Document Generated: 12/15/2025

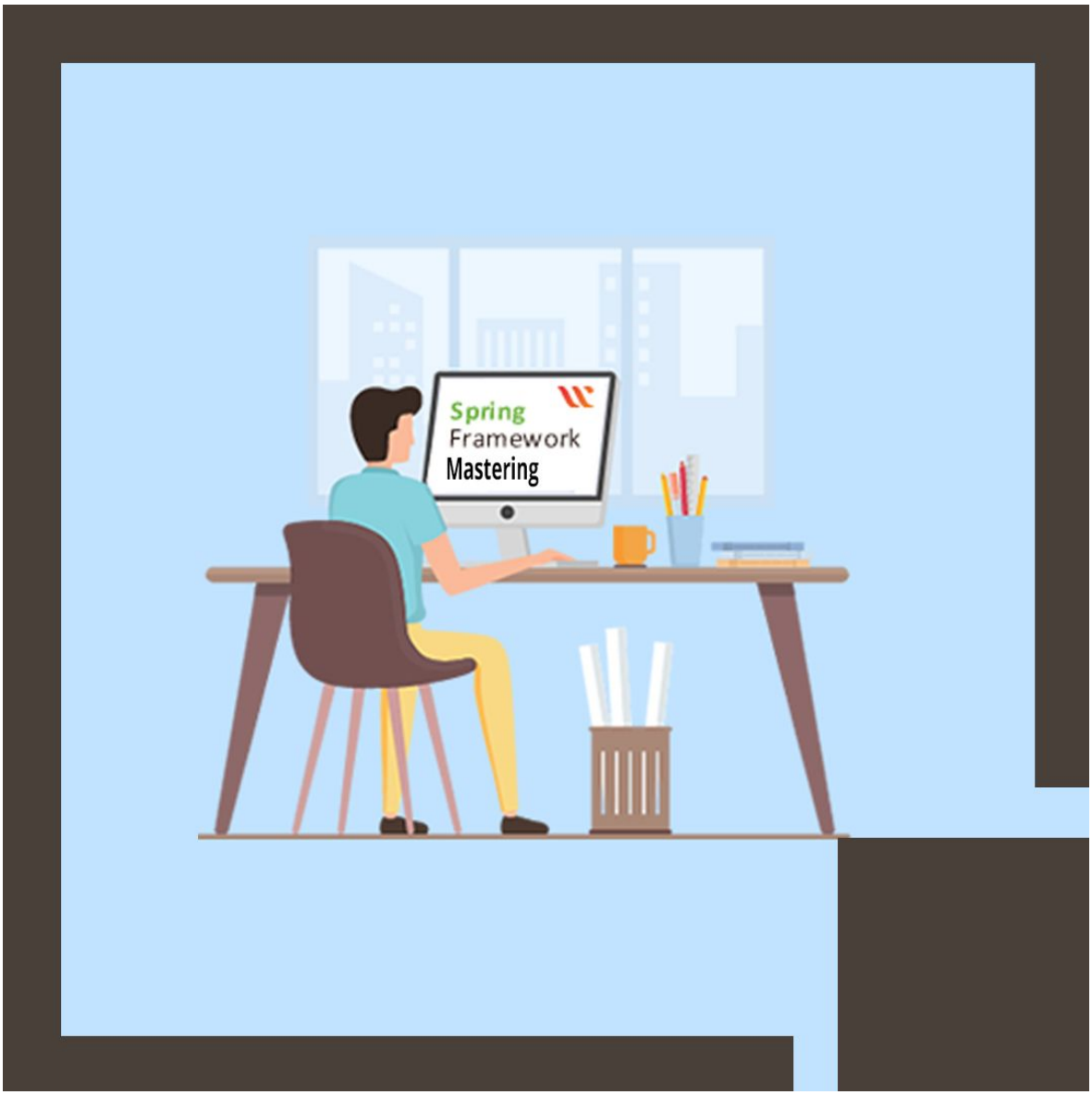
Learning Style: Virtual Classroom

Technology:

Difficulty: Intermediate

Course Duration: 5 Days

Mastering Spring 4.3 (TT3330-S4)



About this course:

Learning the Spring 4.x Framework is a 5 days practical Spring training program for experienced Java developers who want to learn what the Spring Framework is for today's architectures and systems, and how Spring can be used in combination with other frameworks and technologies. This online training provides additional coverage of Aspect-Oriented programming from Spring and the use of Spring Security, and also Spring Boot. It provides a full overview of all new functions in Spring 4.x. Learners can obtain practical experience collaborating with Spring, utilizing Maven for dependency and project management and, additionally, a test-driven method (utilizing JUnit) to the workshops in the program.

The Spring system is a framework that offers a lightweight container that supports the development in the non-invasive fashion of simple to complex components. Flexibility and openness from Spring align with and help gradual growth and testing. The structure of the platform facilitates the layering of features like view-oriented interfaces, transactions, persistence, and capabilities and enterprise systems.

This program aims Spring 4.x, which provides complete support for Java 8 and Java EE 7 (it continues to support previous versions of Java and Java EE). In most of its application program interfaces, Spring encourages the use of lambda expressions and process references.

Spring encourages the production of Java EEs. Spring streamlines specific tasks and favors strong interface-based design. Spring makes it simpler to customize your program and reduces the need for many Java EE design patterns. Spring brings the OO style back into your Java EE app and suits well with other view innovations and HTML5's latest features.

The Java Spring Developer can earn an average salary of $117,087 per annum.

Course Objective:

· Grasp Spring's relationship with AOP, JEE, JDBC, IOC, JSF, Hibernate, EJBs, JMS, and Struts

· Understanding and working to integrate consistency into a Spring program

· Work with JavaConfig and Spring Boot to build Spring apps more effectively and more efficiently

· Demonstrate the problems of complex frameworks like Java EE and how Spring tackles those problems

· Comprehend how to customize the system with annotations, JavaConfig, and XML, and discover each option's benefits

· Comprehend and deal with different options for the integration of view-oriented systems into Spring for web apps

· Create apps that reap the benefits of the Spring container and the declarative design of code assembly of simple modules

· Demonstrate the support Spring has for caching and transactions

· Work with different Spring App Events, implement and register listeners, and track transactional incidents

Audience:

This is a Spring 4.x intermediate-level online training program for developers who want to learn when and how to use Spring in Java and Java EE apps.

Prerequisite:

The participants should have practical experience in fundamental Java development.

Course Outline:

Module 1: Introduction to Spring

Lesson: The Spring Framework

- Understand the value of Spring
- Explore Dependency Injection (DI) and Inversion of Control (IoC)
- Configuring collaborators
- Understand built-in property editors
- Tutorial: Setup Eclipse Oxygen for Using Maven

Lesson: Configuring Spring

- Discover the Spring Container
- Introduce the various ways to configure the Spring context
- Initialize the Spring container
- Accessing beans in the Spring container
- Configure beans using XML
- Resolving bean dependencies using XML
- Exercise: Create a Simple Spring program using XML configuration

Lesson: Advanced Configuration

- Use Spring's special factory beans
- Re-use bean definitions in the configuration file using the "parent" bean concept
- Use property placeholders in the configuration file
- Create custom property editors
- Become familiar with container post-processors
- Exercise: Configuring Dependencies

- Exercise: Advanced XML Configuration

Lesson: Annotation based configuration

- Introduce Spring annotation for defining dependencies
- The @Autowired annotation
- Stereotyping Annotations
- Enabling and filtering the component scan
- Qualifying injection points
- Lifecycle annotations
- The @Value annotation
- Exercise: Create a Simple Spring program using annotations

Lesson: Java-based Configuration

- Introduce Java-based configuration
- The @Configuration and @Bean annotations
- Define bean dependencies
- Define bean scopes
- Bootstrapping Java Config context
- Injection in Configuration classes
- Exercise: Create a Simple Spring program using JavaConfig

Lesson: More Java-based configuration

- Registering BeanFactoryPostProcessorT
- he Environment API
- Using context Profiles
- Conditionally loading beans and configurations
- Using properties in Java based configuration
- The DependsOn and Lazy annotations
- Exercise: Java Based Configuration
- Exercise: Using Profiles

Module 2: Persistence in Spring

Lesson: Overview: Persistence in Spring

- DAO Implementation
- Transaction Support
- Spring Support for JCache
- Spring Data: JPA to NoSQL

Lesson: Spring JDBC

- JdbcDaoSupport - JDBC DAO Implementation
- The jdbcTemplate
- Exception Handling
- Operation Classes

- Exercise: Using Spring JDBC

Module 3: Spring and the Web

Lesson: MVC Overview

- The DispatcherServlet
- Workflow of Request Handling
- Using Handler Mappings
- ModelAndView and View
- Controllers
- Spring Form Tags
- Exercise: Using Spring MVC

Module 4: Advanced Topics

Lesson: Introduction to Aspect Oriented Programming

- Aspect Oriented Programming
- Cross Cutting Concerns

Lesson: Spring AOP

- Spring's AOP in a Nutshell
- The Three Technologies of "Weaving"
- Spring Advice Types
- Exercise: Spring AOP - Combined Advice

Module 5: Spring Boot

Lesson: Spring IO Platform

- Understand the Spring IO Platform
- Understand the IO Bill of Materials
- Understand the IO Foundation
- Learn how the IO Execution will be leveraged
- Learn how Spring Cloud is used for Platform Coordination

Lesson: Spring Boot Overview

- What is Spring Boot
- Explore Spring Boot starters
- Examine Spring Boot’s AutoConfiguration as well as its command-line interface (CLI)
- Understand the Spring Boot Actuator

Lesson: Spring Boot Introduction

- Spring Boot JPA Starter
- Examine Spring Boot’s AutoConfiguration
- Understand the Spring Conditionals
- Understand Spring Boot DevTools
- Exercise: Create a "REST JPA Respository"

Lesson: Advanced Spring Boot

- Explore additional Spring Boot starters
- Bootstrapping Spring Boot
- Understand Spring Boot Actuators
- Create and run a Spring Thymeleaf MVC application
- Exercise: Create a “Thymeleaf MVC With JPA Repository”

Module 6: Spring Security Framework

Lesson: Enterprise Spring Security

- Spring Security Framework
- Security Interceptors
- Authentication Managers
- Wiring in Encoders and Salts
- Access Decision Managers

Lesson: Spring Web Security

- Spring Security Transparent to Client
- Standard Set of Filters
- Spring Security Config File
- Securing Java Code
- Securing Java Spring Beans
- Exercise: Using Spring Security (Pt 1)
- Exercise: Using Spring Security (Pt 2)
- Exercise: Using Spring Security (Pt 3)
- Exercise: Using Spring Security (Pt 4)

Module 7: Spring JMS

Lesson: JMS Overview (Optional)

- Java Message Service (JMS)
- The JMS Factory Model
- JMS Queue Architecture
- Topic Architecture
- Messages

Lesson: Spring and JMS

- JmsTemplate
- Callback Methods
- Spring Messaging Module
- Message Converters
- MessagePostProcessor
- Destinations
- Working with @JmsListener
- Exercise: Using JMS with Spring