Document Generated: 11/14/2026

Learning Style: Virtual Classroom

Technology:

Difficulty: Beginner

Course Duration: 5 Days

Core Java 25 Programming Developer's Workshop (TT2100)



About This Course:

This course explores key enhancements introduced since Java 11, including Records, Sealed Classes, Text Blocks, Pattern Matching, Virtual Threads, and numerous API updates. In addition, Java 25 continues the trend of simplifying the language for newcomers by providing streamlined syntax, enhanced pattern matching, and a more approachable set of defaults that make it easier for

developers new to Java to quickly write correct, readable, and efficient code without getting bogged down in boilerplate.

With Java 25, learning the language as a beginner has become more approachable than ever. New features such as Compact Source Files and Instance Main Methods allow students to create and run small programs with minimal setup, eliminating much of the boilerplate code that used to overwhelm newcomers. Flexible Constructor Bodies simplify class creation by reducing strict syntax requirements, making it easier to focus on concepts instead of technical hurdles. Additionally, Pattern Matching for Primitives streamlines common coding tasks by replacing verbose casting with more natural and readable code. These improvements reflect Javaâ€™s ongoing effort to lower the learning curve, helping new developers get started faster while still building a foundation that is compatible with professional software development practices.

Developers leaving this course will be well-prepared to work on Java 8, Java 11, Java 17, and Java 21 projects, while also being ready to contribute effectively to modern projects using Java 25.

Objectives:

- Understand what object-oriented (OO) programming is and recognize the advantages it provides in today's software development world.
- Gain a solid grasp of the fundamentals of the Java language, including its importance, uses, strengths, and limitations.
- Connect the basics of the Java language to OO programming and the Java Object Model.
- Learn to use Java's exception handling features to build more reliable applications.
- Work with the Java Modular System (Project Jigsaw) to create organized, maintainable applications.
- Design and implement classes that demonstrate inheritance and polymorphism.
- Use collections, generics, autoboxing, and enumerations to efficiently manage data.
- Process large volumes of data using lambda expressions and the Stream API.
- Define and implement abstract, static, and private methods in interfaces.
- Take advantage of Java development tooling available in modern programming environments.

- Write modern Java code using switch expressions for more concise and expressive branching logic.
- Use text blocks to create clean, multi-line string literals.
- Apply pattern matching for instanceof to write safer and more readable type checks.
- Introduce records as immutable data carriers to simplify domain models.
- Use pattern matching in switch statements to simplify conditional logic.
- Apply record patterns to deconstruct and access data directly within records

Specific Java 25 features that are covered in the course include:

- Learn how to use primitive types in patterns, allowing pattern matching to work directly with primitive values for safer and more concise code.
- Understand module import declarations, which simplify modular programming by making module dependencies clearer and easier to manage.
- Explore flexible constructor bodies, which give developers greater control over how constructors are structured and how initialization logic is handled.
- Practice writing compact source files and instance main methods, which reduce boilerplate and make it easier for beginners to create simple Java applications.

Audience:

- Participants should be familiar with basic programming concepts such as variables, control structures, functions/methods, and data structures.

Prerequsites:

- This is a foundational Java programming course designed for attendees who already have prior development experience in another programming language. Participants should be familiar with basic programming concepts such as variables, control structures, functions/methods, and data structures

Course Outline:

- The Java Platform
- Using the JDK
- Using the IntelliJ IDE
- Writing a Simple Class
- Adding Methods to the Class
- Object-Oriented Programming
- Language Statements
- Using Strings and Text Blocks
- Fields and Variables
- Specializing in a Subclass
- Using Arrays
- Records
- Java Packages and Visibility
- Utility Classes
- Inheritance and Polymorphism
- Interfaces and Abstract Classes
- Sealed Classes
- Pattern Matching
- Introduction to Exception Handling
- Exceptions
- Building Java Applications
- Introduction to Generics
- Introducing Lambda Expressions and Functional Interfaces
- Collections
- Using Collections

- Streams
- Collectors