



Master in Java with Springboot

About DevOpsSchool

DevOpsSchool is a unit of "Cotocus PVT Ltd" and a leading platform which helps IT organizations and professionals to learn all the emerging technologies and trend which helps them to learn and embrace all the skills, intelligence, innovation and transformation which requires to achieve the end result, quickly and efficiently. We provide over 40 specialized programs on DevOps, Cloud, Containers, Security, AI, ML and on Big data that are focused on industry requirement and each curriculum is developed and delivered by leading experts in each domain and aligned with the industry standards.

About Course

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on various platforms, such as Windows, Mac OS and various versions of Unix. Generally a language is either compiled or interpreted, but Java is both compiled as well as interpreted. Java is a true object-oriented language, which provides a platform to develop an effective and efficient application and program real life complexities.

Spring Boot is an open-source Java-based framework used for creating microservices. It is developed by Pivotal Team and is used to build stand-alone and production-ready Spring applications.

This tutorial has been prepared for beginners to help them understand basic to advanced concepts related to Java programming language. This course then will teach you how to write industrial-strength object-oriented code. It covers the topics that the language courses.



Co-coordinator - Akanksha Kumari

Call/WhatsApp: - +91 1800 889 7977

Mail Address: -

contact@DevOpsSchool.com

Secondary contact - Patrick

Call/WhatsApp: - +91 7004 215 841

Mail Address: - contact@DevOpsSchool.com

Duration	20 Hours
Mode	Online (Instructor-led, live & Interactive)
Projects (Real time scenario based)	1

FEATURES	DEVOPSSCHOOL	OTHERS
Faculty Profile Check	✓	✗
Lifetime Technical Support	✓	✗
Lifetime LMS access	✓	✗
Top 25 Tools	✓	✗
Interviews Kit	✓	✗
Training Notes	✓	✗
Step by Step Web Based Tutorials	✓	✗
Training Slides	✓	✗
Training + Additional Videos	✓	✗

AGENDA OF THE MASTER IN JAVA WITH SPRINGBOOT

Introduction - Java

- Java Fundamentals - Object-oriented Design

Thinking in Objects

- Basics
- Coherence, Coupling, and God Classes
- What Is an Object?

Defining Behavior and State

- Problems with println()
- Money and Delegation: Getters and Setters Are Evil

Extending Classes

- The Is-a Test
- Simple Extension and Template Method
- Delegation Under Inheritance: Shape
- Liskov Substitution: Stack, Ver. 1
- Fragile Base Classes: Stack, Ver. 2

Defining Class Contracts with Interfaces

- Defining Contracts
- Fixing Fragility
- Dependency Inversion

Working Without Getters and Setters

- The Open Closed Principle
- Builder

Testing OO systems

- Dependency Injection
- Summing Up

What Is Object Orientation?

- Course Overview
- What is Object Orientation?
- From Use Cases to Conceptual Classes
- Class Relations
- Deriving Software Classes, Part 1
- Deriving Software Classes, Part 2
- Demo: Instance and Static Methods

Encapsulation and Abstraction

- Module Overview
- More Class Relationships
- Encapsulation
- Assigning Responsibilities
- Demo: Exercising the System
- Demo: the Single Responsibility Principle
- Demo: Defending Encapsulation
- Abstraction: Encapsulation or Inheritance?

Inheritance and Polymorphism

- Module Overview
- Don't Repeat Yourself
- The Liskov Substitution Principle
- Overriding and Dynamic Dispatch
- Demo: Dynamic Dispatch
- Using Overridden Methods
- Understanding Inheritance
- Abstract Classes and the Open-closed Principle

Interfaces, Composition, and System Design

- Module Overview
- Interfaces What Are They, and Why Do We Need Them?
- Demo: Paying through an Interface
- Demo: The Fragile Base Class Problem
- Delegation and the Decorator Pattern
- The Strategy Pattern
- The Interface Segregation Principle
- The Dependency Inversion Principle

Introducing - Spring Boot

- Getting Started
- Learning Path
- Spring Boot Overview
- Features of Spring Boot
- Summary



Bootstrapping a Simple Application

- Overview
- Demo: Spring Initializr
- Demo: Spring Boot CLI
- Demo: Auto Configuration
- Spring Boot Annotations
- Spring Boot Properties
- Spring Boot Profiles
- Summary

Accessing Data with Spring Boot and H2

- Overview
- H2 Database
- Why Use Templating Engines?
- ORM with JPA
- Entities
- Demo: H2 Integration
- Demo: H2 Console
- Demo: Partials
- Summary

Configuring a Spring MVC Application with Spring Boot

- Overview
- MVC Design Pattern
- Demo: MVC in Action
- Packaging and Deployment
- Demo: Uber Jar
- Summary

Building a RESTful Web Application with Spring Boot

- Overview
- REST Architecture
- Demo: RESTful Web Application
- Demo: Testing with URL
- Response Formats
- Exception Handling
- Demo: Exception Handling
- Summary



Building a GraphQL Server with Spring Boot

- Overview
- GraphQL Overview
- GraphQL Dependencies
- GraphQL Schemas
- Demo: Schemas
- Query Operations
- Mutations
- Exceptions
- Demo: GraphiQL
- Summary

Enabling Actuators, Metrics, and Health Indicators

- Overview
- Actuator
- Custom Endpoints
- Summary

Enabling Actuators, Metrics, and Health Indicators

- Overview
- Actuator
- Custom Endpoints
- Summary

Testing with Spring Boot

- Overview
- Testing Overview
- Unit Testing
- Integration Tests



Thank you!

Connect with us for more info

Call/WhatsApp: - +91 968 682 9970

Mail: -

contact@DevOpsSchool.com

www.DevOpsSchool.com