TriveraTech ®
T E C H N O L O G Y   T R A I N I N G

Java Developer SkillJourney Series

Basic Java Programming for Developers New to OO (TT2120)

Hands-on Java Workshop: Gain the skills required to create efficient, scalable high-performance applications using Java.

Course Snapshot

- **Course: Basic Java Programming for Developers New to OO (TT2120)**
- **Duration**: 5 days
- **Audience & Skill Level:** The basic level course is geared for experienced developers who are new to OO programming and Java. This course is not for new developers.
- **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.
- **Format / Hands-on:** This course combines engaging instructor-led presentations and practical demonstrations with hands-on programming exercises, challenge labs, use case exploration and engaging group activities. Student machines are required.
- **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.

Overview

Geared for experienced developers, **Basic Java Programming for Developers New to OO**, this hands-on, workshop-style course will provide you with an immersive learning experience that will expand your skillset and open doors to new opportunities within the ever-growing technology landscape. Mastering Java and its powerful capabilities will provide you with the competitive edge you need to stand out in today's fast-paced development world.

Working in a hands-on learning environment led by our expert coach, you'll thoroughly explore the foundations of the Java platform, essential programming concepts, and advanced topics, ensuring you acquire a strong understanding of the language and its ecosystem. The object-oriented programming principles taught in this course promote code reusability and maintainability, enabling you to streamline development processes and reduce long-term costs.

As you progress through the course, you will also gain familiarity with using an IDE, enhancing your development workflow and collaboration with other Java developers, enabling you to integrate seamlessly into new projects and teams. You'll also gain practical experience in applying the concepts and techniques learned, solidifying your newly acquired skills and facilitating their direct application in real-world scenarios. You'll exit this course empowered to create robust, scalable, and efficient Java-based applications that drive innovation and growth for your organization.

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.

NOTE: Developers new or newer to programming should consider the **TT2000 Getting Started with Programming, OO and Java Basics** as an alternative.

Learning Objectives

Working in an interactive learning environment, led by our expert facilitator, you'll learn to:
- 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.
- 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

Need different skills or topics? If your team requires different topics or tools, additional skills or custom approach, this course may be further adjusted to accommodate. Our team will collaborate with you to understand your needs and will target the course to focus on your specific learning objectives and goals.

Audience & Pre-Requisites

In order to be successful in this course you should have incoming hands-on experience with another programming language. This course is not for non-developers or new developers. Possible roles that may attend this course include:
- Software Developers: Professionals who have been working with other programming languages and want to expand their skillset by learning Java and its object-oriented features.
- Web Developers: Those who work on web applications and want to enhance their back-end development capabilities with Java.
- Mobile App Developers: Developers who wish to enter the world of Android app development, where Java is a widely-used language for creating mobile applications.
- Full-Stack Developers: Professionals who have experience with front-end technologies and want to deepen their knowledge of back-end development using Java.
- Game Developers: Developers who are interested in creating games for various platforms, including desktop, mobile, and web, using Java as their primary programming language.

Related Courses / Trivera's Java Developer 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 skills, Java for Web / Full Stack, Jakarta / Java 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 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'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: Working with the IDE (This course is also offered using Eclipse – please inquire

for details and options)

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. **Object-Oriented Programming**
- Real-World Objects
- Classes and Objects

- Object Behavior
- Methods and Messages

7. **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

8. **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

9. **Fields and Variables**
- Discuss Block Scoping Rules
- Distinguish between instance

Trivera Technologies ● IT Training, Coaching & Skills Development Solutions   www.triveratech.com ● Info@triveratech.com ● toll free 844.475.4559
Basic Java for Developers New to OO ● TT2120   2024 ● Page 3

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

10. 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

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

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

13. 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

14. 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

15. 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

16. 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

17. Sealed Classes
- Introduce sealed classes
- The sealed and permits modifier
- Sealed interfaces
- Sealed classes and pattern

matching

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

19. Introduction to Exception Handling
- Introduce the Exception architecture
- Defining a try/catch blocks
- Checked vs Unchecked exceptions

20. 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)

21. 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

22. Introduction to Generics
- Generics and Subtyping
- Bounded Wildcards
- Generic Methods

23. Introducing Lambda Expressions and Functional Interfaces
- Understanding the concept of functional programming
- Understanding functional interfaces
- Lambda's and type inference

24. Collections

Trivera Technologies ● IT Training, Coaching & Skills Development Solutions   www.triveratech.com ● Info@triveratech.com ● toll free 844.475.4559

Basic Java for Developers New to OO ● TT2120   2024 ● Page 4

- 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

25. 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

---

Setup Made Simple! Learning Experience Platform (LXP)

All applicable course software, digital courseware files or course notes, labs, data sets and solutions, live coaching support channels, CodeCoach.AI anytime tutor access, 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. Access periods vary by course. 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.

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