Course Outline

Java Spring Bootcamp

Course Summary

Description

This course consists of two parts, Core Java and Spring Boot

In the first part, the Java Bootcamp provides a comprehensive introduction to Java. It is suitable for programmers with good working programming experience (no Java experience needed). In addition to covering basic Java programming, the course provides a solid understanding of the core OO and Java concepts and practices needed to create well-designed Java programs. This includes creating well-designed Java classes, encapsulation, composition, and inheritance/interfaces.

It then moves on to comprehensive coverage of more advanced topics in Java and OO development to provide participants with a strong grounding to use Java in a sophisticated and productive manner. This includes in-depth coverage of functional programming with lambdas and streams, as well as the Java Platform Module System (JPMS). Java modules present a fundamental shift in how applications are organized and interconnected to the libraries they use. Migration strategies are also covered, including a step-by-step case study.

Java Enterprise Edition (JEE) is a powerful platform for building web and database-driven applications. This course provides the information you need to design and build your data-driven web applications. You'll learn the details of the core JEE Web and database technologies and how to use them together. Then the course covers everything you need to know to begin working with the Java Persistence API in a very short time. It covers all the important concepts necessary to access and update data stored in relational databases.

In the second part, Spring 5 provides an evolutionary advance of Spring's powerful capabilities. This course introduces the many Spring Core capabilities, as well as provides guidelines on when and how to use them. It also goes into considerable depth on Spring Boot for dependency management and auto-configuration, as well as Spring REST for creating RESTful resources. This course utilizes Spring Boot's easy configuration and auto-configuration wherever possible. "Classic" Spring configuration (usually more verbose and complicated) is optionally covered in abbreviated form.

The course starts with in-depth coverage of Spring's Core module to reduce coupling and increase the flexibility, ease of maintenance, and testing of your applications. It goes on to cover many of the most important capabilities of Spring, including easing configuration with Spring Boot, integrating JPA persistence layers with Spring and Spring Data, and using Spring's declarative transaction capabilities.

The course includes a solid introduction to Spring REST and coverage of building RESTful resources. It also covers many of the details of Spring Boot, including how to create Boot-based POMs (maven) for simplified dependency management, customizing Boot behaviour, and understanding/managing Boot's auto-configuration.

The course also covers useful technologies provided by Spring and Spring Boot. It goes beyond the technologies of core Spring to cover some of the more interesting and useful capabilities Spring and Spring Boot provide.

ProTech Professional Technical Services, Inc.

Course Outline

Topics

- Fast Track to Java 17 and Development – 5 days
- Intermediate/Advanced Java – 5 days
- JEE Web Development – 2 days
- Java Persistence API (JPA) – 3 days
- Introduction to Spring 5, Spring Boot and Spring Rest – 5 days
- Advanced Spring 5 and Spring Boot – 5 days

Audience

This course is intended for participants who have prior programming experience.

Duration

25 days

Week One Objectives

- Understand Java's importance, uses, strengths and weaknesses
- Understand the release cycle and Long Term Support (LTS) releases
- Understand Java language basics
- Write, compile, and run Java programs
- Use the Java shell (JShell - Java 9+) for interactive programming
- Understand the Object Model and Object Oriented Programming
- Understand and use classes, inheritance, polymorphism
- Create well designed classes and use them in your Java programs
- Use composition and delegation to create objects from other objects
- Understand & use packages to organize code
- Understand and use Java 9 modules
- Understand interfaces, their importance, and their uses
- Use interfaces to implement abstraction
- Learn good Java coding style
- Create well structured Java programs
- Compile and execute programs with the JDK development tools and with an Integrated Development Environment (IDE) of your choice
- Use the core Java libraries (java.lang, java.util)
- Understand & use exceptions for error handling
- Understand the basics of using JDBC and JPA, and use them to access databases from Java
- Use the Java Collections Framework including new API introduced in Java 9-11
- Use other new features such as type inference
- Be aware of, and use the new features of Java 11-17, as well as important advanced features of earlier Java versions
- Understand and use basic I/O streams (Optional)

Week Two Objectives

- Solidify Java foundational knowledge, including the important contracts of class Object
- Understand the uses and consequences of inheritance and composition, and reinforce the role of interfaces
- Reinforce fundamental OO principles such as cohesion, coupling, and polymorphism
- Use the JUnit testing framework and become fluent in writing assertions to verify correct program behavior
- Familiarity with UML modeling in class diagrams and sequence diagrams
- Use advanced techniques for object creation, including factories and singletons
- Use established design patterns for object composition, including Strategy, Decorator, and Facade
- Write and use generic classes and methods
- Learn the use cases for inner classes and refactor existing code to use them when appropriate
- Create and use custom annotations
- Be familiar with reflection and how to use it
- Understand the role of functional interfaces
- Understand lambda expressions and method references, and use them to pass behavior (methods)
- Use the Stream API to perform complex processing of collections and other input sources
- Create and use Java modules, understanding module descriptors, modular JARs, exports and dependencies, and the modulepath
- Understand the structure and behavior of the modular JDK, and how it supports modular applications as well as legacy classpath-based code
    - Migrate classpath-based applications to Java 11, understanding the stages of migration and options available

Week Three Objectives

- Design and build robust and maintainable web applications
- Create dynamic HTML content with Servlets and Java ServerPages, using the JSP Standard Tag Library (JSTL)
- Make Servlets and JSP work together cleanly
- Use JSTL and other Custom Tag Libraries to separate Java and HTML code
- Access databases with JDBC
- Integrate the Data layer with the Web layer
- Understand the JPA architecture
- Create JPA based applications
- Understand and use JPA to map persistent objects to the database
- Create JPA persistence units, and understand JPA persistence contexts and the Entity Lifecycle
- Use the JPA EntityManager
- Work with queries and JPQL (Java Persistence Query Language), as well as the Criteria API (JPA 2)
- Understand and work with collections & associations
    - Value and Entity Types
    - Bidirectional and unidirectional
    - 1-1, 1-N, N-N
- Use versioning support for optimistic locking
- Map inheritance hierarchies using JPA
- Performance tune your JPA applications
- Understand JPA transaction support
- Understand the relationship between JPA / EJB3
- Use JPA entities from session beans (Optional)
- Be familiar with Spring support for JPA (Optional)

Week Four Objectives

- Understand the core principles of Spring, and of Dependency Injection (DI) / Inversion of Control
- Use the Spring Core module and DI to configure and wire application objects (beans) together
- Know the different types of metadata (XML, annotations/@Component, and Java Configuration/@Configuration), and how and when to use them
- Understand and use the complete capabilities of the Core module, such as lifecycle events, bean scopes, and the Spring API
- Use Spring Boot to simplify dependency management and configuration
- Understand and use Boot's auto-configuration
- Customize Boot's behavior with properties and in other ways
- Work with the ORM (Object-Relational Mapping) module to integrate Spring with technologies such as JPA
- Use Spring Data to automatically generate JPA-based repository classes
- Understand and use Spring's transaction support, including the easy-to-use Java annotation support
- Understand REST, and use Spring REST to build RESTful services
- Use Ajax-based front ends with Spring REST
- Use RestTemplate to create Java REST clients

Week Five Objectives

Gain a working knowledge of the following technology areas using Spring and Spring Boot.

- JDBC with Spring and JdbcTemplate
- Spring JMS
- Spring AOP
- Additional capabilities of Spring Security
- Reactive Spring and WebFlux
- Overview of Spring Batch
- Overview of Spring Cloud