

# JAVA MICROSERVICES

## Java Language Environment

- Object Oriented
- Platform Independent
- Automatic Memory Management
- Compiled / Interpreted approach
- Robust
- Secure
- Dynamic Linking
- MultiThreaded
- Built-in Networking

## Java Set Up

- Create account in oracle.com
- JDK 7 download  
(<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html> )
- Install JDK
- Download Eclipse  
(<http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/neon3>)

## Java Fundamentals

- Data types
- Operators
- Control Statements
- Arrays
- Enhanced for-loop
- Enumerated types
- Static import
- Auto boxing
- C-style formatted I/O
- Variable arguments
- Hello World Program
- Auto boxing
- C-style formatted I/O
- Variable arguments

## Packages

- Why packages
- Understanding Classpath
- Access modifiers and their Scope

## Operations

- Basic Operators

- Loop Control
- Decision Making
- Numbers
- Characters
- Strings
- Arrays
- Date & Time
- Regular Expressions
- Methods

## Essentials of Object-Oriented Programming

- Object and Class Definition
- Encapsulation
- Polymorphism
- Polymorphism
- Overriding
- Abstraction
- OOP in Java
- Class Fundamentals
- Using Objects
- Constructor
- Garbage Collection
- Method Overloading
- Method Overriding
- Static Members
- Understanding Interface
- Using Interfaces class

## Exception Handling

- Importance of Exception Handling
- Exception Propagation
- Exception Types
- Using try and catch
- throw, throws, finally
- Writing User Defined Exceptions

## Java Util Package / Collections Framework

- Collection and Iterator Interface
- Enumeration
- List and ArrayList
- Vector
- Comparator
- Set Interface and SortedSet
- Hashtable
- Properties

## I/O Operations in Java

- Byte Oriented Streams
- File Handling
- Readers and Writers

## Multithreaded Programming

- Introduction to Multi-Threading
- Understanding Threads and its States
- Java Threading Model
- Thread class and Runnable Interface
- Thread Priorities
- Thread Synchronization
- Inter thread Communication
- Preventing Deadlocks

## DEVELOPING Java APPS

- Defining a Solution
- Organizing a Concept Solution
- Creating a Program Skeleton
- Defining Error Checking Requirements
- Introduction to Application Security

## Network Programming

- Introduction to Networking
- InetAddress
- URL
- TCP Socket and ServerSocket
- UDP Socket
- Developing a Chat Application

## Generics

- Introduction to Generics
- Using Built-in Generics Collections
- Writing Simple Generic Class
- Bounded Generics
- Wild Card Generics

## Inner Classes

- Nested Top-Level Classes
- Member Classes
- Local Classes
- Anonymous Classes

## Abstract Window Toolkit

- Graphics
- Color and Font
- AWT Components/Controls
- Event Handling and Layouts

## Swing Programming

- Introduction to Swing and MVC Architecture
- Light Weight Component
- Swing Hierarchy
- Atomic Components e.g. JButton, JList and more
- Intermediate Container e.g. JPanel, JSplitPane and more
- Top-Level Container e.g. JFrame and JApplet
- Swing Related Events

## SQL

### JDBC

- Introduction to JDBC
- JDBC architecture
- java.sql Package
- Connection, Statement, ResultSet
- Prepared Statement
- Callable Statement
- Scrollable and Updatable ResultSet
- Batch Updates
- ResultSetMetaData
- Simple Transaction Management
- Four Levels of JDBC drivers, their pros & cons
- Features of JDBC 3.0/4.0

### Packagings the application

- Jar
- War
- Ear
- Useful Third-Party Libs
- ApachePOI
- iText PDF

## SQL

- What are different types of DB's in the market
- What are different types of DB's in the market
- What is RDMS
- What is SQL?

- Table basics
- Selecting data
- Creating tables
- Inserting into a table
- Updating records
- Deleting records
- Drop a table
- Advanced Queries
- Standalone SQL interpreter

## Web Application Essentials

- Developer Environment
- Application Servers
- Application servers in the market
- Set up to run applications
- Desktop Application vs Web application

## Servlets

- Need of Server-side Programming
- What is Servlet
- Web Terminology
- Servlet API
- Servlet Interface
- GenericServlet
- HttpServlet
- Servlet Life Cycle
- Servlet Example
- How servlet works?
- War File
- welcome-file-list
- Load on startup
- ServletRequest
- Servlet Collaboration
- RequestDispatcher
- sendRedirect
- ServletConfig
- ServletContext
- Attribute in Servlet
- Servlet Advance
- Session Techniques
- Cookies in Servlet
- Cookies: Login & Logout
- Hidden Form Field
- URL Rewriting
- HttpSessionSession: Login & Logout
- Event and Listener

- Servlet Filter
- What is Filter
- Authentication Filter
- FilterConfig
- Useful examples
- Servlet Miscellaneous
- ServletInputStream
- ServletOutputStream
- Annotation Servlet
- SingleThreadModel
- SSI

## JSP Introduction

- Life cycle of JSP
- JSP API
- JSP scripting elements
- JSP scriptlet tag
- JSP expression tag
- JSP declaration tag
- 9 Implicit Objects
- JSP Request
- JSP Response
- JSP Config
- JSP Application
- JSP Session
- JSP PageContext
- JSP Page
- JSP Exception
- JSP Directive Elements
- JSP page directive
- JSP include directive
- JSP taglib directive
- JSP Exception
- Action Elements
- jsp:forward
- jsp:include
- Java Bean class
- jsp:useBean
- set & getProperty
- Displaying applet in JSP
- Expression Language
- MVC in JSP
- JSTL
- JSP Custom tags
- Example of Custom Tag
- Attributes
- Iteration

- Custom URI
- CRUD Example
- Pagination Example

## Examples

- Registration Example
- Fetching records
- Improving Performance
- Uploading file
- Downloading file
- Servlet Sending Email
- Write data to PDF
- Login Example

## Spring Training Overview

- Spring is a leading Java/J2EE application framework. In this training, you will learn on how to use Spring to rapidly build Java applications.

## Spring Training Prerequisites

- Java and JSP development experience and experience with EJB and Java web services is recommended.

## Spring Training Objectives

- Understand the scope, purpose, and architecture of Spring
- Use Spring's Inversion of Control to declare application components, rather than hard-coding their states and lifecycles
- Use Dependency Injection to further control object relationships from outside the Java code base
- Create validators for business objects, and associate them for application-level and unit-testing uses
- Build a Web application as a Spring DispatcherServlet and associated application context, with declared beans acting as controllers, command objects, and view resolvers
- Build and manage HTML forms with Spring command objects and custom tags
- Use Spring interceptors to implement horizontal features in the Web application
- Connect business objects to persistent stores using Spring's DAO and ORM modules

## Spring Training Course Content

### Spring Tutorial

- What is Spring

- Spring Modules
- Spring Application

## Dependency Injection in Spring

- Constructor Injection
- Injecting Dependent Object
- Constructor Injection with collection
- Constructor Injection with collection having dependent object
- Constructor Injection with Map
- Constructor Injection with Map having dependent object
- Inheriting Bean
- Setter Injection
- Setter Injection with Dependent Object
- Setter Injection with Collection
- Setter Injection with Collection having dependent object
- Setter Injection with Map
- Setter Injection with Map having dependent object
- Constructor Injection vs Setter Injection
- Autowiring
- Factory Method

## Spring AOP

- AOP Concepts
- Spring1.2 AOP DTD
- SP AspectJ Annotation
- SP AspectJ XML

## Spring JdbcTemplate

- JdbcTemplate Example
- PreparedStatement
- ResultSetExtractor
- RowMapper
- NamedParameter
- SimpleJdbcTemplate

## Spring ORM

- Spring with ORM
- Spring with Hibernate
- Introduction of Hibernate

## Spring MVC

- Multiple Controller
- Request Response
- MVC Form Example



- MVC CRUD Example
- MVC Pagination Example
- MVC File Upload

## Spring MVC Tiles

- Spring MVC Tiles

## Spring Java Mail

- Spring Java Mail

## Spring Web

- Spring with Struts2
- Login Example

### 1) Spring Boot Configuration

– Understanding the Spring Boot auto configuration

## Spring Boot Tutorial

- SB Introduction
- Spring Boot Features

## Spring Boot Project

- Spring Maven Project
- SB Project Wizard
- Spring Initializr
- Spring Boot CLI
- Spring Boot Example
- SB Annotations
- Spring Boot DM
- Spring Boot Properties
- Spring Boot Starters
- Spring Boot Actuator

## Spring Boot View

- SB Thymeleaf View
- Spring Boot Database
- Spring Boot Hibernate
- Spring Boot JDBC
- Project Development
- SWAGGER UI
- Terminology notes
- Swagger UI overview

- The Swagger UI Petstore example
- Authorize your requests
- Make a request
- Verify that your pet was created
- Some sample Swagger UI doc sites
- Create a Swagger UI display with an OpenAPI spec document
- Challenges with Swagger UI
- Auto-generating the Swagger file from code annotations
- Sorting out the various Swagger tools
- Project with Swagger

Get Knowledge on [Advanced Java Tutorial](#) from Tek Classes will give you 100% satisfaction in 360-degree angle.

TekClasses