

## Course: 305 : Object Oriented Programming

Course Code	305
Course Title	Object Oriented Programming
Credit	4
Teaching per Week	4 Hrs
Minimum weeks per Semester	15 (Including Class work, examination, preparation etc.)
Review / Revision	June 2015
Purpose of Course	Understand object oriented programming concepts and skills necessary for developing programs using C++.
Course Objective	<p>C++ runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. This course has been designed for the beginners to help them understand the basic to advanced concepts related to C++ Programming languages.</p> <p>To make students understand the importance of OOP methodology.</p> <p>To make students understand exception handling and file handling.</p> <p>To make students understand various types of OOP programming techniques.</p>
Pre-requisite	Basic knowledge of what is computer program and C programming language
Course Out come	After studying this, students will be able to understand how OOP principles work and importance of various coding techniques of OOP. This course will also help students appreciate the role of Exception handling and File handling techniques. After successful completion students will be able to follow particular programming methodology and will understand how to apply it for their application.
Course Content	<p><b>Unit 1. Principles of object oriented programming</b></p> <ol style="list-style-type: none"> <li>1.1. Procedures oriented programming Vs object oriented programming</li> <li>1.2. Basic concepts of object oriented programming (Encapsulation, Polymorphism etc)</li> <li>1.3. Benefits of object oriented programming</li> <li>1.4. Structure &amp; Classes</li> <li>1.5. Encapsulation and Data Hiding</li> <li>1.6. Constructors</li> <li>1.7. Friend Function</li> <li>1.8. Inline Function</li> <li>1.9. Dynamic Object Creation &amp; destruction</li> <li>1.10. Destructor</li> </ol> <p><b>Unit 2. Object Oriented Properties</b></p> <ol style="list-style-type: none"> <li>2.1. Introduction to Object Oriented Properties</li> <li>2.2. Abstraction</li> <li>2.3. Inheritance             <ol style="list-style-type: none"> <li>2.3.1. Type of Inheritance</li> <li>2.3.2. Constructors and Destructor Calls during Inheritance</li> <li>2.3.3. Abstract Class</li> </ol> </li> </ol> <p><b>Unit 3. Polymorphism</b></p> <ol style="list-style-type: none"> <li>3.1 Static Polymorphism</li> </ol>

	<p>3.1.1 Operator Overloading  3.1.2 Function Overloading and Type Conversion  3.2 Dynamic Polymorphism  3.2.1 Overriding  3.2.2 Virtual Function</p> <p><b>Unit 4. Data Files</b>  4.1 Manipulators (In-Built , User Defined)  4.2 File Modes  4.3 File Functions  4.4 Error Handling During File Operation</p> <p><b>Unit 5. Exception Handling</b>  5.1 Introduction to Exception  5.2 Try ... Catch</p>
Reference Book	<ol style="list-style-type: none"> <li>1. Let us C++, Yaswant Kanitkar - TMH Publication</li> <li>2. Programming with C++, E Balaguruswamy - BPB Publication</li> <li>3. C++ and Object Oriented Programming Paradigm, Jana - PHI</li> <li>4. The Complete Reference C++, Herbert Schildt - TMH</li> <li>5. The C++ Programming Language, Stroustrup – Addison Wesley</li> <li>6. OOP in Turbo C++, Robert Lafore - Galgotia Publication</li> <li>7. C++ Primer, Lippman – Addison Wesley</li> <li>8. Object Oriented Programming Fundamentals &amp; Applications, Probal Sengupta – PHI</li> </ol>
Teaching Methodology	Class Work, Discussion, Self Study, Seminars and/or Assignments
Evaluation Method	30% Internal assessment. 70% External assessment.