



Course Overview

Spring is a general purpose framework that can be used in parts or collectively. Main concepts proposed by spring are Inversion of Control (IOC) and Aspect Oriented Programming (AOP). Apart from this, spring provides Transaction Management support, Implementation of MVC for developing web applications, Template implementations for JDBC and ORM frameworks, facility to integrate enterprise services etc. The purpose of this course is to train Trainee in IOC and AOP implementation of spring as well to introduce them to Template and MVC implementation. This course introduces the Spring Framework, the leading full-stack framework for Java EE applications. Topics covered include the Spring container, dependency injection, data validation, aspect-oriented programming, the JDBC Template, and the Hibernate Template. A Web application is also presented to illustrate the use of the Spring Web MVC design pattern.

Hibernate is widely used ORM framework. Basic functionality of an ORM framework is to persist objects of an application into database i.e. ORM frameworks free application developers from persistence logic so that they can concentrate on implementing the business logic. The purpose of this course is to provide in depth knowledge of Hibernate to the Trainees. This course introduces Hibernate, a popular open-source object/relational mapping (ORM) tool that helps Java developer's store and access persistent objects. Topics covered include Hibernate configuration, the Hibernate mapping file, inheritance, collections, associations, and the Hibernate Query Language (HQL).

Audience Profile

This course combines several courses, the descriptions for who should attend are listed below: This course is for Java developers and architects who wish to explore a popular, open-source alternative to traditional Java EE programming. This course is for Java developers creating or maintaining applications that use a relational database and Java SE or Java EE, Java developers and architects investigating ORM alternatives.

At Course Completion

After completing this course, students will have a fundamental understanding of how to:

- Configure beans in a Spring configuration file.
- Use setter and constructor injection with Spring beans.
- Create property files for error messages and to support internationalization.
- Use the Hibernate template to integrate Hibernate and spring.
- Write forms and controllers for Spring Web applications.
- Describe the purpose and benefits of an object/relational mapping tool
- Configure database connection properties in the Hibernate configuration file
- Use a Hibernate mapping file to map a Java class to a database table
- Distinguish between entity and value types
- Configure primary key generators for persistent classes
- Describe and use the Hibernate strategies for mapping inheritance hierarchies

HIBERNATE v3.0

- **Introduction to Hibernate**
 - Hibernate Architecture & Configuration
 - Hibernate Support for Other Technologies
 - Installing Hibernate
 - A "Hello world" stand alone application
 - Servlet–Based Hibernate application
- **Creating persisting classes**
 - Mapping a basic Java Class, Class with Binary Data, Serializable Class, Class with Data/ calendar attributes, Read-only class, Class using Versioning /Timestamps
- **Mapping Inheritance with java classes**
 - Table-Per – class, subclass , concrete-subclass Hierarchy Mapping
 - Persistence interfaces
- **Working with collections**
 - Associations
 - Lazy initialization
 - Mapping Sets/Sorted Sets, Lists, Arrays, Bidirectional Association
- **Using persistent objects**
 - Persisting Objects
 - Loading Data into an Object
 - Updating and finding Objects
- **Scalar queries and hibernate query language**
 - Queries, Named Queries, SQL Queries
 - Hibernate Queries language
- **Hibernate caching**
 - Setting Up a Session Factory Cache
 - Using the persistent Object cache
 - Setting Up EHcache
- **Hibernate transactions and locking**
 - Configuration
 - Database support
 - Using Transactions
 - The Transactions API
 - Transaction Example Using Oracle
 - Locking
- **Hibernate and eclipse**
 - Hibernate
 - Hiberclipse
 - Hibernate synchronizer

SPRING v3.0

- **What is spring?**
 - Spring modules
 - Understanding dependency Injection
 - Applying aspect-oriented programming
- **Basic bean wiring**
 - Containing your Bean
 - Creating bean
 - Injecting into bean properties
 - Auto wiring
 - Controlling bean creation
- **Advising beans**
 - Introducing AOP
 - Creating classic spring aspects
 - Creating advice
 - Defining Pointcuts and Advisors
 - Using proxyFactory Bean
 - Datatype
 - Autoproxying
- **Hitting the database**
 - Using JDBC with Spring
 - Working with JDBC Templates
 - Integrating Hibernate with Spring
- **Building contract-first web services in spring**
 - Introducing Spring-WS
 - Defining Contract (First!)
 - Handling messages with service endpoints
 - Wiring it all together
 - Consuming Spring-WS Web services
- **Spring and enterprise java beans**
 - Wiring EJBs in Spring
- **Handling web requests**
 - Getting started with Spring MVC
 - Mapping and Handling requests to controller
 - Handling exceptions