

**Structure of Computer Science/Information Technology (IT) Syllabus  
II YEAR III SEMESTER**

**Paper-III : OBJECT ORIENTED PROGRAMMING USING JAVA**

**Course Objectives**

As the business environment becomes more sophisticated, the software development (software engineering is about managing complexity) is becoming increasingly complex. As of the best programming paradigm which helps to eliminate complexity of large projects, Object Oriented Programming (OOP) has become the predominant technique for writing software in the past decade. Many other important software development techniques are based upon the fundamental ideas captured by object-oriented programming.

**Course Outcomes**

At the end of this course student will:

1. Understand the concept and underlying principles of Object-Oriented Programming
2. Understand how object-oriented concepts are incorporated into the Java programming language
3. Develop problem-solving and programming skills using OOP concept
4. Understand the benefits of a well structured program
5. Develop the ability to solve real-world problems through software development in high-level programming language like Java
6. Develop efficient Java applets and applications using OOP concept
7. Become familiar with the fundamentals and acquire programming skills in the Java language.

**UNIT-1**

**FUNDAMENTALS OF OBJECT – ORIENTED PROGRAMMING** :Introduction, Object Oriented paradigm, Basic Concepts of OOP, Benefits of OOP, Applications of OOP, Java features: **OVERVIEW OF JAVA LANGUAGE**: Introduction, Simple Java program structure, Java tokens, Java Statements, Implementing a Java Program, Java Virtual Machine, Command line arguments. **CONSTANTS, VARIABLES & DATA TYPES**: Introduction, Constants, Variables, Data Types, Declaration of Variables, Giving Value to Variables, Scope of variables, Symbolic Constants, Type casting, Getting Value of Variables, Standard Default values; **OPERATORS & EXPRESSIONS**.

**UNIT-II**

**DECISION MAKING & BRANCHING**: Introduction, Decision making with if statement, Simple if statement, if. Else statement, Nesting of if. else statements, the else if ladder, the switch statement, the conditional operator. **LOOPING**: Introduction, The While statement, the do-while statement, the for statement, Jumps in loops.

**Structure of Computer Science/Information Technology (IT) Syllabus**

**CLASSES, OBJECTS & METHODS:** Introduction, Defining a class, Adding variables, Adding methods, Creating objects, Accessing class members, Constructors, Method overloading, Static members, Nesting of methods;

**UNIT-III**

**INHERITANCE:** Extending a class, Overloading methods, Final variables and methods, Final classes, Abstract methods and classes;

**ARRAYS, STRINGS AND VECTORS:** Arrays, One-dimensional arrays, Creating an array, Two – dimensional arrays, Strings, Vectors, Wrapper classes;

**INTERFACES: MULTIPLE INHERITANCE:** Introduction, Defining interfaces, Extending interfaces, Implementing interfaces, Assessing interface variables;

**UNIT-IV**

**MULTITHREADED PROGRAMMING:** Introduction, Creating Threads, Extending the Threads, Stopping and Blocking a Thread, Lifecycle of a Thread, Using Thread Methods, Thread Exceptions, Thread Priority, Synchronization, Implementing the ‘Runnable’ Interface.

**MANAGING ERRORS AND EXCEPTIONS:** Types of errors : Compile-time errors, Runtime errors, Exceptions, Exception handling, Multiple Catch Statements, Using finally statement,

**UNIT-V**

**APPLET PROGRAMMING:** local and remote applets, Applets and Applications, Building Applet code, Applet Life cycle: Initialization state, Running state, Idle or stopped state, Dead state, Display state.

**PACKAGES:** Introduction, Java API Packages, Using System Packages, Naming conventions, Creating Packages, Accessing a Package, using a Package.

**MANAGING INPUT/OUTPUT FILES IN JAVA:** Introduction, Concept of Streams, Stream classes, Byte Stream Classes, Input Stream Classes, Output Stream Classes, Character Stream classes: Reader stream classes, Writer Stream classes, Using Streams, Reading and writing files.

**Reference Books:**

1. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw-Hill Company.

**Structure of Computer Science/Information Technology (IT) Syllabus**

2. John R. Hubbard, Programming with Java, Second Edition, Schaum's outline Series, TATA McGraw-Hill Company.
3. Deitel & Deitel. Java TM: How to Program, PHI (2007)
4. Java Programming: From Problem Analysis to Program Design- D.S Mallik
5. Object Oriented Programming Through Java by P. Radha Krishna, Universities Press (2008)

**Student Activity:**

- 1. Create a front end using JAVA for the student database created**
- 2. Learn the difference between ODBC and JDBC**

**OBJECT ORIENTED PROGRAMMING USING JAVA LAB**

1. Write a program to perform various String Operations
2. Write a program on class and object in java
3. Write a program to illustrate Function Overloading & Function Overriding methods in Java
4. Write a program to illustrate the implementation of abstract class
5. Write a program to implement Exception handling
6. Write a program to create packages in Java
7. Write a program on interface in java
8. Write a program to Create Multiple Threads in Java
9. Write a program to Write Applets to draw the various polygons
10. Write a program which illustrates the implementation of multiple Inheritance using interfaces in Java
11. Write a program to assign priorities to threads in java