion, Composition Vs. Inheritance.		
	3.2	Introduction to Virtual Functions, Abstract Base Classes And Concrete Classes, Polymorphism, New Classes And Dynamic Binding, Virtual Destructors, Pointers to virtual functions		
4	**●I/O Streams, files and Templates**		**03**	**CO3**
	4.1	Concept of streams, cin and cout objects, C++ stream classes, Unformatted and formatted I/O, manipulators, File stream, C++ File stream classes, File management functions,		
	4.2	Templates, Exceptions and STL : What is template? function templates and class templates, Overview and use of Standard Template Library,		
5	**●Exception Handling and Test Driven Development**		**03**	**CO4**
	5.1	Introduction, Basics of C++ Exception Handling: Try Throw, Catch, Throwing an Exception, Catching an Exception, Rethrowing an Exception, Exception specifications, Processing Unexpected Exceptions, Stack Unwinding, Constructors, Destructors and Exception Handling, Exceptions and Inheritance.		
	5.2	Test Driven Development using C++		
		Total	**15**	

Module 0 is to introduce the concepts of software testing in the syllabus. It will have no weightage in End-semester examination. It may be evaluated through IA and/or Lab.

 # Self-learning topics may be evaluated through IA and/or Lab.

● Instructor needs to provide additional resources to students for in-depth understanding and practical applicability of the indicated topic/topics.