

# **Core Java and Advanced Java** **Syllabus**

## **Introduction**

Programming language Types and Paradigms, Computer Programming Hierarchy, How Computer Architecture Affects a Language ? , Why Java ? ,Flavors of Java, Java Designing Goal, Role of Java Programmer in Industry, Features of Java Language, JVM –The heart of Java , Java's Magic Bytecode.

## **The Java Environment:**

Installing Java, Java Program Development , Java Source File Structure ,Compilation, Executions.

## **Basic Language Elements:**

Lexical Tokens, Identifiers, Keywords, Literals, Comments ,Primitive Datatypes, Operators Assignments.

## **Object Oriented Programming**

Class Fundamentals , Object & Object reference , Object Life time & Garbage Collection, Creating and Operating Objects , Constructor & initialization code block, Access Control, Modifiers, methods Nested , Inner Class & Anonymous Classes ,Abstract Class & Interfaces Defining Methods, Argument Passing Mechanism , Method Overloading, Recursion, Dealing with Static Members, Finalize() Method, Native Method. Use of "this " reference, Use of Modifiers with Classes & Methods, Design of Accessors and Mutator Methods Cloning Objects, shallow and deep cloning, Generic Class Types.

## **Extending Classes and Inheritance**

Use and Benefits of Inheritance in OOP, Types of Inheritance in Java, Inheriting Data members and Methods , Role of Constructors in inheritance , Overriding Super Class Methods ,Use of "super" , Polymorphism in inheritance ,Type Compatibility and Conversion Implementing interfaces.

## **Package**

Organizing Classes and Interfaces in Packages , Package as Access Protection , Defining Package ,CLASSPATH Setting for Packages , Making JAR Files for Library Packages Import and Static Import Naming Convention For Packages.

## **Exception Handling:**

The Idea behind Exception ,Exceptions & Errors ,Types of Exception ,Control Flow In Exceptions, JVM reaction to Exceptions ,Use of try, catch, finally, throw, throws in Exception Handling ,In-built and User Defined Exceptions, Checked and Un-Checked Exceptions.

## **Array & String :**

Defining an Array, Initializing & Accessing Array, Multi -Dimensional Array, Operation on String, Mutable & Immutable String, Using Collection Bases Loop for String, Tokenizing a String, Creating Strings using StringBuffer .

## **Thread :**

Understanding Threads , Needs of Multi-Threaded Programming ,Thread Life-Cycle, Thread Priorities ,Synchronizing Threads, Inter Communication of Threads ,Critical Factor in Thread -DeadLock,

## **A Collection of Useful Classes**

Utility Methods for Arrays ,Observable and Observer Objects , Date & Times ,Using Scanner Regular Expression, Input/Output Operation in Java(java.io Package),Streams and the new I/O Capabilities ,Understanding Streams, The Classes for Input and Output, The Standard Streams, Working with File Object, File I/O Basics, Reading and Writing to Files, Buffer and Buffer Management, Read/Write Operations with File Channel, Serializing Objects .

## **GUI Programming**

Designing Graphical User Interfaces in Java, Components and Containers, Basics of Components, Using Containers, Layout Managers, AWT Components, Adding a Menu to Window, Extending GUI Features Using Swing Components, Java Utilities (java.util Package) The Collection Framework : Collections of Objects , Collection Types, Sets , Sequence, Map, Understanding Hashing, Use of ArrayList & Vector.

**Event Handling**

Event-Driven Programming in Java, Event- Handling Process, Event-Handling Mechanism, The Delegation Model of Event Handling, Event Classes, Event Sources, Event Listeners, Adapter Classes as Helper Classes in Event Handling.

**Database Programming using JDBC**

Introduction to JDBC,JDBC Drivers & Architecture, CURD operation Using JDBC, Connecting to non-conventional Databases.

**Java Server Technologies  
Servlet**

Web Application Basics, Architecture and challenges of Web Application, Introduction to servlet, Servlet life cycle, Developing and Deploying Servlets, Exploring Deployment , Descriptor (web.xml), Handling Request and Response.