

Core Spring

Delivery Methods

- Instructor-led training
- Live-online
- Onsite training

Course Duration

- Four days of instructor-led training
- 50% lecture, 50% hands-on lab

Target Audience

- Developers
- Architects

Prerequisites

- Experience with developing applications using Java

Pricing

Contact your VMware® representative or a VMware Authorized Training Center for pricing information.

More Information

Courses are conveniently scheduled around the world. Go to <http://www.vmware.com/education> to find the class that is right for you.

Onsite training is available for customers who prefer to bring a SpringSource/VMware Certified Instructor to their own facilities. For more information about onsite classes, including facility requirements, go to <http://www.vmware.com/education>.

Course Overview

Core Spring is the four-day flagship Spring Framework training. In this course, students build a Spring-powered Java application that demonstrates the Spring Framework and other Spring technologies like Spring AOP and Spring Security in an intensely productive, hands-on setting.

Completion of this training entitles each student to receive a free voucher to schedule an exam at a Pearson VUE Center to become a Spring Certified Professional.

Course Objectives

At the end of the training, you should have an understanding of Spring and associated technologies and be able to do the following:

- Use the Spring Framework to develop Java applications.
- Use dependency injection to set up and configure applications.
- Test Spring-based applications.
- Set up Spring configuration using XML, annotations, and Java configuration.
- Use Hibernate and JDBC with Spring to access relational databases.
- Use Spring support for transactions.
- Use aspect-oriented programming (AOP) to add behavior to objects.
- Develop a basic Web application with Spring MVC.
- Use Spring Security to secure Web applications.
- Use Spring with RMI, HttpInvoker, and JMS for remote communication.
- Add management with the JMX API.

Course Modules

| | |
|---|--|
| 1 Introduction to Spring <ul style="list-style-type: none"> • The Spring application context • XML configuration • Working with existing singletons and factories • Working with multiple configuration files | 8 Integrating Spring with Hibernate <ul style="list-style-type: none"> • Quick introduction to ORM with Hibernate • Benefits of using Spring with Hibernate • Hibernate configuration in Spring • Exception handling |
| 2 Understanding the Bean Life Cycle <ul style="list-style-type: none"> • XML namespaces • Initialization, use, and destruction phases • Working with Spring interceptors • Externalizing constant values into properties files • Bean scopes | 9 Database Transactions with Spring <ul style="list-style-type: none"> • Declaring a Spring Transaction Manager • @Transactional annotation • Configuring isolation levels • Configuring transaction propagation • Transactions and integration testing |
| 3 Simplifying Application Configuration <ul style="list-style-type: none"> • Bean definition inheritance • Inner beans • p and util namespaces • Dependency injection of collections • Spring Expression Language (SpEL) | 10 Spring in a Web Application <ul style="list-style-type: none"> • Configuring Spring in a Web application (using Spring MVC, Struts, JSF) • Introduction to Spring MVC • Defining Spring MVC controllers using annotations • Spring MVC in the view layer • <mvc/> namespace (new in Spring 3.0) |
| 4 Annotation-Based Dependency Injection <ul style="list-style-type: none"> • Autowiring and component scanning • Stereotype annotations • Java-based configuration • Mixing configuration styles • When to use XML, annotations, and Java configuration | 11 Spring Security <ul style="list-style-type: none"> • What problems does Spring Security solve? • Configuring authentication • Intercepting URLs • The Spring Security tag library for JSPs • Security at the method level • Customizing the Spring Security filter chain |
| 5 Testing a Spring-Based Application <ul style="list-style-type: none"> • Annotations for integration testing with Spring (using JUnit) • Advanced concepts: application context caching and the @DirtiesContext annotation | 12 Remoting <ul style="list-style-type: none"> • Problems with traditional RMI • Using Spring remoting over RMI • Using the Spring HttpInvoker for remote access over HTTP |
| 6 Aspect-Oriented Programming <ul style="list-style-type: none"> • What problems does AOP solve? • Differences between Spring AOP and AspectJ • Defining pointcut expressions • How to use the types of advice: around, before, after | 13 JMS <ul style="list-style-type: none"> • Introduction to JMS • Configuring JMS resources with Spring • Sending and receiving messages using the Spring JmsTemplate and message listener containers |
| 7 Data Access and JDBC with Spring <ul style="list-style-type: none"> • How Spring integrates with existing data access technologies • DataAccessException hierarchy • jdbc namespace • Simplifying jdbc access using the Spring JdbcTemplate | 14 JMX <ul style="list-style-type: none"> • Introduction to JMX • Configuring Spring to export MBeans automatically • Exporting a Spring bean as an MBean |



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com
 © 2011 VMware, Inc. All rights reserved. The product or workshop materials is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/download/patents.html>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

VMware warrants that it will perform these workshop services in a reasonable manner using generally accepted industry standards and practices. THE EXPRESS WARRANTY SET FORTH IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE SERVICES AND DELIVERABLES PROVIDED BY VMWARE, OR AS TO THE RESULTS WHICH MAY BE OBTAINED THEREFROM. VMWARE WILL NOT BE LIABLE FOR ANY THIRD-PARTY SERVICES OR PRODUCTS IDENTIFIED OR REFERRED TO CUSTOMER. All materials provided in this workshop are copyrighted by VMware ("Workshop Materials"). VMware grants the customer of this workshop a license to use and make reasonable copies of any Workshop Materials strictly for the purpose of facilitating such company's internal understanding, utilization and operation of its licensed VMware product(s). Except as set forth expressly in the sentence above, there is no transfer of any intellectual property rights or any other license granted under the terms of this workshop. If you are located in the United States, the VMware contracting entity for the service will be VMware, Inc., and if outside of the United States, the VMware contracting entity will be VMware International Limited.