



☎ +44 7989 401397

✉ info@olsensoft.com

## Java SE 12 Advanced Techniques

(5 days)

### Course overview

This course takes a deeper look into some of the advanced techniques and APIs available in contemporary Java, focussing specifically on Java 8 through Java 12. You'll learn all about the new modular approach introduced in Java 9, as well as a thorough investigation of functional programming techniques via lambdas, functional interfaces, and streams. The course also covers important parts of the API including concurrency, data manipulation, and security.

You can use this course as part of your preparation for Oracle exam 1Z0-816, Java SE 11 Programmer II.

### What you'll learn

- Modular Java
- Functional programming techniques
- Effective concurrency
- Data access techniques
- Security
- Localization

### Prerequisites

- At least 6 months experience as a Java developer

### Course details

- **Java Class Fundamentals:** Final classes; Overview of inner classes; Regular inner classes; Method-local inner classes; Anonymous inner classes; Static nested classes; Enumerations
- **Lambda expressions:** What are lambda expressions; Formal syntax for lambda expressions; Lambda expression simple syntax; Lambda expressions that return a value; Lambda expressions with multiple arguments; Lambda expressions and scope; Common usage scenarios
- **Method enhancements:** Method references; Constructor references; Default methods; Static methods in interfaces
- **Functional interfaces:** Function; UnaryOperator; BiFunction; BinaryOperator; Predicate; Consumer; Supplier; Additional new functional interfaces
- **Streams:** Sequential vs. parallel streams; Immediate vs. terminal operations; Stream example; Lazy evaluation; A closer look at immediate and terminal operations; Primitive-specialized streams
- **Modular Programming in Java:** Getting started with Java modules; Using the modular JDK; Creating a modular application; Creating a multi-module app; Modularized JARs

- [Going Further with Java Modules](#): Standalone applications; Services; The unnamed module; Automatic modules; Migration Strategies
- [Concurrency](#): Runnable and Callable classes; Java synchronization language features; Designing thread-safe classes; Recommendations for synchronizing resource access
- [Additional Multithreading Issues](#): Using concurrent collections; Using synchronizers and locks; Thread pooling techniques; Using the executor framework; Using pooling effectively
- [File Handling](#): Overview of working with files; Reading and writing text files; Reading and writing binary files; Serialization; Overview of NIO2; Basic operations; File visitors; Directory watching; File attributes
- [Accessing Databases using JDBC](#): JDBC drivers and connections; Statements and results; Obtaining metadata; DataSources; Metadata; JDBC escape syntax; Transaction management; Additional techniques
- [Annotations and Reflection](#): Understanding annotations; Using standard annotations; Defining custom annotations; Overview of reflection; Dynamic retrieval of information using reflection; Creating instances of extensibility objects
- [Security](#): Security managers; Policy files; Sandboxing; Cryptographic random numbers;
- [Localization and Resource Bundles](#): Locales; Resource bundles; Locale-specific formatting and parsing