

Java and OO Development

Course Summary

Description

Fast Track to Java and OO Development provides a comprehensive introduction to Java. It is suitable for programmers with good working programming experience (no Java experience needed). It can be taught to audiences with less experience by reducing coverage of the optional, more advanced, topics.

In addition to covering basic Java programming, the course provides a solid understanding of the core OO and Java concepts and practices needed to create well-designed Java programs. This includes creating well-designed Java classes, encapsulation, composition, and inheritance/interfaces.

The material and labs include coverage of useful new Java capabilities such as the Java 9+ module structure, and other recent features and API enhancements. It also introduces important APIs such as the Java Collections Framework, and JDBC/JPA. This course is suitable for environments using Java 11-. It also covers details of the Java release cycle, and Long Term Support (LTS) releases.

Be prepared to work hard and learn a great deal! The course can be held on-site & customized to fit your needs.

Objectives

At the end of this course, students will be able to:

- Understand Java's importance, uses, strengths and weaknesses
- Understand the release cycle and Long Term Support (LTS) releases
- Understand Java language basics
- Write, compile, and run Java programs
- Use the Java shell (JShell - Java 9+) for interactive programming
- Understand the Object Model and Object Oriented Programming
- Understand and use classes, inheritance, polymorphism
- Create well designed classes and use them in your Java programs
- Use composition and delegation to create objects from other objects
- Understand & use packages to organize code
- Understand and use Java 9 modules
- Understand interfaces, their importance, and their uses
- Use interfaces to implement abstraction
- Learn good Java coding style
- Create well structured Java programs
- Compile and execute programs with the JDK development tools and with an Integrated Development Environment (IDE) of your choice
- Use the core Java libraries (java.lang, java.util)
- Understand & use exceptions for error handling
- Understand the basics of using JDBC and JPA, and use them to access databases from Java
- Use the Java Collections Framework including new API introduced in Java 9-11
- Use other new features such as type inference
- Be aware of, and use the new features of Java 11-, as well as important advanced features of earlier Java versions
- Understand and use basic I/O streams (Optional)

Java and OO Development

Course Summary (cont'd)

Topics

- Java Overview
- Class and Object Basics
- More on Classes and Objects
- Flow of Control
- More about Classes and Objects
- Strings, Arrays, and Dates/Times
- Packages and Modules
- Composition and Inheritance
- Interfaces
- Exceptions
- Java Collections and Generics
- Database Access with JDBC and JPA
- I/O Streams (Optional)
- Additional Language Features (Optional)

Audience

This course is suitable for programmers with good working programming experience (no Java experience needed). It can be taught to audiences with less experience by reducing coverage of the optional, more advanced, topics.

Prerequisites

Working knowledge of some programming language - no Java experience needed

Duration

Five days

Java and OO Development

Course Outline

- I. *Java Overview*
 - A. Language and Platform Features
 - B. A Simple Java Program
 - C. The Java Release Cycle
 - D. Program Lifecycle
 - E. The Java SE Development Kit (JDK)
 - F. LABS:
 - G. Hello World: A Simple Application
- II. *Class and Object Basics*
 - A. The Object Model and Object-Oriented Programming
 - B. Classes, References, and Instantiation
 - C. Adding Data to a Class Definition
 - D. Adding Methods (Behavior)
 - E. LABS:
 - F. Exploring Types and Object Instances
 - G. Introducing your IDE
 - H. Writing and Using a Class Definition with Fields and Methods
- III. *More on Classes and Objects*
 - A. More about Methods
 - B. Encapsulation and Access Control, public and private Access
 - C. Constructors and Initialization
 - D. static Members of a Class
 - E. LABS:
 - F. Accessor Methods
 - G. Encapsulation / Access Protection
 - H. Writing and Using Constructors
 - I. (Optional) Static Members
- IV. *Flow of Control*
 - A. Branching: if, if-else, switch
 - B. Iteration: while, do-while, for, break, continue
 - C. LABS:
 - D. Data Validation
- V. *More about Classes and Objects*
 - A. Type-safe Enums
 - B. Wrapper Classes
 - C. Java 8+ Date/Time Support
 - D. Working with References
 - E. Formatted Output
 - F. LABS:
 - G. Using enums
 - H. Working with Dates and Times
- I. Using the Debugger
- VI. *Strings, Arrays, and Dates/Times*
 - A. String, StringBuffer, StringBuilder
 - B. Arrays, Primitive Arrays, Arrays of Reference Types
 - C. varargs
 - D. LABS:
 - E. Using Strings and Arrays
- VII. *Packages and Modules*
 - A. Package Overview - Using Packages to Organize Code
 - B. Creating Packages, package Statement, Required Directory Structure
 - C. import statements
 - D. Classpath
 - E. Java Modules Overview
 - F. Defining Modules, Requires, and Exports
 - G. Module Path and Classpath - Differences and Coexistence
 - H. LABS:
 - I. Using Packages
 - J. Using Java Modules
- VIII. *Composition and Inheritance*
 - A. Using Composition to Deal With Complexity
 - B. Composition/HAS-A, Delegation
 - C. Using Inheritance to share commonality
 - D. IS-A, extends Inheriting Features
 - E. Constructor Issues
 - F. Overriding Methods, @Override, Using Polymorphism
 - G. Abstract Classes
 - H. LABS:
 - I. (Optional) Working with Composition
 - J. Using Inheritance
 - K. Polymorphism
 - L. [Optional] Polymorphism
- IX. *Interfaces*
 - A. Defining and Implementing Interfaces
 - B. Using Interfaces
 - C. Default Methods and static Methods
 - D. LABS:
 - E. Using Interfaces to Remove Implementation Dependencies
 - F. Using Default and Static Methods

Java and OO Development

Course Summary (cont'd)

X. *Exceptions*

- A. Exceptions and the Exception Hierarchy
- B. Throwing Exceptions and Checked Exceptions
- C. try and catch
- D. Handling Exceptions
- E. try and catch
- F. Program Flow with Exceptions
- G. Multicatch, finally, try-with-resources
- H. LABS:
- I. Throwing Exceptions
- J. Checked Exceptions

XI. *Java Collections and Generics*

- A. The Collections Framework and its API
- B. Collections and Java Generics
- C. Collection, List, Set, Map
- D. Autoboxing
- E. Collections of Object (non-generic)
- F. Using ArrayList, HashSet, and HashMap
- G. for-each Loop
- H. Processing Items With an Iterator
- I. [Optional] More About Generics
- J. LABS:
- K. Using Collections and Generics

XII. *Database Access with JDBC and JPA*

- A. JDBC Overview
- B. JDBC Defined Types
- C. Example of JDBC Usage
- D. JPA Overview
- E. JPA Architecture and Programming View
- F. Mapping Entity Classes with Annotations
- G. Persistence Unit, EntityManagerFactory, and EntityManager
- H. Working with JPA (Find by primary key and inserts)
- I. LABS:
- J. Mapping an Entity Class
- K. Using JPA
- L. Insert/Query Demo

XIII. *I/O Streams (Optional)*

- A. I/O Streams Overview
- B. Readers and Writers
- C. Exception Handling
- D. Byte Streams
- E. Working with Files
- F. NIO, Paths, Files
- G. LABS:
- H. Reading and Writing Files
- I. Using Byte Streams (Optional)

XIV. *Additional Language Features (Optional)*

- A. Functional Interfaces and Lambda Expressions
- B. Switch Expressions
- C. Other Java Features
- D. LABS:
- E. Working with Lambdas (Demo)