

## Spring: Core Training

### COURSE DETAILS

---

<b>Course Code:</b>	VM-SCT
<b>Current Version:</b>	1.0
<b>Delivery Type:</b>	Instructor-led
<b>Duration:</b>	4 days

### PREREQUISITES

---

Some developer experience using Java, an IDE (Eclipse, STS or IntelliJ) and build tools such as Maven or Gradle

### COURSE CONTENT

---

This 4-day course offers hands-on experience with the major features of Spring and Spring Boot, which includes configuration, data access, REST, AOP, auto-configuration, actuator, security, and Spring testing framework to build enterprise and microservices applications. On completion, participants will have a foundation for creating enterprise and cloud-ready applications.

This course prepares students for the Spring Professional certification exam.

### COURSE OBJECTIVES

---

By the end of the course, you should be able to meet the following objectives:

- Spring configuration using Java Configuration and Annotations
- Aspect oriented programming with Spring
- Testing Spring applications using JUnit 5
- Spring Data Access - JDBC, JPA and Spring Data
- Spring Transaction Management
- Simplifying application development with Spring Boot
- Spring Boot auto-configuration, starters, and properties
- Build a simple REST application using Spring Boot, embedded Web Server and fat JARs or classic WARs
- Implementing REST client applications using RestTemplate and WebClient
- Spring Security
- Enable and extend metrics and monitoring capabilities using Spring Boot actuator
- Utilize Spring Boot enhancements to testing

### COURSE MODULES

---

#### 1 Introduction to Spring

- Java configuration and the Spring application context
- @Configuration and @Bean annotations
- @Import: working with multiple configuration files
- Defining bean scopes
- Launching a Spring Application and obtaining Beans

#### 2 Spring JAVA Configuration: A Deeper Look

- External properties & Property sources
- Environment abstraction

## Spring: Core Training

- Using bean profiles
  - Spring Expression Language (SpEL)
- 3 Annotation-based Dependency Injection
- Component scanning
  - Autowiring using @Autowired
  - Java configuration versus annotations, mixing.
  - Lifecycle annotations: @PostConstruct and @PreDestroy
  - Stereotypes and meta-annotations
- 4 Factory Pattern in Spring
- Using Spring FactoryBeans
- 5 Advanced Spring: How Does Spring Work Internally?
- The Spring Bean Lifecycle
  - The BeanFactoryPostProcessor interception point
  - The BeanPostProcessor interception point
  - Spring Bean Proxies
  - @Bean method return types
- 6 Aspect-oriented programming
- What problems does AOP solve?
  - Defining pointcut expressions
  - Implementing various types of advice
- 7 Testing a Spring-based Application
- Spring and Test-Driven Development
  - Spring 5 integration testing with JUnit 5
  - Application context caching and the @DirtiesContext annotation
  - Profile selection with @ActiveProfiles
  - Easy test data setup with @Sql
- 8 Data Access and JDBC with Spring
- How Spring integrates with existing data access technologies
  - DataAccessException hierarchy
  - Spring's JdbcTemplate
- 9 Database Transactions with Spring
- Transactions overview
  - Transaction management with Spring
  - Transaction propagation and rollback rules
  - Transactions and integration testing
- 10 Spring Boot Introduction
- Introduction to Spring Boot Features
  - Value Proposition of Spring Boot
  - Creating a simple Boot application using Spring Initializer website
- 11 Spring Boot Dependencies, Auto-configuration, and Runtime
- Dependency management using Spring Boot starters
  - How auto-configuration works
  - Configuration properties
  - Overriding auto-configuration
  - Using CommandLineRunner

## Spring: Core Training

### 12 JPA with Spring and Spring Data

- Quick introduction to ORM with JPA
- Benefits of using Spring with JPA
- JPA configuration in Spring
- Configuring Spring JPA using Spring Boot
- Spring Data JPA dynamic repositories

### 13 Spring MVC Architecture and Overview

- Introduction to Spring MVC and request processing
- Controller method signatures
- Using @Controller, @RestController and @GetMapping annotations
- Configuring Spring MVC with Spring Boot
- Spring Boot packaging options, JAR or WAR

## WHO SHOULD ATTEND

---

Application developers who want to increase their understanding of Spring and Spring Boot with hands-on experience and a focus on fundamentals.