

# JAVA

**Duration: 6 Months**

## Core Topics

### 1. J2SE

- a. OOPS
- b. Abstract & Interface
- c. String
- d. Multithreading
- e. Input Output
- f. Exception Handling
- g. Reflection

### 2. J2EE

- a. Servlet 3.0
- b. Connectivity with Database
- c. Webserver & Web Application
- d. Myclipse
- e. Tomcat
- f. Weblogic
- g. JSP2.1
- h. JSTL
- i. Webservices

### 3. ADVANCE TOOLS:

- a. STRUTS 2
- b. HIBERNATE
- c. SPRING 2.5

### 4. MAJOR PROJECT



Java™

ITI IDNI

# J2SE(JAVA 2 STANDARD EDITION)

## **INTRODUCTION TO JAVA**

- History and Features of Java
- Comparison of C, C++, and Java
- Java Versions and its domain areas
- Life cycle of Java program
- Writing first Java program

## **DATA TYPES, IDENTIFIERS AND VARIABLES**

- Java data types
- The concept of Variables, Operators, and control Statements

## **INPUT/OUTPUT**

- Getting Input from the keyboard using Scanner class
- Displaying Output with println, and print methods

## **CLASSES & OBJECTS**

- The concepts of OOPS
- What is the class and object
- How to create a class and object
- The content of class (blocks, variables, constructors, methods etc)
- Access modifiers
- The Constructors (the internals)

## **METHODS IN JAVA**

- Method prototype and signature
- Method overloading and methods with var-args
- Method with parameters (primitives and reference)
- How to return primitive/reference from a method

## **RELATIONSHIP BETWEEN OBJECTS**

- The implementation of Association
- How to implement association and aggregation in java

## **INHERITANCE & POLYMORPHISM**

- Concept of Inheritance
- The 'this' and 'super' keyword
- The introduction to Object class (the ultimate super-class) and its methods
- The garbage collection in java
- The Concept of Abstract Class and abstract methods
- Concept of Interface
- Multiple Inheritance using interfaces
- Abstract Class Verus Interface

## **EXCEPTION HANDLING**

- What is Exception in Java
- Need of exception in Java
- Difference of exception and error
- The try/catch/throw/throws/finally keywords

- Multiple Exceptions
- Custom Exception

## **MULTI THREADING**

- What is a Java thread
- Life cycle of Java thread
- Writing Thread using Thread Class and Runnable Interface
- Daemon and Non-Daemon threads
- Synchronization in java
- Thread collaboration

## **DATABASE CONNECTIVITY (THE JDBC)**

- Concepts of JDBC
- Need Of Jdbc
- Types Of JDBC Drivers
- Type1: Jdbc-Odbc bridge Driver
- Type2: Native API Driver
- \Type3: Middleware layer Driver
- Type4: Pure Java Driver
- Difference between the JDBC drivers
- Connecting with Databases (MS Access, and Oracle)
- Executing query in database
- Other miscellaneous topics in JDBC

## **PACKAGES**

- Concept of Packages
- Need of packages
- How to create packages using compiler
- How to use packages
- Concept of JAR file
- How to create the JAR file

## **STRING**

- What is String
- Creating String literals
- The constructors of String
- The methods of String
- String immutability and its use

## **STRING BUFFER & STRING BUILDER**

- What is StringBuffer class
- The constructors of StringBuffer
- The methods of StringBuffer
- The StringBuilder and its usage

## **REFLECTION API**

- What is reflection API
- Need of reflection
- The Method, Field, Constructor, and Modifier class
- Implementation of the reflection concepts
- Access private members of a class using Reflection API

## **SERVLETS AND JSP**

- Introduction of Servlet
- Difference b/w CGI, PHP, ASP, and Servlet
- Lifecycle (callback) Methods Of Servlet
- Servlet Implementation & Configuration
- ServletRequest and ServletResponse Interface and their method
- SevletConfig and ServletContext Interface and their method
- Introduction of Web server i.e. Sun Glassfish, Apache Tomcat, Oracle weblogiz
- Introduction to IDE (Integrated Development Environment) like: Eclipse, NetBeans
- The JSP (Java Server Pages)
- The concept of Java Beans
- Using Java Beans with the JSP

## **THE COLLECTION FRAMEWORK**

- Introduction of Collection Framework
- Need of Collection
- Collection API (the classes and interfaces)
- The methods of Collection interface
- List Interface (the ArrayList, Vector, LinkedList, and Stack classes)
- The Iterator, ListIterator, and Enumeration interfaces
- Set Interface (the HashSet, and LinkedHashSet classes)
- Map Interface (the Hashtable, HashMap, LinkedHashMap classes)
- The Comparable and Comparator interfaces
- The TreeMap and TreeSet classes
- The Collections class and its need
- How Collection Framework is used in industry (the real project development)
- Writing an application using Collection API and JDBC

# **J2EE(JAVA 2 ENTERPRISE EDITION)**

## **DATABASE CONNECTIVITY USING JDBC 4.X**

- Overview of native and ODBC Drivers
- Introduction to JDBC
- Type of JDBC drivers
- Using different type of drivers
- Defining properties based Connection Factory
- Performing basic database operations
- Insert, Delete,
- Delete
- Update
- select
- PreparedStatement
- Difference b/w Statement & PreparedStatement
- Setting Query parameters
- Executing Queries
- CallableStatement
- Creating PL/SQL Stored procedures and functions
- Creating Callable statements
- Executing procedures & functions

- Misc
  - Batch Updation
  - Transacting queries
  - Programmatic initialization of database
  - Scrollable & Updatable Resultset
  - ResultSetMetaData
  - DatabaseMetaData
  - Connection Pooling and DataSource
  - Implementation of DAO design pattern

## **SERVLET**

- Introduction to JEE
- Overview of Dynamic web applications
- Introduction to Servlet
- Servlet life cycle
- Annotations and their use in Servlet development
- Difference b/w web server & application server
- Installing and configuring web & application server
- Web Application formats
- Deployment modes supported by servers
- Creating and deploying a simple web application
- Understanding the role of ServletRequest & ServletResponse
- Difference b/w parameters and attributes
- Using different content type for response
- Using RequestDispatcher for Include & Forward
- Initialization Parameters and their use
- ServletConfig & ServletContext
- Inter application collaboration
- State Management and use of HttpSession & Cookies
- Hidden Form fields and URL Rewriting
- Using Http only Cookies
- Listeners and web application events
- Applying Filters to servlets
- File Uploading & downloading
- Security

## **JAVA SERVER PAGES (JSP)**

- Advantages of JSP over Servlet
- JSP Architecture
- Integration of JSP & Servlet API.
- Using Java Beans in JSP
- Defining Custom Tags

## **JSTL & EXPRESSION LANGUAGE**

- Project
- Project Description
- Project Architecture
- Project Conventions and Deliverables
- Project Interface Design
- Creating project prototype
- Testing, Modification & Acceptance...
- Integration Testing and project deployment

## **ANNOTATION:**

- Definition
- Use

## **RMI**

- Stub
- Skeleton
- Marshalling
- Unmarshalling

## **WEB SERVICES**

- WSDL definition
- Soap based web services
- Restful web services
- XFIRE tool for web services

# **STRUTS 2**

## **STRUTS 2 FRAMEWORK**

- MVC/Model2
- Filter Dispatcher
- Action, Result & Interceptors
- ValueStack, OGNL & Data Transfer
- ActionContext

## **WORKING WITH STRUTS 2 ACTIONS**

- Introducing Struts 2 actions
- Packaging your actions
- Implementing actions
- Transferring data onto objects
- File uploading: a case study

## **ADDING WORKFLOW WITH INTERCEPTORS**

- Why intercept requests?Interceptors in action
- Surveying the built-in Struts 2 interceptors
- Declaring interceptors
- Building your own interceptor

## **DATA TRANSFER: OGNL AND TYPE CONVERSION**

- Data transfer and type conversion:
- OGNL and Struts 2
- Customizing type conversion

## **BUILDING A VIEW: TAGS**

- An overview of Struts tags
- Control tags
- Miscellaneous tags
- Using JSTL and other native tags
- A brief primer for the OGNL expression language

## **UI COMPONENT TAGS**

- Why we need UI component tags
- Tags, templates, and themes
- UI Component tag reference

## **RESULTS IN DETAIL**

- Life after the action
- Commonly used result types
- Global results

## **INTEGRATING WITH SPRING AND HIBERNATE**

- Why use Spring with Struts 2?
- Adding Spring to Struts 2
- Why use the Java Persistence API with Struts 2?

## **EXPLORING THE VALIDATION FRAMEWORK**

- Getting familiar with the validation framework
- Wiring your actions for validation
- Validation framework advanced topics

## **UNDERSTANDING INTERNATIONALIZATION**

- The Struts 2 framework and Java i18n
- A Struts 2 i18n demo
- Struts 2 i18n: the details
- Overriding the framework's default locale determination
- SQL Basic & Advanced & Functions
- Working with Procedures and Functions
- Working with Triggers
- Working with Sequences ATE 3.0

## **INTRODUCTION TO MAPPING OBJECTS TO RELATIONAL DATABASE**

- Serialization 2
- XML 2
- Object-Oriented Database Systems
- Mapping

# **HIBERNATE**

## **INTRODUCTION TO HIBERNATE**

- Hibernate Architecture
- Hibernate Configuration
- Hibernate's Support for Other Technologies
- Installing Hibernate
- A 'Hello World' stand-alone application
- A Servlet-Based Hibernate Application

## **CREATING PERSISTENT CLASSES**

- Mapping a Basic Java Class
- Mapping a Class with Binary Data
- Mapping a Read-Only Class
- Mapping Classes Using Versioning/Timestamps
- **MAPPING INHERITANCE WITH JAVA CLASSES**
- Table-Per-Class Hierarchy Mapping
- Table-Per-Subclass Hierarchy Mapping
- Table-Per-Concrete-Class Hierarchy Mapping
- Persisting interfaces

## **WORKING WITH COLLECTIONS**

- Associations
- Lazy initialization
- Mapping Maps/SortedMaps
- Mapping Set/SortedSets
- Mapping a Bidirectional Association
- **USING PERSISTENT OBJECTS**
- Persisting Objects
- Loading Data into an Object

## **SCALAR QUERIES AND HIBERNATE QUERY LANGUAGE**

- Queries
- Named Queries
- SQL Queries
- Hibernate Query Language
- **HIBERNATE CACHING**
- Setting Up a SessionFactory Cache
- Using the Persistent Object Cache
- Using the Query Cache

## **HIBERNATE TRANSACTIONS AND LOCKING**

- Configuration
- Database Support
- Using Transactions
- The Transaction API
- Transaction Example Using Oracle
- Locking

## **HIBERNATE AND XDOCLLET**

- Introduction to XDoclet
- Hibernate XDoclet Tags and Description
- **HIBERNATE AND ECLIPSE**
- Hiberclipse
- Hibernate Synchronizer

## **INTEGRATION OF STRUTS 2, HIBERNATE AND SPRING**

- Struts and Hibernate
- Spring and Hibernate
- Struts, Hibernate and Spring



# SPRING 2.5

## **WHAT IS SPRING?**

- Spring modules
- Understanding dependency injection
- Applying aspect-oriented programming

## **BASIC BEAN WIRING**

- Containing your beans
- Creating beans
- Injecting into bean properties
- Autowiring
- Controlling bean creation

## **ADVANCED BEAN WIRING**

- Declaring parent and child beans
- Applying method injection
- Injecting non-Spring beans

## **ADVISING BEANS**

- Introducing AOP
- Creating classic Spring aspects
- Creating advice
- Defining pointcuts and advisors
- Using ProxyFactoryBean
- Autoproxying

## **HITTING THE DATABASE**

- Learning Spring's data access philosophy
- Configuring a data source
- Using JDBC with Spring
- Working with JDBC templates
- Using Spring's DAO support classes for JDBC
- Integrating Hibernate with Spring
- Caching

## **SPRING AND POJO-BASED REMOTE SERVICES**

- An overview of Spring remoting
- Working with FtM1
- Remoting with Hessian and Burlap
- Using Spring's HttpInvoker
- Spring and web services

## **BUILDING CONTRACT-ONG WEB SERVICES IN SPRING**

- Introducing Spring-WS
- Defining the contract (first)
- Handling messages with service endpoints
- Wiring it all together
- Consuming Spring-WS web services

## **SPRING AND ENTERPRISE JAVABEANS**

- Wiring EJBs in Spring
- Spring and EJB3

## **HANDLING WEB REQUESTS**

- Getting started with Spring MVC
- Handling requests with controllers
- Handling exceptions

## **INTEGRATING WITH OTHER WEB FRAMEWORKS**

- Using Spring with Struts
- Registering the Spring plug-in with Struts
- Working Spring into WebWork 2/Struts 2

## **Course Benefits (USP)**

- ✚ Well-equipped Labs
- ✚ Major Project with Practical Approach
- ✚ Any time Lab facility
- ✚ Worldwide recognized I TURN Certificate
- ✚ Confirmation & Training Letter
- ✚ 100% Job Assistance
- ✚ Quality study material
- ✚ Well Experienced faculty
- ✚ One of the most vast curriculam.