

**211303**  
**Object Oriented Programming**

<b>Credit Points</b>	<b>Teaching Hrs/Week</b>	<b>Practical Hrs/Week</b>
03	2	2

<b>Objective</b>	<ul style="list-style-type: none"> <li>• Understand the concepts of Object oriented Programming which helps in writing software applications.</li> </ul>
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>▪ Knowledge of Computer Programming in general</li> </ul>

<b>Unit</b>	<b>Topic Name</b>	<b>Details</b>	<b>Hrs</b>
<b>I</b>	<b>Introduction</b>	Types of Programming-Why object oriented programming -Object Oriented Paradigm- Concepts of Object Oriented Programming: Objects and Classes, Data Abstraction and Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Communication-Benefits-Applications. Comparison of Java and C++.	04
<b>II</b>	<b>Classes and Objects</b>	Review of Basic Language Constructs: Datatypes, Variables, Constants, Expressions, Arrays, Types of Statements: Decision Making, Looping. Definition-Class Members: Variables and Methods-Object Creation-Access Specifiers and Visibility Control-Accessing Class Members-Inline Members-Constructors: Definition, Types of Constructors-Destructors-Array of objects-Static Members.	09
<b>III</b>	<b>Polymorphism</b>	Operator overloading: Overloading Unary, Binary operators, Rules for overloading-Function Overloading-Virtual Functions: Pure Virtual Functions.	06
<b>IV</b>	<b>Inheritance</b>	Introduction-Types of Inheritance: Single Inheritance, Multiple Inheritance, Multi-level Inheritance, Hierarchical Inheritance, Hybrid Inheritance-Access Control-Implementation.	04
<b>V</b>	<b>Templates, STL</b>	Introduction-Class Templates-Function Templates Standard Template Library: Introduction-Components of STL: Iterators, Containers, Algorithms-Applications.	05
<b>VI</b>	<b>File Handling</b>	File: Introduction-File Definition-Operations on Files	03

Lab/ Term Work
<p>Term work shall consist of a record in the form of a journal consisting of at least seven assignments based on the topics taught in the theory. Sample list is given below. However staff members may modify them and produce a varied list of programs based on the sample list if required. For each of the following programs, student should write an <b>algorithm, and only then code the program</b>. The student has to take the <b>printout of the programs along with the input/output</b>.</p> <ol style="list-style-type: none"> <li>1. Write a program to represent an account in a bank (define a class) with the following parameters: name of the account holder, account number, type of account, balance amount. Provision should be given to assign initial values and deposit/withdraw a user-given amount, add user accounts, modify user accounts. If withdrawal transaction is to be done check whether the balance after withdrawal is valid. The program should be able to handle the number of customers dynamically. Report should be available for all the account holders of the bank and for individual account holders as per user's input. Use constructors and destructors.</li> <li>2. Write a program to create two classes to store the value of weights of various items in Kilograms and grams and pounds in a grocery store. Provision should be given to add/subtract the weights immaterial of the units. The final result should be in one of the formats i.e., either Kilograms and grams or pounds.</li> <li>3. Write a program to overload unary and binary operators as per the guidance of the in-charge.</li> <li>4. Write a program which demonstrates function overloading and the concept of virtual functions.</li> <li>5. Write a program to implement multiple and multi-level inheritance using a call centre set-up as example.</li> <li>6. Write a program to implement any entity like stack, queue as template.</li> <li>7. Write a program which uses the concept of files to read information regarding a customer from the phone call taken by a call-center professional. Updating, Searching and display operations should be performed . Include a system to also calculate the time of the call in minutes and grade the call center professional.</li> </ol>

<b>Text Books</b>	<ol style="list-style-type: none"> <li>1. Ira Pohl, "Object-Oriented Programming Using C++", Pearson Education Asia, 2003.</li> <li>2. Herbert Schildt, "The Java 2: Complete Reference", Fourth edition, TMH, 2002 (Chapters1-11,13,17)</li> </ol>
<b>Reference Books</b>	<ol style="list-style-type: none"> <li>1. Timothy Budd, "An Introduction to Object Oriented Programming", 2<sup>nd</sup> Edition, Pearson Education Asia.</li> <li>2. E. Balagurusamy, "Object Oriented Programming with C++", 3rd Edition, Tata McGraw Hill.</li> <li>3. Robert Lafore, "Object Oriented Programming in C++", 4<sup>th</sup> Edition, SAMS.</li> <li>4. ISRD Group, " Introduction to Object Oriented Programming through Java",</li> <li>5. E. Balagurusamy, "Programming with Java", 3rd Edition, Tata McGraw Hill.</li> <li>6. Anthony Sintes, "SAMS Teach Yourself Object Oriented Programming in 21</li> </ol>

	Days”, SAMS.
<b>Related Websites</b>	<ol style="list-style-type: none"> <li>1. <a href="http://www.aonaware.com/OOP3.htm">http://www.aonaware.com/OOP3.htm</a></li> <li>2. <a href="http://www.roseindia.net/java/beginners/oop-in-java.shtml">http://www.roseindia.net/java/beginners/oop-in-java.shtml</a></li> <li>3. <a href="http://www.academictutorials.com/oops/oops-concepts.asp">http://www.academictutorials.com/oops/oops-concepts.asp</a></li> <li>4. <a href="http://www.exforsys.com/tutorials/oops.html">http://www.exforsys.com/tutorials/oops.html</a></li> <li>5. <a href="http://java.sun.com/docs/books/tutorial/java/concepts/index.html">http://java.sun.com/docs/books/tutorial/java/concepts/index.html</a></li> </ol>

<b>Examination Scheme</b>	<b>Term Work</b> – 75 marks	Lab Performance, Viva, Exam
	<b>Final Theory Paper</b> – Nil	NIL