Advance Java Training

Prerequisites: The participants must have strong programming skills with java and JavaEE web applications.

Objectives:

This training introduces following topics in practical oriented approach

- Web services with Java
- REST services with Java
- MicroServices introduction
- Spring Framework for web applications
- Hibernate for ORM
- Spring and Hibernate Integration
- Spring Boot for High level MicroServices
- Agile development Practices TDD and BDD

Training Methodology: The theoretical topics are discussed interactively and technical details are demonstrated with practical examples. The participants work on the hands on exercises which strengthen the concepts learned.

Course Material: Presentations, documents and sample programs with exercise case studies will be shared with the participants.

Classroom Setup: The Intel core compatible CPU with minimum 4GB RAM and 500GB HDD with Windows 7/8/10 64 bit System and JDK1.8 64 bit, Adobe Acrobat Reader, WinZip to be installed. The zip files for Eclipse-Jee-Photon, Apache Tomcat 8.0 web server and others will be shared.
The participants must have admin rights on their systems to run and configure the applications.
Live internet connection with reasonable speed and download permissions should be provided in the training room to update the required patches.

Training Duration: 7 days

Training Facilitator: Prakash Badhe.

Course Plan

DAY 1

Distributed computing with Java

- Client-server web applications
- RPC with Java RMI
- Application to application communication

Introduction to SOA

- Introduction to Service Oriented Architecture
- Defining a service
- Entities of SOA
- Characteristics of SOA
- Different SOA solutions
- Benefits of SOA

XML Introduction

- Internet standards overview
- Introduction to XML
- XML Document parts
- Elements and Attributes
- Namespaces
- XML and XSD
- Schemas and validation
- Java API for XML processing
- DOM and SAX API
- XML-Java serialization and de-serialization
- XSLT (Transformations) overview

Web Services Introduction

- SOA Architecture with web services
- Introduction to Web services
- Need for Web services
- Cost-centric, Integration–centric, Business–driven, Reuse-centric
- Describe previous technologies for Remote Procedure Calls and Messaging
- Web Services as a SOA solution
- The Web Service Architecture
- Role of XML, SOAP, WSDL and UDDI (registering & discovering services) in the Web Services Stack
- Interoperability across platforms
- Advantages of Web Services
- Web service standards
- Web service frameworks

Web Services with Java

- Java web service standards
- Java Web services architecture
- SOAP,WSDL, UDDI standards
- RPC over XML and Http
- Introduction to XML-RPC
- XML serialization
- SOAP Runtime libraries
- SOAP message structure.
- Using Apache CXF Framework
- Web service development approach

- Top down and bottom up build
- XML Data Exchange
- Create web service
- Top down - starting with WSDL
- Bottom up - integrate with application code
- Create web service clients
- Monitor the web services interaction

More with SOAP

- Simple Object Access Protocol (SOAP)
- The SOAP transport mechanisms
- Structure of SOAP Messages
- Header, Body and Fault elements
- Sending and receiving of SOAP Messages
- SOAP with Attachments.
- Error handling with SOAP Faults

DAY 2

Web Services Description Language (WSDL)

- The purpose and functionality of WSDL
- The basic structure of a WSDL document
- Describe the common ways by which parties exchange WSDL documents
- Create web service with WSDL
- Define java clients for the web services

The JAX-WS web services

- The JAX-WS web service standard
- Apache CXF architecture
- Transport configuration and usage
- Asynchronous SOAP services
- Using and implementing interceptors
- JAX-WS client
- The CXF web service configuration
- Create a client/server using CXF and WSDL-first methodology
- Deploy CXF using Spring and/or Tomcat
- Use the service invocation context to acquire incoming message properties
- The MTOM (SOAP with attachments)
- Understanding keys and certificates
- SSL for point-to-point security

DAY 3

Web services Limitations

- SOAP specification versions and compatibilities.
- XML dependencies
- SOAP,WSDL and dependencies
- Interoperability issues
- Heavyweight architecture
- Performance and integration issues.
- Porting/migration issues

The' REST' Way of Services

- Representational State Transfer (REST) architecture
- Exposing the Resource with Identifiers
- REST Operations/verbs with Http methods
- Get,Post,Put,Delete,Head,Options methods
- "CRUD" operations mapped to verbs / methods
    - Use of HTTP
    - Use of URIs
    - Use of Content Types
    - CRUD Operations with http methods
- The JAX-RS standard specifications.
- JAX-RS annotations.
- The REST service java frameworks
- The RestEasy framework configurations.
- Create the first REST Service
- REST service annotations
- The REST service components.
- Applications, Resources, and Providers.
- Define REST Client with Java Net library
- Consume the REST service with RestEasy Client
- REST versus SOAP

JAX-RS Request and Response Annotations

- Path Parameters
- Query Parameters
- Form and Matrix Parameters
- Cookie and Header Parameters
- Response type and configuration
- Utilize the HTTP Headers and HTTP Status Code
- Error Handling

Produce Response

- Supported Return Types
- The output as JSON and XML.
- The Response Class
- Response Entities
- RequestBody and ResponseBody
- Upload/download file
- JAX-RS @Context

- Stateless and Stateful services

DAY 4

MicroServices Introduction

- Monolithic Architecture
- Distributed Architecture
- MicroService and API Ecosystem
- SOA vs. Microservice
- MicroService & API
- Combining MicroServices in single front-end service
- Microservice examples
- Create a new MicroService.
- Define clients and consume the MicroService

Spring Framework Introduction

- Frameworks, libraries and components
- Why Spring?
- Spring Architecture
- Spring features

Dependency Injection in Spring

- Dependency Injection concepts
- IOC as Dependency Injection Container
- Injecting dependencies
- Spring bean configurations.
- Spring Beans and xml
- Delegating component creation to the Spring bean factory
- Dependency Injections with setter methods, constructors and methods implementations
- DI configurations with XML and annotations

DAY 5

MVC Web Applications with Spring

- MVC Architecture
- Separation of implementations
- Spring MVC Components.
- ApplicationContext
- Dispatcher and Controllers
- Dispatcher mappings
- ModelAndView
- View components
- Mapping Components
- View Resolvers

- Internationalization
- Controllers
- Design Web MVC application with Spring

Introduction to Hibernate 4.x

- ORM Framework
- Hibernate Introduction
- Hibernate configuration
- Mapping XML Document
- Build an Hibernate application
- Using SessionFactory and Session objects
- Persistence lifecycle
- Object identity

DAY 6

Data Access with Hibernate

- Different types of HBM mapping
- Loading and updating java objects from database
- Object id to primary key mapping
- Object id generation strategies
- Object state persistent, transient and detached.
- Logging Configuration
- Working with compound keys
- JPA and Hibernate
- Configuration with annotations

Mapping Relationships

- Understand Entity relationships
- Setting up a one to many relationship
- Understanding uni-directional and bi-directional relationship
- Mapping relationships from objects to tables
- Setting up a one to one relationship
- Setting up many to one relationship
- Setting up many to many relationship

Spring ORM and DAO support.

- Spring Hibernate Support.
- Data Access Object (DAO) pattern with Hibernate
- Using HibernateTemplate and HibernateDAOSupport
- Design Spring MVC Web application with Hibernate

DAY 7

Spring Boot Framework

- Spring Boot high level features
- The Spring Boot CLI
- Manage dependencies with maven
- Applications with Spring-Boot
- Unit testing with JUnit

MicroServices with Spring Boot

- Spring Boot Starters
- Building as a Runnable JAR
- Create and deploy REST web services with Spring Boot.
- Manage the security features with Spring Boot.
- Manage JPA Data access application with Spring Boot.
- Building and Deploying an Application

Application Development Practices

Traditional software development process

- Waterfall model
- Pros and cons
- Limitations and Challenges
- Managing change
- Client view of the application
- Switching to TDD
- Making it Agile.

Test-Driven Development Process

- The Unit and Integration Testing
- The Test First Approach
- The TDD process.
- Why Adopt TDD
- Where to Begin TDD
- The TDD cycle : Red-Green-Refactor
- Implement a case study in TDD way
- Using TDD to develop application

Code Refactoring Process

- Change the code structure without affecting behaviour
- Make it more reusable and flexible.
- Test re-factored Code
- The Refactoring techniques
- Refactoring Concepts and Best Practices
- Identify and Implement Refactoring
- Loose coupling with code abstractions

Introduction to BDD

- BDD way of test specifications
- What is BDD
- The Cucumber BDD test framework
- Gherkin Language
- Cucumber JVM usage
- Getting started with Cucumber
- The features, scenarios and steps
- The Cucumber annotations
- Cucumber with JUnit test runner

****