Design Patterns & Java8 & Spring intro

40 Hours

Course Overview:

This course discovers some of the most useful, important and common design patterns in Java; reusable code patterns that you can use to structure your program and perform common tasks. In addition, the course will teach you about new features of Java 8 and the Spring framework to gain experience of better development using spring.

This course includes important advanced Java topics such as reflection, multithreading, TDD concepts and Maven and functional programming with lambdas and streams. It teaches a number of useful techniques that enhance productivity and good system design - which may otherwise take Java developers years to absorb on their own.

Who Should Attend:

The course is hands on, and requires that students be comfortable with writing general Java code at an intermediate level, including the use of interfaces.

Required Skills:

At least 2 years of Java programming experience

Course Contents:

Design patterns

- Anti-patterns -
- OOP principles and best practice: SOLID
- Strategy
- Singleton
- Factory
- Builder (and immutable objects)
- Proxy
- Decorator
- Chain of responsibilities
- IOC

Reflection

- ClassLoaders
- Classpath
- Class / Method / Constructor / Field
- Annotations
- org.reflections
- How to scan packages
- Dynamic proxy
- Invocation handler
- CGLIB

New in Java 8

- Lambda / functional programming
- Effectively final
- Static and default methods in interfaces

Introduction to Spring

- Context
- Spring bean
- Xml
- Spring Annotations support
- Bean Definition
- BeanPostProcessor

Multithreading

- Threads
- Runnable / Callable
- How to stop threads correctly
- Synchronization / volatile / locks
- Wait / notify
- Executors

Lombok

- Annotation Processors
- Pojo annotations
- Functional annotations
- Delombok

TDD main concepts

- JUNIT
- Mockito
- Spring Test

Maven

- Maven & Procedural Build Tools
- The POM
- The Build Lifecycle
- Standard Project Layout
- Running Maven
- Artifacts & Dependencies & Dependency Management
- Repositories
- POM Inheritance (parent)
- Cross-project Configuration (reactor)
- Profiles
- Installation and Deployment
- Plugins
    - dependency:…
    - enforcer
    - writing custom plugin
- Lifecycle and Packaging