Fast Track to Java 11 for OO Experienced Developers (C#, C++, etc.)

JumpStart Your Java 11 Programming Skills with Latest Features & Best Practices for Building Solid Java Applications

Course Snapshot

- **Course: Fast Track to Java 11 for Object Oriented Experienced Developers (TT2104-J11)**
- **Duration**: 4 days
- **Audience & Skill-Level** : Introductory level topics for experienced programmers with practical, current development experience in another OO language such as C++ or C#. This course is not for non or new developers.
- **Focus / Tools**: This course supports Java 11 LTS (Long-Term Support) edition, but is also available for Java 8 LTS or Java 17 LTS. This course is offered using Eclipse IDE but is also available for IntelliJ. Please inquire for details.
- **Hands-on:** This course is approximately 50% hands-on lab to lecture ratio. Student machines are required.
- **Delivery Options**: This course is available for **in-person presentation**, **live online** / **virtual presentation**, or can be presented in a **blended learning** or **short course format**.
- **Public Schedule**: This course has active dates on our open enrollment **Public Schedule**.
- **Customizable**: This course agenda, topics and labs can be further adjusted to target your specific training skills objectives, tools and learning goals. Please ask for details.

Overview

If you're an experienced OO developer (coming from a C# or C++ background, etc.) who needs to transition to programming in Java, this fast-paced, hands-on course will get you there quickly. **Fast Track to Java 11 for OO Experienced Developers** is a four-day, lab-intensive class where you'll quickly be immersed in working with the latest Java 11 programming techniques, using best practices for writing solid, robust (and well-written!) modern object-oriented applications. In addition to learning excellent, current coding skills in Java, you'll explore the new improved features for better performance and new capabilities for addressing rapid application development that Java 11 brings to the table.

This course includes several key aspects that were introduced in Java 9, Java 10 and Java 11 including the Java Modular System, Local Variable Type Inference and several API updates. This course also includes a Quick Look at what's next in Java's latest editions.

Learning Objectives

This "skills-centric" course is designed to train attendees in core OO coding and Java development skills, coupling the most current, effective techniques with the soundest industry practices. Our engaging instructors and mentors are highly experienced practitioners who bring years of current "on-the-job" experience into every classroom.

Working in a hands-on learning environment, guided by our expert team, attendees will learn to:

- Understand not only the fundamentals of the Java language, but also its importance, uses, strengths and weaknesses
- Understand the basics of the Java language and how it relates to OO programming and the Object Model
- Learn to use Java exception handling features
- Work with the Modular system (Project Jigsaw)
- Understand and use classes, inheritance and polymorphism
- Understand and use collections, generics, autoboxing, and enumerations
- Process large amount of data using Lambda expressions and the Stream API
- Abstract, static and private methods in interfaces
- Take advantage of the Java tooling that is available with the programming environment being used in the class
- Specific Java 11 features covered: Using the Local Variable Type in Lambda expressions; Updates made to the String API
- Time Permitting: Quick look ahead – Explore Java 12, Java 13, Java 14 and Beyond

Audience & Pre-Requisites

This is a fast-paced **introductory-level** Java programming course, designed for experienced developers who wish to get up and running with Java, or who need to reinforce sound Java coding practices, immediately. Attendees should have a working knowledge of developing OO software applications (coming from C++, C# or other OO programming backgrounds)..

Related & Java Essentials Alternative Courses: We offer othercourses with similar topics, geared for different experience levels or background:

- TT2000 Java for Non-Developers | Getting Started with Programming and Java Basics (5 days)
- TT2001 Java for Non-Developers (Optional Part 2) | OO and Java Basics (4 days)
- TT2120 Basic Java and OO Programming Essentials for Developers New to OO (C, COBOL, 4GL, etc.) (5 days)
- TT2104 Fast Track to Java Programming for OO Developers (C+, C#, etc.) (4 days)

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, Java EE, Spring, REST, Microservices, Unit Testing / TDD, Java secure coding, mobile development and more. Please see our **Java Developer Journey Courses & Learning Paths** for options based on your specific role and goals. We're happy to collaborate with you to recommend the best next steps in your learning journey.

Enhanced Learning Services: Please also ask about our **Pre-Training Class OnRamp & Prep / Primer** offerings, **Skills Gap Assessment Services**, **Case Studies**, **Knowledge Check Quizzes**, **Skills Immersion Programs & Camps**, **Collaborative Mentoring Services** and **Extended Learning Support** & **Post Training** services.

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 will work with you to tune this course and level of coverage to target the skills you need most.

Session: Java: A First Look

Lesson: The Java Platform

- The Java Platform
- Lifecycle of a Java Program
- Responsibilities of JVM
- Documentation and Code Reuse

Lesson: 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
- Lab: Exploring MemoryViewer

Lesson: The Eclipse Paradigm

- Become more familiar with Eclipse workbench concepts
- Explore the paradigm used by Eclipse, consisting of editors, views and perspectives in detail
- Introduce some commonly used views
- Explain Perspectives
- Tutorial: Setup Projects in Eclipse

Session: Getting Started with Java

Lesson: 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
- Lab: Create a Simple Class

Lesson: 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
- Lab: Create a Class with Methods

Session: Essential Java Programming

Lesson: Language Statements

- Arithmetic operators
- Operators to increment and decrement numbers
- Comparison operators
- Logical operators
- Return type of comparison and logical operators
- Use for loops
- Lab: Looping
- Lab: Language Statements

Lesson: Using Strings

- Create an instance of the String class
- Test if two strings are equal
- Get the length of a string Parse a string for its token components
- Perform a case-insensitive equality test
- Build up a string using StringBuffer
- Contrast String, StringBuffer, and StringBuilder
- Lab: Fun with Strings
- Lab: Using StringBuffers and StringBuilders

Lesson: 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
- Safely casting references to a more refined type
- Lab: Creating Subclasses

Lesson: 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
- Lab: Field Test

Lesson: Using Arrays
- Declaring an array reference
- Allocating an array
- Initializing the entries in an array
- Writing methods with a variable number of arguments
- Lab: Creating an Array

Lesson: Local-Variable Type Inference
- Explain type inference
- Infering types of local variables
- The var reserved type name
- Benefits of using var
- Backward compatibilty
- Lab: Using Local-Variable Type Inference (optional)

Lesson: 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
- Visibility in the Java Modular System
- Correctly executing a Java application class

Session: Object Oriented Development

Lesson: 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
- Lab: Salaries - Polymorphism

Lesson: 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
- Lab: Mailable - Interfaces

Session: Exception Handling

Lesson: Introduction to Exception Handling
- Introduce the Exception architecture
- Defining a try/catch blocks
- Checked vs Unchecked exceptions
- Lab: Exceptions

Lesson: Exceptions
- Defining your own application exceptions
- Automatic closure of resources
- Suppressed exceptions
- Handling multiple exceptions in one catch
- Lab: Exceptional

Session: Java Developer's Toolbox

Lesson: 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
- Lab: Using Primitive Wrappers
- Lab: Enumerations (optional)

Session: Advanced Java Programming

Lesson: Introduction to Generics
- Generics and Subtyping
- Bounded Wildcards
- Generic Methods
- Legacy Calls To Generics
- When Generics Should Be Used
- Lab: DynamicArray
- Lab: Adding Generics to Dynamic Array (optional)

Lesson: Lambda Expressions and Functional Interface
- Understanding the concept of functional programming
- Writing lambda expressions
- Understanding functionalinterfaces
- Explore the difference between anonymous classes and lambda expressions

Session: Working with Collections

Lesson: 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
- Lab: Create a simple Game using Collections

Lesson: Using Collections
- Collection Sorting
- Comparators
- Using the Right Collection
- Lambda expressions in Collections
- Lab: Using Collections

Session: Stream API

Lesson: 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
- Lab: Working with Streams

Lesson: 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
- Lab: Collecting

Session: The Java Module System

Lesson: Introduction to the Module System
- Introduce Project Jigsaw
- Classpath and Encapsulation
- The JDK internal APIs
- Java 9 Platform modules
- Defining application modules
- Define module dependencies
- Implicit dependencies
- Implied Readability
- Exporting packages
- Lab: Defining Modules

Additional Topics: Time Permitting

These topics will be included in your course materials but may or may not be presented during the live class depending on the pace of the course and attendee skill level and participation.

Lesson: Java Date/Time
- The Date and Calendar classes
- Introduce the new Date/Time API
- LocalDate, LocalDateTime, etc.
- Formatting Dates
- Working with time zones
- Manipulate date/time values
- Lab: Agenda

Lesson: Java 12 and beyond
- Overview of Java releases
- LTS vs non-LTS versions
- Overview of Records
- Introducing Multi-line String
- Exploring Switch expresions
- Explaining Helpfull Nullpointers
- Overview of Pattern Matching with instanceof
- Introducing Sealed types

Student Materials & Lab Environment

All course software (limited versions, for course use only), digital courseware files or course notes, labs / data sets and solutions (as applicable) are provided for you in our "easy access / no install required" high-speed remote lab environment. Our tech team works with every student to ensure everyone is set up with working access and ready to go prior to every course start date, ensuring a smooth delivery and great hands-on experience. Please ask for details.

For More Information

All courses can be presented **onsite** or **online,** or in a **combined / flex / blended learning format**, tailored to target your specific audience, needs and learning goals. We also offer focused, flexible **short courses, self-paced learning** options, **recorded sessions** and more. We train beginner to advanced skills in all areas we cover, and offer **New Hire / Cohort Training**, **Boot Camps**, **Skills Immersion Programs**, **Reskilling Programs**, **Skills Migration & Transition Programs**, and more. We collaborate with you to ensure all courses are truly targeted to meet your specific needs and learning skills, maximizing your valuable training time, as well as your important budget.

Please also visit our extensive **Public Training Schedule** for training for smaller groups or individuals. Please contact us for course details, **Corporate Rates** and **Special Discount Offers**.

For more information about our dedicated training services, collaborative coaching services, courseware licensing options, public course schedule, training management services, partner programs, or to see our complete list of course offerings 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.