



# Course Brochure

## Core Java & Adv Java

### Overview

•Core Java training course is intended for students without an extensive programming background. It covers language fundamentals and implementation of pure object oriented programming principles using java API and developing GUI applications. This course also covers how to use built in java classes and how to apply them in the real time environment.

### Pre-requisites

•Basic programming knowledge in any languages like c or c++ is must.

### Applications

## COURSE CONTENTS

### CORE JAVA

#### Introduction

- ❖ Programming language
- ❖ Role of Java Programmer in Industry
- ❖ Java History
- ❖ Features of Java Language
- ❖ JDK, JRE, JVM
- ❖ JVM Architecture

#### Language Fundamentals

- ❖ **The Java Environment.**
  - Installation
  - Java Program Development
  - Java Source File Structure
  - Compilation & Executions
- ❖ **Basic Language Elements**
  - Java Tokens, Identifiers
  - Keywords, Literals, Comments
  - Primitive Data types
  - Type Casting
  - Operators
  - Condition Statements
  - Control Statements
  - Arrays
  - Var-args
  - Command line Arguments
  - Enum Type
  - Assertion

#### Object Oriented Programming

- ❖ OOPS Fundamentals
- ❖ Class & Object
- ❖ Naming Conventions
- ❖ Encapsulation
- ❖ Constructors
- ❖ Method Overloading
- ❖ Recursion
- ❖ Access Modifiers
- ❖ This keyword
- ❖ Design of Accessor and Mutator Methods.
- ❖ Inheritance

- ❖ Method Overriding
- ❖ Covariant Return Types
- ❖ Super Keyword
- ❖ Polymorphism
- ❖ Static Binding and Dynamic Binding
- ❖ Static, final
- ❖ Initialization Blocks & Static Blocks
- ❖ Abstract Class
- ❖ Interfaces
- ❖ Objects Cloning- shallow and deep cloning
- ❖ Strictfp keyword
- ❖ Relationships in Java

#### Wrapper Classes

- ❖ Introduction
- ❖ Byte, Short, Integer, Long
- ❖ Float, Double
- ❖ Character
- ❖ Boolean
- ❖ Autoboxing & Unboxing
- ❖

#### String Handling

- ❖ String
- ❖ StringBuffer
- ❖ StringBuilder
- ❖ StringTokenizer

#### Packages

- ❖ Organizing Classes and Interfaces in Packages
- ❖ Introduction to all pre-defined Packages
- ❖ User Defining Package
- ❖ CLASSPATH Setting for Packages
- ❖ Import and Static Import

#### Exception Handling

- ❖ Exceptions & Errors
- ❖ Control Flow In Exceptions
- ❖ Checked and Un-Checked Exceptions
- ❖ Try and catch block
- ❖ Multiple catch block
- ❖ Nested try block
- ❖ Finally block

## COURSE CONTENTS

- ❖ Throw And Throws
- ❖ Exception Propagation
- ❖ Exception Handling with Method Overriding
- ❖ User Defined Exceptions
- ❖ Final Vs finally Vs finalize

### Inner Classes

- ❖ Introduction
- ❖ Member Inner Class
- ❖ Static Inner Class
- ❖ Local Inner Class
- ❖ Anonymous Inner Class
- ❖ Nested Interface

### Multi Threading

- ❖ Understanding Threads
- ❖ Needs of Multi-threaded Programming
- ❖ Thread Life-Cycle
- ❖ Creating Threads
- ❖ Thread Scheduler
- ❖ Multiple Threads in a program
- ❖ Thread Priorities
- ❖ Joining a Thread
- ❖ Thread Group
- ❖ ShutdownHook
- ❖ Garbage Collection
- ❖ Runtime Class
- ❖ Thread Synchronizing
- ❖ Synchronized Block
- ❖ Deadlock in Java
- ❖ Wait(), notify() and notifyAll()
- ❖ Inter Thread Communication
- ❖ Reentrant Monitor

### Java Reflection

- ❖ Java Reflection API
- ❖ Where Reflection is used
- ❖ Java.lang.Class class
- ❖ newInstance() Method
- ❖ Determining the class object
- ❖ Javap tool
- ❖ Creating javapool
- ❖ Call private methods

### Input/Output Operation in Java(java.io Package)

- ❖ Understanding Streams
- ❖ The Classes for Input and Output
- ❖ The Standard Streams
- ❖ Working with File Object
- ❖ Reading and Writing Files
- ❖ Byte Streams
- ❖ Character Streams
- ❖ Serialization

### Internationalization

- ❖ ResourceBundle class
- ❖ I18N with Date
- ❖ I18N with Time
- ❖ I18N with Number
- ❖ I18N with Currency

### Generics

- ❖ Advantage of Java Generics
- ❖ Generic Class
- ❖ Generic Method
- ❖ Wildcard in Java Generics

### Utility Classes & Collections Framework

- ❖ Date & Time
- ❖ Utility Methods for Arrays
- ❖ Timer and Timer Task for Job Scheduling
- ❖ Using Scanner
- ❖ Introduction to Collection Framework
- ❖ List interface
- ❖ ArrayList, LinkedList & Vector
- ❖ Trace the element of Collection
- ❖ Set interface
- ❖ HashSet, LinkedHashSet & TreeSet
- ❖ Queue Interface
- ❖ PriorityQueue
- ❖ ArrayDeque
- ❖ Map interface
- ❖ HashMap, LinkedHashMap, TreeMap & Hashtable

## COURSE CONTENTS

- ❖ EnumSet
- ❖ EnumMap
- ❖ Comparable & Comparator Interfaces
- ❖ Properties Class
- ❖ Observable and Observer Objects
- ❖ Regular Expression
- ❖ **Networking in Java:**
- ❖ **AWT & Swings:**
- ❖ **Applets:**
- ❖ **Annotations:**

### Java New Features

- ❖ forEach() method
- ❖ default methods in Interfaces
- ❖ Functional Interfaces and Lambda Expersions
- ❖ Java Stream API
- ❖ Java Date/Time API
- ❖ Lambda Expressions
- ❖ Collection API improvements
- ❖ Concurrency API improvements
- ❖ Java IO improvements
- ❖ Method References
- ❖ Functional Interfaces
- ❖ Stream API
- ❖ Strings in Switch
- ❖ Java StringJoiner
- ❖ Binary Literals
- ❖ Underscore Between Literals
- ❖ Multi Catch Similar Exceptions
- ❖ Try with Resources
- ❖ Diamond Syntax
- ❖ Type Interface
- ❖ Instrumentation

## ADV JAVA

### Course Introduction

- ❖ Real-time Java Application Awareness
- ❖ Technology Mapping with Adv Java Concepts
- ❖ Database Applications
- ❖ Web Applications
- ❖ Recap of Core Java

### JDBC

- ❖ Introduction to JDBC
- ❖ JDBC Drivers & Architecture
- ❖ JDBC Driver Types
- ❖ DriverManager, Connection, Statement, ResultSet
- ❖ CRUD Operation using JDBC
- ❖ DAO & DTO
- ❖ Prepared Statement & CallableStatement
- ❖ ResultSetMetaData & DatabaseMetadata
- ❖ Batch Updates
- ❖ Scrollable ResultSet
- ❖ Updatable ResultSet

### Working with Rowsets

- ❖ JdbcRowSet
- ❖ CachedRowSet
- ❖ WebRowSet
- ❖ JoinRowSet
- ❖ FilteredRowSet
- ❖ Handling Binary Data
- ❖ Type mapping & SQL3 Data types
- ❖ ParameterMetaData
- ❖ Transaction Management
- ❖ Connection Pooling
- ❖ JDBC 4.0 Features

### Servlets

- ❖ Web Application Architecture
- ❖ Need of Server side Programming
- ❖ Comparison with CGI
- ❖ Http Protocol & Http Methods
- ❖ Web Servers & Application Servers
- ❖ Introduction to Servlets
- ❖ Servlet API
- ❖ Servlet Life Cycle
- ❖ Apache Tomcat Server
- ❖ Developing and Deploying Servlets
- ❖ Generic Servlets
- ❖ ServletConfig, ServletContext, ServletRequest, ServletResponse
- ❖ Exploring Deployment Descriptor(web.xml)
- ❖ Servlet in Eclipse
- ❖ Database Operation Using Servlets
- ❖ HttpServlets
- ❖ Session Tracking & Management

## COURSE CONTENTS

- ❖ Transferring Request
- ❖ Accessing Web Context
- ❖ Passing INIT and CONTEXT Parameter
- ❖ Filter Servlets
- ❖ Servlet Chaining
- ❖ Servlet-Browser Communication
- ❖ Include & Forward Mechanism
- ❖ Internationalization using Servlets
- ❖ Event Listeners in Servlets
- ❖ ServletRequestWrapper
- ❖ ServletResponseWrapper
- ❖ Web Application Security
- ❖ File Uploading & Downloading
- ❖ Annotation based Servlet Development

### Java Beans

- ❖ Component Model Services
- ❖ Java Beans Properties
- ❖ Bean Persistence
- ❖ Introspection

### Java Server Pages

- ❖ JSP Introduction
- ❖ Basic JSP Architecture
- ❖ Life Cycle of JSP
- ❖ Servlets Vs JSP
- ❖ JSP API
- ❖ JSP Scriptlet Tag
- ❖ JSP Expression Tag
- ❖ JSP Declaration Tag
- ❖ JSP Implicit Objects
- ❖ JSP Page Directive
- ❖ JSP include Directive
- ❖ JSP taglib Directive
- ❖ Exception Handling in JSP
- ❖ JSP Action Elements
- ❖ jsp:forward
- ❖ jsp:include
- ❖ Java Bean Class
- ❖ jsp:useBean
- ❖ jsp:setProperty
- ❖ jsp:getProperty

- ❖ Displaying Applet in JSP
- ❖ Jsp:plugin Jsp:param
- ❖ Jsp:fallback JSP to Servlets & Servlets to JSP
- ❖ JSP Session
- ❖ JSP Expression Language
- ❖ JSP Custom Tags
- ❖ Building war file & deployment in different Servers
- ❖ MVC Architecture
- ❖ WAR file Creation & Deployment

### JSP Standard Tag Library (JSTL)

- ❖ Introduction to JSTL
- ❖ Advantages of JSTL
- ❖ JSTL Core Tags
- ❖ JSTL Function Tags
- ❖ JSTL Formatting Tags
- ❖ JSTL XML Tags
- ❖ JSTL SQL Tags

### AJAX

- ❖ Synchronous & Asynchronous Communications
- ❖ AJAX Introduction
- ❖ AJAX based Web Application Architecture
- ❖ AJAX XMLHttpRequest
- ❖ AJAX Examples

#### Tools

**ANT**      **Log4J**  
**MAVEN**   **jUNIT**

#### SERVERS

**Tomcat**  
**Weblogic**

#### IDEs Using

**Eclipse**  
**NetBeans**  
**MyEclipse**

#### Database

**Oracle**  
**MySQL**