Java Developer SkillJourney Series

Fast Track to Core Java Programming for OO Developers (TT2104)

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

Course Snapshot

- **Course: Fast Track to Core Java Programming for OO Developers (C++, C#) (TT2104)**
- **Duration**: 4 days
- **Audience & Skill Level:** 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.
- **Format / Hands-on:** This course is approximately 50% hands-on, combining engaging instructor-led presentations and practical demonstrations with programming exercises, challenge labs, use case exploration and engaging group activities. Student machines are required.
- **Tools**: 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 available. Please inquire for options.
- **Flexible Delivery Options**: This course can be delivered for your team or organization online-live (virtual), onsite in-person, self-paced or across our immersive blended learning experience platform (LXP).
- **Public Schedule**: This course is currently available on our Public Open Enrollment Schedule.
- **Customizable**: We're flexible! This course agenda, topics, labs, hours and delivery modalities can be adjusted to target your specific training skills objectives, tools and learning goals. Please ask for details.

Description

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.

Working in a skills-immersive, hands-on learning environment led by our expert instructor, you'll 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'll build applications, create subclasses, and develop custom exception handling. You'll 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'll be proficient in the JDK, be able to navigate your course IDE, and will be able to develop complex Java programs.

You'll 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.

Learning Objectives

This course combines engaging instructor-led presentations and useful demonstrations with valuable hands-on labs and engaging group activities. 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 that are covered in the course include:

- Switch Expressions
- Text blocks
- Pattern matching for instanceof
- Introduce records as carrier of immutable data

Specific Java 21 features that are covered in the course 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.

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++.

Related Courses | Java Essentials Suite

The following is a subset of our related basic Java Foundation courses. Please visit www.triveratech.com for the full catalog and complete list.

- TT2000 Getting Started with Programming, OO and Java Basics for Non-Developers (5 days)
- TT2103 JumpStart to Java for OO Developers (such as C++, C#) (3 days)

- TT2104 Fast Track to Java Programming for Experience OO Developers (such as C++, C#) (4 days)
- TT2120 Basic Java Programming for Developers New to OO (such a C, Mainframe) (5 days)
- TT2135 Migrating from Java 11 to Java 17 (1 day)
- TT2136 Migrating from Java 11 to Java 21 (1 day)
- TT2010 Java Developer SkillJourney – Career Experience Program (for Core Java) (multi-week)
- TT2015 Java Full Stack Developer SkillJourney – Career Experience (for full stack / front and back end) (multi-week)

Next Steps / Follow-on Courses: We offer a wide variety of follow-on courses for next-level Java development skills, Java for Web / Full Stack, Jakarta EE, Spring, REST, Microservices, Unit Testing / TDD, Java secure coding, mobile development and more. Please see our **Java Developer Courses, Learning Paths & SkillJourneys** for options based on your specific role and goals.

Enhanced Learning Services: Please ask about our robust Learning Experience Platform (LXP), Skills Assessment & Skills Prep Services, Skills Immersion Programs & Boot Camps, Coaching and Mentoring Services and Extended Learning Support programs.

Setup Made Simple! All applicable course software, digital courseware files or course notes, lab environment, data sets and solutions, live coaching support channels and rich extended learning and post training resources are provided for you in our "easy access, no install required" online **Learning Experience Platform (LXP)**. We'll collaborate with you to ensure your team is set up and ready to go well in advance of the class. Please inquire about set up details and options for your specific course of interest.

Course Topics / Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We'll work with you to tune this course and level of coverage to target the skills you need most. Topics, agenda and labs are subject to change, and may adjust during live delivery based on audience 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

For More Information

For more information about our training services (instructor-led, self-paced or blended), collaborative coaching services, robust Learning Experience Platform (LXP), Career Experiences, public course schedule, partner programs, courseware licensing options or to see our complete list of course offerings, solutions and special offers, please visit us at **www.triveratech.com**, email **Info@triveratech.com** or call us toll free at **844-475-4559.** Our pricing and services are always satisfaction guaranteed.

TRIVERA TECHNOLOGIES • Collaborative IT Training, Coaching & Skills Development Solutions
www.triveratech.com • toll free +1-844-475-4559 • Info@triveratech.com • Twitter TriveraTech

ONSITE, ONLINE & BLENDED TRAINING SOLUTIONS • PUBLIC / OPEN ENROLLMENT COURSES
LEARNING EXPERIENCE PLATFORM (LXP) • COACHING / MENTORING • ASSESSMENTS • CONTENT LICENSING & DEVELOPMENT
LEARNING PLAN DEVELOPMENT • SKILLS IMMERSION PROGRAMS / RESKILLING / NEW HIRE / BOOT CAMPS
PARTNER & RESELLER PROGRAMS • CORPORATE TRAINING MANAGEMENT • VENDOR MANAGEMENT SERVICES

Trivera Technologies is a Woman-Owned Small-Business Firm