

# Fast Track to Core Java Programming for OO Experienced Developers - TT2104

Gain the latest skills, techniques and best practices required to create robust, well designed and efficient modern Java applications

**Duration:** 4 Days

**Skill Level:** Introductory

**Available Format:** Instructor-Led Online; Instructor-Led, Onsite In Person ; Blended;  
On Public Schedule

**Fast Track to Java Programming for OO Developers** is a four-day, fast-paced hands-on course that helps experienced object oriented developers (coming from C++, C#, etc.) quickly get up and running with programming in Java, using the latest coding skills. You'll exit the course well-prepared to tackle real-world programming tasks, equipped with an in-depth knowledge of Java and a practical skill set to create robust, efficient, and secure Java applications.

## What You'll Learn

### Overview

**Fast Track to Java Programming for OO Developers** is a four-day, fast-paced hands-on course that helps experienced object-oriented developers (coming from C++, C#, etc.) quickly get up and running with programming in Java, using the latest coding skills. You will exit the course well-prepared to tackle real-world programming tasks, equipped with an in-depth knowledge of Java and a practical skill set to create robust, efficient, and secure Java applications.

Working in a skills-immersive, hands-on learning environment led by our expert instructor, you will explore the intricacies of the Java platform and the Java Standard Edition, working seamlessly with the Java Development Kit (JDK), and using the development environment (IDE) proficiently. The course emphasizes best practices in OO development, reinforcing core object-oriented principles such as inheritance,

polymorphism, interfaces, and abstract classes, allowing you to design software applications, develop back-end systems, or contribute to enterprise-level projects.

The course centers around applying these skills to real-world challenges, ensuring you can leverage them in your role or project right after class. Throughout the course you will build applications, create subclasses, and develop custom exception handling. You will also gain proficiency in using Java's arrays, records, packages, and utility classes. The course also delves into the advanced features of Java programming, including working with generics, understanding functional programming with lambda expressions, and using the Collection API. You will be proficient in the JDK, be able to navigate your course IDE, and will be able to develop complex Java programs.

You will exit this course equipped with the skills required to tackle a variety of programming challenges and create efficient, robust Java applications with ease.

**NOTE:** This course uses Java 21, which also covers the fundamental concepts and techniques in Java 11 and 17. This course is suited for Java 11, Java 17 and Java 21 skills development. Earlier versions are available. Please inquire for options.

## Objectives

This course is rich with hands-on activities and examples, combining robust real-world hands-on labs with expert instruction, engaging activities and group discussions and review. You'll learn and practice new skills under the guidance of our expert instructor, who will prepare you to apply these in the job, role or project with confidence.

### Throughout the course you'll:

- Understand the fundamentals of the Java platform, its lifecycle, and the responsibilities of the Java Virtual Machine (JVM), enabling you to create efficient and reliable Java applications.
- Gain proficiency in using the JDK, including navigating its file structure, utilizing the command-line compiler, and executing Java applications, ensuring a smooth development process.
- Master the IDE, including its interface, project management, and module creation, to enhance productivity, collaboration, and overall development workflow.
- Develop solid skills in writing Java classes, defining instance variables, creating object instances, and implementing main methods, forming a strong foundation in Java programming.

- Acquire expertise in adding methods to Java classes, writing constructors, and leveraging the 'this' keyword, allowing you to create more sophisticated and customizable Java applications.
- Comprehend and apply core object-oriented programming concepts, such as encapsulation, inheritance, and polymorphism, to create modular, maintainable, and reusable code.
- Enhance your knowledge of Java language statements, including arithmetic, comparison, and logical operators, as well as loops and switch expressions, to develop more complex and efficient Java applications.
- Learn to effectively handle exceptions, create custom exception classes, and use try/catch blocks to ensure the robustness and reliability of your Java applications, minimizing potential runtime issues.
- Explore and understand the Java Modular System and its impacts on accessibility and visibility.
- Gain proficiency in working with collections in Java, which includes learning about the different collection implementations (Set, List, and Queue), using iterators, and sorting collections. This will enable you to manage data effectively in your Java programs.

**Specific Java 17 features include:** Switch Expressions, Text blocks, Pattern matching for instanceof, Introduce records as carrier of immutable data

**Specific Java 21 features include:** Sequenced Collections, Pattern matching in Switch statements, Record Patterns

If your team requires different topics, additional skills or a custom approach, our team will collaborate with you to adjust the course to focus on your specific learning objectives and goals.

effectively in your Java programs.

## Audience

This course is designed to quickly help experienced programmers transition their existing object oriented development skills (from C++, C# .Net, etc.) to Java.

**This course is not for non-OO developers, very new or non-developers.**

## Pre-Requisites

To ensure a smooth learning experience and maximize the benefits of attending this course, you should have **prior hands-on programming experience in another OO programming language such as C# or C++**.

TEST-01                      Mastering Machine Learning Operations (MLOps) and AI Security Boot Camp

## Agenda

Please note that this list of topics is based on our standard course offering, evolved from current industry uses and trends. We will work with you to tune this course and level of coverage to target the skills you need most. Course agenda, topics and labs are subject to adjust during live delivery in response to student skill level, interests and participation.

### 1. The Java Platform

- Introduce the Java Platform
- Explore the Java Standard Edition
- Discuss the lifecycle of a Java Program
- Explain the responsibilities of the JVM
- Executing Java programs
- Garbage Collection
- Documentation and Code Reuse

### 2. Using the JDK

- Explain the JDK's file structure
- Use the command line compiler to compile a Java class
- Use the command line Java interpreter to run a Java application class

### 3. Using the IntelliJ IDE

- Introduce the IntelliJ IDE
- The Basics of the IntelliJ interface
- IntelliJ Projects and Modules
- Creating and running Java applications
- Tutorial: Exploring your IDE ( with IntelliJ 2023.2 (Community Edition) or Eclipse IDE

### 4. Writing a Simple Class

- Write a Java class that does not explicitly extend another class

- Define instance variables for a Java class
- Create object instances
- Primitives vs Object References
- Implement a main method to create an instance of the defined class
- Java keywords and reserved words

## **5. Adding Methods to the Class**

- Write a class with accessor methods to read and write instance variables
- Write a constructor to initialize an instance with data
- Write a constructor that calls other constructors of the class to benefit from code reuse
- Use the this keyword to distinguish local variables from instance variables

## **6. Language Statements**

- Arithmetic operators
- Operators to increment and decrement numbers
- Comparison operators
- Logical operators
- Return type of comparison and logical operators
- Use for loops
- Switch Expressions
- Switch Expressions and yield

## **7. Using Strings and Text Blocks**

- Create an instance of the String class
- Test if two strings are equal
- Perform a case-insensitive equality test
- Contrast String, StringBuffer, and StringBuilder
- Compact Strings
- Text Blocks
- Unicode support
- Lab: Fun with Strings
- Lab: Using StringBuffers and StringBuilders

## **8. Fields and Variables**

- Discuss Block Scoping Rules
- Distinguish between instance variables and method variables within a method
- Explain the difference between the terms field and variable
- List the default values for instance variables

- Final and Static fields and methods

## **9. Specializing in a Subclass**

- Constructing a class that extends another class
- Implementing equals and toString
- Writing constructors that pass initialization data to parent constructor
- Using instanceof to verify type of an object reference
- Overriding subclass methods
- Pattern matching for instanceof
- Safely casting references to a more refined type

## **10. Using Arrays**

- Declaring an array reference
- Allocating an array
- Initializing the entries in an array
- Writing methods with a variable number of arguments

## **11. Records**

- Data objects in Java
- Introduce records as carrier of immutable data
- Defining records
- The Canonical constructor
- Compact constructors

## **12. Java Packages and Visibility**

- Use the package keyword to define a class within a specific package
- Discuss levels of accessibility/visibility
- Using the import keyword to declare references to classes in a specific package
- Using the standard type naming conventions
- Introduce the Java Modular System
- Visibility in the Java Modular System

## **13. Utility Classes**

- Introduce the wrapper classes
- Explain Autoboxing and Unboxing
- Converting String representations of primitive numbers into their primitive types
- Defining Enumerations
- Using static imports
- Introduce the Date/Time API

- LocalDate / LocalDateTime etc.
- Apply text formatting
- Using System.out.printf

#### **14. Inheritance and Polymorphism**

- Write a subclass with a method that overrides a method in the superclass
- Group objects by their common supertype
- Utilize polymorphism
- Cast a supertype reference to a valid subtype reference
- Use the final keyword on methods and classes to prevent overriding

#### **15. Interfaces and Abstract Classes**

- Define supertype contracts using abstract classes
- Implement concrete classes based on abstract classes
- Define supertype contracts using interfaces
- Implement concrete classes based on interfaces
- Explain advantage of interfaces over abstract classes
- Explain advantage of abstract classes over interfaces

#### **16. Sealed Classes**

- Introduce sealed classes
- The sealed and permits modifier
- Sealed interfaces
- Sealed classes and pattern matching

#### **17. Pattern Matching**

- Pattern Matching in switch statements
- Pattern Matching and sealed classes
- Record Patterns

#### **18. Introduction to Exception Handling**

- Introduce the Exception architecture
- Defining a try/catch blocks
- Checked vs Unchecked exceptions

#### **19. Exceptions**

- Defining your own application exceptions
- Automatic closure of resources
- Suppressed exceptions

- Handling multiple exceptions in one catch
- Enhanced try-with-resources
- Helpful NullPointerException(s)

## **20. Building Java Applications**

- Explain the steps involved in building applications
- Define the build process
- Introduce build scripts
- Explain the standard folder layout
- Resolving project dependencies
- Tutorial: Importing code Using Maven

## **21. Introduction to Generics**

- Generics and Subtyping
- Bounded Wildcards
- Generic Methods

## **22. Introducing Lambda Expressions and Functional Interfaces**

- Understanding the concept of functional programming
- Understanding functional interfaces
- Lambda's and type inference

## **23. Collections**

- Provide an overview of the Collection API
- Review the different collection implementations (Set, List and Queue)
- Explore how generics are used with collections
- Examine iterators for working with collections

## **24. Using Collections**

- Collection Sorting
- Comparators
- Using the Right Collection
- Lambda expressions in Collections
- Sequenced Collections

## **Bonus Topics / Time Permitting**

- Streams
- Understanding the problem with collections in Java
- Thinking of program solutions in a declarative way



- Use the Stream API to process collections of data
- Understand the difference between intermediate and terminal stream operations
- Filtering elements from a Stream
- Finding element(s) within a Stream
- Collecting the elements from a Stream into a List

### Collectors

- Using different ways to collect the items from a Stream
- Grouping elements within a stream
- Gathering statistics about numeric property of elements in a stream

## Follow On Courses

TT3503	Test Driven Development (TDD) and Unit Testing Essentials
TT3320	Core Spring Quick Start   Introduction Spring 6.x and Spring Boot
TT3335	Mastering Spring 5.x Developer Boot Camp
TTAI2300	Quick Start to Prompt Engineering for Software Developers
TTAI2305	Turbocharge Your Code! Generative AI Boot Camp for Developers
TT2211	Intermediate Java Programming   Next-Level Java Skills

## Related Courses

TT2136	Migrating Java 11 to Java 21   Java 21 New Features and Skills
TT2120	Basic Java Programming for Developers New to OO (C, COBOL, etc.)
TT2100	Core Java Programming Developer's Workshop
TT2000	Getting Started with Programming, OO & Java Basics for Non-Developers

**Setup Made Simple!** All of our AI for Business course software, digital course files or course notes, labs, data sets and solutions, live coaching support channels and rich extended learning and post training resources are provided for you in our easy access, single source, no install required online Learning Experience Platform (LXP), remote lab and content environment. Or we can provide a local installation (trial edition) to setup and use on your machine. Access periods and versions vary by course. Please inquire about set up details and options for your specific course of interest. Regardless of setup option, we will collaborate with you to ensure your team is set up and ready to go well in advance of the class.

**Ways to Learn:** At Trivera, we believe that Experience is Everything. Our customizable, hands-on courses are delivered live online, onsite, or in a blended format for maximum flexibility. We provide real-time expert-led training and coaching for all skill levels, from small groups to enterprise-wide programs, ensuring every learner gains the latest, most relevant job-ready skills they can apply with confidence. This course is also available for individuals or small groups on our extensive Public Schedule (see current dates below). We look forward to helping you take the next steps in your modern web developer learning journey.

## For More Information

Please [contact us](#) or call 844-475-4559 toll free for more information about our training services (instructor-led, self-paced or blended), coaching and mentoring services, public course enrollment or questions, partner programs, courseware licensing options and more.