

Java Course Contents



Objectives of this Course:

- To understand the web applications that are scalable, maintainable.
- To understand the architecture and design of web applications.
- To understand Servlet Technology for generation of Dynamic response to create Simple Web Applications.
- To understand how to separate the application concerns based on functionality.
- To understand effective and clean division between controllers, models and view using Spring MVC.
- To understand the Data Persistence using Hibernate Framework.
- To create awareness of Quality Unit Tests, Repeatable Automated Process and Security Concerns.
- To understand the Modern Development techniques using JS Frameworks like Angular JS, Node JS and Express JS.

Prerequisites:

- Knowledge of Basic Programming Techniques, Basic Database Functionalities and Basic SDLC.

Course Outline

Day 1:

Soft Skill Training

Day 2 – 4:

DBMS – DBMS Concepts and SQL

- Introduction to Databases.
- Database Models.
 - Relational Model
- Data Design and Normalization
- Structured Query Language and its categories
 - DDL – DML – DQL – DCL – TCL
- Oracle built in Functions
- SELECT statement varieties with clauses
 - WHERE clause
 - GROUP BY clause
 - HAVING clause
 - ORDER BY clause
- Retrieving data from multiple tables
- Joining the tables – Join variants
 - Equi and Non-Equi Joins
 - Self-Join
 - Cartesian Product
 - Outer Join with modern syntax
- Subqueries
 - Nested Subqueries
 - Co-related Subqueries

DBMS – PL / SQL

- PL/SQL Program Structure.
- Procedural programming using PL/SQL
 - Data Types and Scalar / Anchor Declarations
 - SELECT INTO statement
 - Use of CURSORS in PL/SQL
- Scenario Based Learning
- Modularized programming in PL/SQL
 - Anonymous v/s Named PL/SQL blocks
 - Stored Procedures
 - Functions
 - Triggers
- Exception Handling In PL/SQL

Day 5 – 7:

Core Java – Language Fundamentals

- Different programming paradigms.
- Transformation from OO to PO Paradigm.

- Java Architecture
- A JAVA Program – General Strategy
- Source Code – Byte Code – Path - Class path
- Java – Execution Level Essentials - JDK – JRE – JVM
- Class Loaders and JVM Architecture

Core Java – Object Oriented Programming

- A Look at OOP.
- Objects, Classes and Interfaces
- Influential Reuse Mechanisms in OO Languages
 - Composition – Inheritance
- Extending classes - Inheritance Hierarchy
- super and this
- Abstract classes and overriding methods
- Java Modifiers – Access and Non-Access
- Polymorphism and Polymorphic Behavior
 - Static Binding v/s Dynamic Binding
- Creating and Using Packages and Interfaces
- Scenario based learning

Core Java – Exception Handling

- What is Exception? Java's exception handling mechanism.
- try – catch block
- finally block
- Exception Propagation and throw keyword
- throw v/s throws
- Exception handling with method overriding
- Custom Exceptions
- Scenario based learning

Core Java – Threads and Concurrency

- Multithreaded programming / Lifecycle of Thread.
- Thread Management
- Multithreaded programming
- Extending Thread class – Implementing Runnable Interface
- Synchronization and Thread Safety

Core Java – Collections and Generics

- Collections framework.
- ArrayList – LinkedList – HashMap – Hashtable – HashSet
- Other classes and interfaces in Collection Framework
- Comparable v/s Comparator
- Generic Collection

Core Java – Annotations and Reflections

- Annotations in Java.
 - Levels of annotations

- Built-In Annotations – Custom Annotations
- Reflection API
- Modifying the run time behavior of a class

Unit Testing Using JUnit

- What is test and test case
- What is unit testing
- Why we need unit testing
- How to do unit testing by manual
- Problem with manual unit testing
- Introduction about JUnit
- Advantage of JUnit
- Unit test cases using JUnit

Advance Java - Java Database Connectivity

- JDBC Introduction - JDBC architecture - JDBC driver types.
- JDBC API and various interfaces
 - Connection Interface
 - Statement Interface
 - ResultSet Interface
- Steps to establish a JDBC Connection
- Handling parameterized queries using PreparedStatement Interface
- Embedding PL/SQL objects using CallableStatement interface
- Transaction Management in JDBC

Day 8 - 11:

Web Technologies – Recap of HTML 5 & CSS 3

Web Technologies – JavaScript & jQuery

- Introduction to JavaScript
- Introduction To jQuery
- Selection and DOM Traversal
- Working with JavaScript Events

Advanced Web Technologies – Angular JS

- AngularJS - AJS Introduction
- AJS Expressions, Directives, Controllers
- AJS HTML DOM / AJS HTML Events
- AJS Modules, Forms
- AAjax, Data, and Angular
- Testing in Angular
- AJS Application
- Services and Factories
- Custom Directives
- Internalization

Boot Strap

- Bootstrap - Bootstrap Themes
- Bootstrap CSS Ref
- Bootstrap JS Ref
- Bootstrap Grids

Day 12 - 13:**Advance Java - Servlet**

- Describe web applications
- Define Model-View-Controller (MVC) architecture
- Describe Java Servlet technology
- Describe the web container behavior
- Develop a simple HTTP Servlet
- Configure and deploy a Servlet
- Design a controller component
- Describe MVC Architecture from Servlet perspective.
- Create an HTML form and discuss how HTML form data is sent in an HTTP request

Advance Java - Servlet Technology in Detail

- Sharing Application Resources Using the Servlet Context
 - Describe the purpose and features of the Servlet Context
- Developing Web Applications Using Session Management
 - Describe the purpose of session management
 - Design a web application that uses session management
 - Develop Servlets using session management
 - Describe the cookies implementation of session management
- Describe the URL-rewriting implementation of session management

Using Filters in Web Applications

- Describe the web container request cycle
- Describe the Filter API
- Develop a filter class
- Configure a filter in the web.xml file
- Understanding authentication mechanisms
 - HTTP Basic authentication
 - HTTP Digest authentication
 - HTTP Client authentication
- Defining authentication mechanisms for web applications

Day 14 - 15:**Developing Java Server Pages**

- Describe JSP page technology
- Write JSP code using scripting elements
- Write JSP code using the page directive
- Write JSP code using standard tags
- Using JavaBeans with JSP actions
 - using <jsp:useBean>

- using <jsp:setProperty>
- using <jsp:getProperty>
- using <jsp:include>
- using <jsp:forward>
- Effective use of JSP

Day 16 – 18:

Introduction to Hibernate Framework

- Object Persistence
 - Object/Relational Paradigms
 - O/R Mismatch
 - Object Relational Mapping (ORM)
 - Java ORM/Persistent Frameworks
- Hibernate Architecture
 - Hibernate Architecture and API
 - Hibernate Installation/Setup
 - Configuration
 - Configuration Properties
 - Mapping Files
- Persistent Classes
 - POJOs
 - JavaBeans - Basic Mapping
 - Class to Table Mappings - Property Mapping
 - Identifiers – Generators - Natural Keys
 - Identifier Exposure
- Quoting SQL Tables and Columns
- Working with Persistent Objects
 - Entity Lifecycle
 - Transient State - Persistent State
 - Persistent Object Updates and Automatic Dirty Checking
 - Detached State - Deleted
 - Object Identifiers Revisited
 - Merge
- Associations
 - Many-to-one - Bidirectional Many-to-one
 - Many-to-one List - Many-to-one Map
 - Many-to-one Bag - Many-to-one Array
- Other Associations
 - One-to-one - Bidirectional One-to-one
 - Many-to-many - Bidirectional Many-to-many
 - Many-to-many Identifier Bag
- Value Type Collections
- Inheritance
- Transitive Persistence
 - Custom Mapping

Day 19 - 22:**Introduction to Spring Framework**

- Spring Overview
 - Spring Philosophy
 - Spring Architecture and Modules
- A First Spring Application
 - Dependency Injection
 - Spring Containers
 - BeanFactory
 - ApplicationContext
 - Bean Definitions & Property Injection
 - Bean Lifecycle
 - Wiring Alternatives
 - Autowiring
 - Autowiring Collections
- Pros and Cons of Autowiring

Spring MVC with Spring JDBC Template

- Spring MVC Architecture
 - Spring MVC Configuration and Setup
 - DispatcherServlet
 - Context Configuration
 - Context Loaders
 - InternalResourcesViewResolver - BeanNameViewResolver
 - XMLViewResolver - ResourceBundleViewResolver
 - Using Multiple View Resolvers
- Annotation Based Controllers
- Validation
 - JSR-303 Validation
- Spring Views
- Form Tag Library
 - Binding Form Data
 - Annotation Driven Formatting
- Formatting
 - Formatting Annotations

Integrating Hibernate with Spring MVC**Integration of AngularJS with Spring MVC****Day 23 - 32:****Introduction to Agile (Scrum)****Project Gladiator****Day 33 - 34:****Brainbench Test Preparation & Test**