Course: 102: Object Oriented Programming Methodology

Course Code	**102**
Course Title	**Object Oriented Programming Methodology**
Credit	4
Teaching per Week	4 Hrs
Minimum weeks per Semester	15 (Including Class work, examination, preparation, holidays etc.)
Review / Revision	June 2020
Purpose of Course	This course introduces the concepts of object-oriented programming and skills necessary for developing programs in C++.
Course Objective	1. To make students understand concepts of object-oriented paradigm 2. To make students develop C++ programs 3. To make students learn capabilities of an object-oriented programming language
Course Outcome	CO1- Articulate the principles of Object Oriented Problem solving and programming. CO-2-To demonstrate the differences between traditional imperative design and object Oriented Design CO-3-Outline the essential features and elements of C++ programming language. CO-4- To grasp and apply the concepts of class, method, constructor, abstraction, inheritance and Static Polymorphism. CO-5- To understand and apply Dynamic Polymorphism in real world applications. CO-6- To implement Genericity through the usage of Templates. Mapping between Cos and PSOs

Mapping between COs with PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
CO1	■		■	■	■			
CO2	■			■	■	■	■	
CO3	■					■		
CO4		■	■	■		■	■	
CO5		■			■	■	■	■
CO6		■			■	■	■	■

Pre-requisite	Nil
Course Content	**Unit 1: C++ Basics** 1.1 Data Types 1.2 Pointers 1.2.1 Pointer Arithmetic 1.2.2 Array of Pointers 1.2.3 Dynamic Array 1.3 ios Class 1.4 Input and Output 1.5 Manipulators **Unit 2: Introduction to Object Oriented Programming** 2.1 Structure, classes and Objects 2.2 Encapsulation & Data Hiding 2.3 Constructors 2.4 Friend Functions 2.5 Inline Functions 2.6 Dynamic Object Creation & Destruction 2.7 Static Members 2.8 this Pointer 2.9 Destructors **Unit 3: Object Oriented Properties** 3.1 Introduction to Object Oriented Properties 3.2 Abstraction

	3.3 Polymorphism 3.3.1 Operator Overloading 3.3.2 Function Overloading & Type Conversions 3.4 Inheritance 3.4.1 Types of Inheritance 3.4.2 Constructor & Destructor calls during Inheritance 3.5 Dynamic Polymorphism 3.5.1 Overriding 3.5.2 Virtual Functions 3.5.3 Abstract Class **Unit 4: Data Files and Exception Handling** 4.1 Streams 4.2 File Types and Modes 4.3 File Pointers & their manipulations 4.4 Sequential Input & Output operations 4.5 Random access 4.6 Error handling during File operations 4.7 Exception Handling **Unit 5: Generic Programming and C++ Standard Template Library (STL)** 5.1 Template Classes 5.2 Template Functions 5.3 Implementation of Object-Oriented Properties on Template Classes 5.4 STL 5.4.1 Algorithms 5.4.2 Containers 5.4.3 Functions 5.4.4 Iterators
Reference Books	1. The C++ Programming Language, Stroustrup, Addison Wesley 2. The Complete Reference C++, Schildt, Tata McGraw Hill 3. OOP in Turbo C++, Robert Lafore, Galgotia Publication 4. C++ Primer, Lippman, Addition Wesley 5. Object Oriented Programming with ANSI and Turbo C++, Kamthane, Pearson Education 6. Thinking in C++, Bruce Eckel, Pearson 7. Object Oriented Modelling & Design, Rumbaugh, PHI 8. Object Oriented Analysis & Design with Application, Grady Booch, LPE 9. Standard C++ with Object Oriented Programming, Paul S. Wang, Thomson 10. C++ Primer Plus, Stephan Prata, Addison Wesley 12. Programming with ANSI C++, Bhushan Trivedi, Oxford University Press
Teaching Methodology	Class Work, Discussion, Self Study, Seminars and/or Assignment
Evaluation Method	30% Internal assessment 70% External Assessment