**Hibernate and Spring Training**

**Duration: 5 days (Hibernate 2 days, Spring 3 days)**

Prerequisites:

1. Knowledge of Java Language.
2. knowledge of XML
3. Knowledge of SQL
4. Knowledge of Servlets and JSP
5. Some knowledge of working with Web Servers like apache Tomcat.

Table of Contents:

**Hibernate:**

**Day 1 and Day 2:**

1. Hibernate Basics
   1. Issues with Persistence layers and Object-Relational Mapping (ORM)
   2. Hibernate Overview and Benefits
   3. Elements of Hibernate
   4. Hibernate Architecture and API
   5. Basic Steps
2. Configuring Hibernate
   1. Hibernate Installation/Setup
   2. Configuration
   3. Configuration Properties
   4. Mapping Files
   5. hibernate.cfg.xml
   6. Load the Configuration
   7. Create the SessionFactory
3. Creating Persistent Classes
   1. POJOs
   2. JavaBeans
   3. Basic Mapping
   4. Class to Table Mappings
   5. Property Mapping
   6. Identifiers
   7. Generators
4. Using Persistent Objects
   1. Opening Session
   2. Saving Data
   3. Populate Database
   4. Closing/Flushing the Session
   5. Exception Handling
   6. Loading Data
   7. Updating Data
5. Collections
   1. Collection Types
   2. Simple Value Collections
   3. Entity Collections
   4. Bidirectional Associations
   5. Cascading Operations
   6. Sorting Collections
   7. Composite Elements
6. Simple Composition and Associations
   1. Association Versus Component
   2. Many-to-one
   3. Components
   4. One-to-one
   5. Component Versus One-to-One
7. Inheritance
   1. Table per Subclass
   2. Table per Class Hierarchy
   3. Discriminator
   4. Subclass
   5. Joined-Subclass
8. Hibernate Query Language
   1. Logging SQL
   2. From Clause
   3. Aliases
   4. WHERE Clause
   5. Associations in WHERE Clause
   6. Select Clause
   7. Aggregates
   8. Joins
   9. Use Fetch to Reduce Database Access
   10. Query Interface
   11. Parameter Queries
   12. Hibernate Query
   13. HQL Parameters
   14. Named Queries
   15. Native SQL
   16. 4 Ways to fetch data
   17. Session.list(..)
   18. Session.iterate(..)
   19. Session.scroll(..)
   20. Session.load(..)
9. Hibernate cache management (Demo only)
   1. cache in hibernate
   2. first level cache
   3. Second Level Cache
   4. Cache Configuration

**Spring Framework**

**Day 3**

1. **Introduction to Spring** 
   1. Spring Technology Benefits
   2. Spring Architecture Overview
   3. Quick overview of IoC, AOP, Templates, etc.
2. **Creating your first Spring application (hands on)**
   1. Configuring Spring application context
   2. Installing Spring in your application
   3. Creating a simple application
3. **Understanding IOC (hands on)**
   1. Configuring Beans
   2. Configuring Collaborators
   3. Wiring options (constructor, by type, by name)
   4. Exploring annotations
      1. @Component
      2. @Service,@Repository,@Controller,@Bean etc
      3. @Configuration,@ComponentScan,@Autowired,@Qualifier
4. **Spring AOP(hands on)**
   1. Introduction to AOP
   2. AOP Concepts (JointPoint, Pointcut, Aspect, Introduction)
   3. Spring support for AOP
   4. [AOP XML configuration](http://www.springframework.org/docs/reference/new-in-2.html#new-in-2-aop-configuration)
   5. [Support for @AspectJ aspects](http://www.springframework.org/docs/reference/new-in-2.html#new-in-2-aop-aspectJ)

**Day 4:**

1. **JDBC Support and DAO and Transaction (limited to CRUD Operations)**
   1. JDBC Template (Intro to Templates)
   2. Using DAO support
   3. Demo application using Spring JDBC
   4. Using Spring Transaction
2. **Spring ORM (hands on)**
   1. Integration of Spring with Hibernate
   2. Integrating Spring with JPA
   3. Implement a DAO with Spring JPA/Hibernate
3. **Spring MVC (Demo)**
   1. Spring MVC request cycle
   2. Components of Spring MVC
   3. A Demo Application: Spring MVC login application (3-4 JSP)
4. Spring Boot (**hands on**)
   1. Introduction to Spring Boot
   2. Spring Boot Application configuration (application.properties)
   3. Create a Spring Boot based Web Application

**Day 5:**

1. Spring REST Services (**hands on**)
   1. Introduction to REST Concepts
   2. Http GET, POST, PUT, DELETE etc. methods and how do they work
   3. Implement CRUD application with Spring REST
   4. Integrate Hibernate with the REST Application
   5. Use RestTemplate to access Spring REST Services