



☎ +44 7989 401397

✉ info@olsensoft.com

Java SE Programming

(5 days)

Course overview

This is our mainstream course for developers wishing to get up to speed with Java. You will learn how to implement object-oriented applications using Java, and how to make use of common APIs in the Java SE library. You'll also get an introduction to modules, a major new feature that will impact the way we all write Java code going forward.

You can use this course as the first part of your preparation for Oracle exam 1Z0-819, Java SE 11 Developer. To complete your preparation for this course, you can also attend our Java SE Advanced Techniques course.

What you'll learn

- Writing clean Java code
- Implementing object-oriented solutions in Java
- Using core Java SE classes
- Handling exceptions and assertions
- Working with files and databases
- Using multithreading and concurrency
- Introduction to modules

Prerequisites

- At least 6 months programming experience
- Familiarity with OO concepts would be an advantage

Course details

- **Getting Started with Java:** Java goals and principles; Installing and using the Java SE; JAR files; Using an IDE
- **Java Language Fundamentals:** Basic syntax rules; Defining classes; Defining and using packages; Declaring and using variables; Useful Java classes; Wrapper classes
- **Operators and Flow Control:** Survey of common operators; Decision making; Looping constructs; for-each style loops
- **Defining and Using Classes:** Essential concepts; Defining a class; Creating and using objects; Initialization; Using the static keyword
- **Arrays:** Declaring and using arrays; Traversing arrays; Using the Arrays class; Multi-dimensional arrays
- **Additional Language Features:** Autoboxing / unboxing; Varargs; Type-safe enumerations; Static imports

- [Useful Java SE Classes and Techniques](#): The Console class; The StringBuilder class; Formatting techniques; Regular expressions
- [Inheritance](#): Overview of inheritance; Defining superclasses and subclasses; Polymorphism; Abstract classes and methods; Final classes and methods
- [Interfaces](#): Overview of interfaces; Defining and implementing interfaces; Using interfaces in client code
- [Collections and Generics](#): Overview of collections and generics; The need for generics; Using collections; Defining generic classes; Defining generic methods
- [Exceptions and Assertions](#): Overview of exception handling; Throwing and catching exceptions; Standard exception classes; Defining new exception classes; Working with assertions
- [File Handling](#): Overview of working with files; Reading and writing text files; Reading and writing binary files; Serialization
- [Multithreading](#): Creating multiple threads; Synchronizing threads; Synchronization classes; Concurrency API
- [Accessing Databases using JDBC](#): JDBC drivers and connections; Statements and results; Obtaining metadata; Additional techniques
- [Working with Modules](#): Overview of Java modules; Understanding how the JDK is modularized; Declaring modules; Using other modules