Getting Started with Programming, OO & Java Basics for Non-Developers - TT2000

Jumpstart your programming journey with hands-on Java training, clear explanations, and guided practice in writing and structuring code

Duration: 5 Days
Skill Level: Introductory
Available Format: Instructor-Led Online; Instructor-Led, Onsite In Person ; Blended; On Public Schedule

Programming is one of the most valuable skills you can develop, and Java is a great place to start. This expert-led course makes learning Java 21 accessible, engaging, and hands-on—even if you have never coded before. Through a structured approach, you will gain a clear understanding of how programs work, write real Java code, and develop problem-solving skills essential for software development. You will work with key programming concepts like loops, methods, and object-oriented design while using industry-standard tools like IntelliJ or Eclipse, and the Java Development Kit (JDK). With a strong focus on practical learning, you will troubleshoot errors, build structured applications, and gain confidence in writing and reading Java code. Whether you are an aspiring developer, a technical professional, or someone who wants to collaborate more effectively with software teams, this course gives you the skills and experience needed to take your first steps into programming.

What You'll Learn

Overview

Learning to program opens up a world of possibilities, whether you are looking to build applications, improve your problem-solving skills, or just understand how software works. **Getting Started with Programming, OO, and Java 21 Basics for Non-Developers** is a hands-on, expert-led course designed to make coding approachable, even if you have never written a line of code before. You will learn how programs work, how to think like a developer, and how to write and organize Java code in a way that

makes sense. With plenty of hands-on practice, you will gain confidence using Java 21's latest features, working with IDEs, and understanding key concepts like variables, loops, methods, and object-oriented programming.

This course is built for technically-minded beginners who want real coding experience in a structured, supportive environment. Over five days, you will write your own Java programs, explore core programming concepts, and see firsthand how to build applications from the ground up. You will also work with essential Java tools, handle errors, and learn best practices for writing clean, efficient code. With expert guidance and a focus on hands-on learning, you will walk away with practical skills that you can use immediately. Whether you are starting your journey into development, working with technical teams, or just curious about how programming works, this course will give you the foundation you need to get started.

Becoming a modern software developer is like learning a new language; it requires study, practice, and dedication well beyond this course to apply your new skills effectively. While this five day program won't transform you into an experienced developer, it will lay a solid foundation in coding basics using Java, while teaching you to think like a programmer. Although this course is technical in nature, our instructors will guide you every step of the way, providing a supportive environment for you to explore, ask questions, and prepare for your next learning milestones.

Objectives

Learning how to code and become a modern software developer is like trying to learn and become fluent in a new spoken language. Learning any new language takes study, practice, more study, and more practice, to truly be able put your newly learned skills to work in a practical way. This course won't make you an experienced developer in the five days we have with you, but we'll ensure that you're provided with a solid introduction to coding basics, along with real hands-on experience programming in Java. All the while focused on learning how to Think Like a Programmer. Please note that this course is for beginners new to programming, but it is technical in nature. Our instructors are there to guide you through the process and provide you with a trusted platform to dig into something new, ask questions, and leave the class ready to take the next steps in your learning journey.

In this course, you will gain the essential programming skills needed to write, structure, and troubleshoot Java applications with confidence. By the end, you will be able to:

- Write and run Java programs using IntelliJ (or Eclipse, if requested) and the Java Development Kit (JDK) to understand how code is compiled and executed.

- Use variables, loops, conditionals, and methods to control program flow and manage data efficiently.
- Apply object-oriented programming principles, including classes, objects, inheritance, and polymorphism, to design well-structured applications.
- Work with Java's core features, such as arrays, strings, exceptions, and collections, to build functional and organized code.
- Understand and apply best practices in writing clean, reusable, and maintainable Java code.
- Build confidence in troubleshooting errors, debugging programs, and thinking like a developer.

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 for beginners who want to learn programming from the ground up, whether you are an aspiring developer, a technical professional looking to collaborate with engineers, or someone supporting software projects. No prior programming experience is required, but you should be comfortable using a computer and working with basic software tools.

Please note that although this course is for beginner-level students, it is technical in nature. **If you're moving from a truly non-technical role into coding for the first time, please reach out to us for some additional guidance** or light course prep suggestions which can really be helpful before you jump into this course head-on. We want your experience to be exciting, challenging and useful, without being overwhelming. We're here to help!

Attendees might include:

- Technically-minded attendees who want or who want to begin the process of becoming an OO application developer
- Technical team members from non-development roles, re-skilling to move into software and application development roles within an organization
- Recent college graduates looking to apply their college experience to programming skills in a professional environment, or perhaps needing to learn the best practices and standards for programming within their new organization

- Technical managers tasked with overseeing programming teams, or development projects, where basic coding knowledge and exposure will be useful in project oversight or communications needs

Pre-Requisites

To ensure a smooth learning experience and to gain the most from attending this course, you should have the following incoming skills:

- Basic computer literacy: Familiarity with computer operating systems, file management, and general navigation to ensure a smooth learning experience.
- Foundational knowledge of IT concepts: Understanding of essential IT terminologies and concepts, such as computer networks, software applications, and data storage.
- Analytical thinking: Ability to analyze problems and think critically to develop logical solutions, fostering a programmer's mindset.

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 may adjust during live delivery based on audience skill-level, needs and participation.

1. Overview of Computer Programming

- Explain what a program is
- Explain why there are different types of languages
- Explain what a compiler is
- Explain what an interpreter is
- Lab: Matching Terms

2. Features of a Program

- Understand what the entry and exit points of an application are
- Explain what variables are
- Explain what programming instructions are
- Explain what errors and exceptions are
- Understand what programming algorithms are

3. Software Development Life Cycle

- Explain the purpose of the software development life cycle
- Explain what each phase is for

- Explain the difference between the software development life cycle and a methodology

4. Thinking in Objects

- Understand the difference between a class and an object
- Deconstruct an object into attributes and operations
- Map an object to a class
- Define inheritance
- Lab: Designing an Application

5. 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

6. 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

7. The IntelliJ Paradigm

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

8. 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

- Lab: Create a Simple Class

9. 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

10. Exploring Object-Oriented Programming

- Real-World Objects
- Classes and Objects
- Object Behavior
- Methods and Messages
- Lab: Define and use a New Java class

11. Inheritance, Abstraction, and Polymorphism

- Encapsulation
- Inheritance
- Method Overriding
- Polymorphism
- Lab: Define and use Another Java Class

12. Language Statements

- Arithmetic operators
- Operators to increment and decrement numbers
- Comparison operators
- Logical operators
- Return type of comparison and logical operators
- Use for loops
- Swtch Expressions
- Switch Expressions and yield
- Lab: Looping (optional)
- Lab: Language Statements
- Lab: Switch Expressions

13. 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

14. 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

15. 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
- Pattern matching for instanceof
- Overriding subclass methods
- Safely casting references to a more refined type
- Lab: Creating Subclasses

16. 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

17. Formatting Strings

- Format a String using the formatter syntax
- Apply text formatting
- Use String.format and System.out.printf
- Lab: Textblocks

18. Records

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

19. 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
- The Java Modular System
- Defining Modules

20. 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
- Deprecating classes and methods
- Lab: Enumerations

21. 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

22. 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

23. 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: Interfaces

24. Introduction to Exception Handling

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

25. 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)
- Lab: Exceptional
- Lab: Helpful Nullpointers

26. 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

27. 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

28. 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

Follow On Courses

TEST-01	Mastering Machine Learning Operations (MLOps) and AI Security Boot Camp
TT3503	Test Driven Development (TDD) and Unit Testing Essentials
TT3320	Core Spring Quick Start | Introduction Spring 6.x and Spring Boot
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

TT2100	Core Java Programming Developer's Workshop
TT2120	Basic Java Programming for Developers New to OO (C, COBOL, etc.)
TT2104	Fast Track to Core Java Programming for OO Experienced 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.