

Course description

This practical four-day course is aimed at developers who rapidly need to gain the skills to develop robust commercial applications, using the latest features of the Java Standard and Enterprise Editions. Following the fundamentals, a test driven development approach is used to assemble database-connected web and rich client applications.. The Hibernate object-relational mapping framework is used to implement the data layer with JavaServer Faces and JavaFX for the presentation layers. Build automation with Gradle and revision control with Git are also employed. Other topics include the collections classes, lambda expressions and concurrent programming. Design principles and patterns are discussed in the context of the various topics. Course hours are 9:00 to 5:00 at air-conditioned offices in Reading and each delegate has the use of a computer.

What you will learn

- Fundamentals of object-oriented programming in Java
- Test Driven Development including mock objects
- Distributed revision control with Git and build automation with Gradle
- Lambda expressions
- Object-relational mapping with Hibernate and JPA
- Web applications and JavaServer Faces
- Enterprise beans and Dependency Injection
- Desktop applications with JavaFX
- Concurrent and asynchronous programming
- The Java platform module system
- Design principles and patterns

Who should attend

Object-oriented developers who are looking to update their Java skills with current techniques and APIs

Prerequisites

Previous use of a programming language and some familiarity with the concepts of object-oriented programming is recommended

In-house training

In addition to scheduled public courses, we can deliver the course as customised on-site training for up to 8 delegates, for a fee of £3900+VAT. This includes all expenses and the use of laptops, if required.

Contact

To discuss training options or to make a booking, please telephone 0118 966 4994 or email mailbox@javaconsult.co.uk

Course instructor



Originally a Civil Engineer with a BSc from London University the course instructor, Simon Dineen, has 15 years' experience in .NET training and development.

Course outline

1. [Java Platform Overview](#)

- Overview of the Java Development Kit and the Java and the Java Runtime Environment
- The Development Environment (Eclipse or IntelliJ)
- Other tools and frameworks including Gradle, Git, Hibernate, JUnit and Mockito

2. [Java Syntax Review](#)

- Defining a class with fields and methods
- Operators and precedence
- Strings and immutable types
- Mutable types and the StringBuilder class
- Conversion, casting and autoboxing
- Declare arrays of primitive or class types
- Conditionals and loops
- Catching and throwing exceptions
- Enumerated types

3. [Class Design](#)

- UML class diagrams and sequence diagrams
- Packages and encapsulation
- Constructors and method overloading
- Inheritance compared with composition
- Invoke a specific base class constructor
- Method overriding and Polymorphism
- Abstract classes and interfaces

4. [Build automation with Gradle](#)

- The standard directory structure
- Build lifecycle, plugins and tasks
- Dependency management

5. [Distributed revision control with Git](#)

- Initializing a repository and cloning an existing repository
- Indexing and committing changes
- Viewing the commit history and comparing versions of a file

- Creating and merging branches
- Pushing a branch to a remote server

6. Test-driven development

- Writing unit tests with “arrange, act, assert” phases
- The “Red, Green, Refactor” development cycle
- Parameterized tests with the JUnitParams library
- Interactions testing with spies and stubs using the Mockito library
- Writing integration tests and including these in the Gradle build

7. Collections

- Using generic collections including Sets, Lists, and Maps
- Writing interfaces with default and static methods
- Functional interfaces, anonymous inner classes and lambda expressions

8. Databases

- Using JDBC to connect to a MySQL database and execute SQL expressions
- Writing try-with-resources blocks to ensure connections are closed
- Using the Hibernate object-relational mapping library to store and retrieve objects
- Mapping associated classes to related tables with the Java Persistence API

9. Streams

- Building Byte Streams and Character Streams with the java.io and java.nio packages
- Serializing objects to a file

10. Web applications

- Java EE7 containers and the Glassfish application server
- Writing a Servlet and understanding HTTP methods
- Build and deploy a web archive to Glassfish with Gradle
- The JavaServer Faces framework, facelets and expression language
- CDI backing beans and scopes

11. Enterprise beans

- Deploying stateless and stateful session beans
- Using dependency injection to invoke enterprise beans

12. Web services

- Deploying a JAX-RS service
- Producing and consuming JSON in response to HTTP GET and POST requests
- Using an HTTP debugger to test the service

13. JavaFX desktop applications

- Building a user interface from a hierarchical graph of nodes and handling events
- Understanding JavaFX properties and establishing bindings between properties
- Connecting a TableView component to the data layer built previously

14. Concurrency

- Building multi-threaded code with Executors and the Callable and Future interfaces
- Using thread safe collections and the AtomicInteger class
- Parallel processing with lambda expressions
- Building asynchronous computations with chained CompletionStages
- Running test methods in parallel with the tempus-fugit library
- Updating JavaFX components from a worker thread

15. The Java platform module system

- The Java 9 module concept
- Naming modules, requiring modules and exporting packages
- Automatic modules and the unnamed module
- Modular JARs

16. Design principles and patterns

- A number of principles and patterns are discussed during the course including the Strategy, Factory Method, Abstract Factory, Singleton, Decorator and Observer patterns and the design principles on which these are built