Introduction to Programming and Java Basics for Non-Developers

Kickstart Your Software Development Skills with this Gentle Introduction to Coding, Object Oriented Development and Java

Course Snapshot

- **Course: Introduction to Programming & Coding | Java Basics for Non-Developers (TT2000 / TTCODE101-J)**
- **Duration**: 5 days
- **Audience & Skill-Level**: This is a very **basic-level** course geared for students **new to programming**.
- **Hands-on Learning:** This course is approximately 50% hands-on lab to lecture ratio, combining engaging expert lessons, demos and group discussions with real-world, skills-focused machine-based labs and exercises. Student machines are required.
- **Delivery Options**: This course is available for **onsite private classroom presentation**, **live online virtual presentation**, or can be presented in a **blended learning format**. Please also ask about our **Self-Paced** / **Video** / **QuickSkills** or **Mini-Camp** / **Short Course** flexible delivery options.
- **Public Schedule**: This course has active dates on our live-online 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.

Programmers are in demand!

Companies are constantly challenged to keep their applications, development projects, products, services (and programmers!) up to speed with the latest industry tools, skills, technologies and practices to stay ahead in the ever-shifting markets that make up today's fiercely competitive business landscape. The need for application, web and mobile developers and coders is seemingly endless as technologies regularly change and grow to meet the modern needs of demanding industries and clients.

There are hundreds of roles that reply on coding knowledge and programming skills. Having programming skills can enable you to:

- Develop applications and contribute to application development projects and teams
- Integrate, change. expand or maintain outsourced development work being brought back in house
- Use or maintain software, tools, systems or networks that require coding skills to operate, handle data or complete other tasks
- Collaborate and communication more effectively with developers
- More effectively manage development teams or projects that involve programming

To meet the need to bring this knowledge and skill in house, companies might

- Locate, engage and hire experienced developers
- Outsource development work to third party firms
- Onboard new hires or recent college graduates / cohorts with limited practical experience, then train them specifically to fill these roles
- Invest in and 're-skilling' current employees into more modern and productive roles within the firm, retaining and leveraging existing company knowledge and talent.

Learning to code, even to a basic level, is a truly worthwhile investment. Having coding skills, or even basic knowledge, can elevate your ability gain a new job in programming, help you gain longer term job security in your current organization, or can even expand your current role into other areas in your team or organization.

Overview

Introduction to Programming and Java Basics for Non-Developers is a skills-focused, hands-on coding course that teaches students the fundamentals of programming object oriented (OO) applications with Java to a basic level, using sound coding skills and best practices for OO development. This course is presented in a way that enables you to embrace the fundamentals of coding as well as an introduction to Java, in a gentle paced environment that focuses on coding basics.

Throughout the course, you'll explore the application development cycle, structure of programs, and specific language syntax. The course introduces important algorithmic constructs, string and character manipulation, dynamic memory allocation, standard I/O, and fundamental object-oriented programming concepts. The course explains the use of inheritance and polymorphism early on so

you can practice these skills extensively in the hands-on labs. Structured programming techniques and error handling are emphasized. The course includes the processing of command line arguments and environment variables, so you'll be able write flexible, user-friendly programs. **You'll leave this course armed with the required skills to begin your learning journey as a Java programmer using modern coding skills and technologies.**

NOTE: Although this course is geared for non-developers, it is helpful for attendees to have some background in IT and to be comfortable working with computers, with the ultimate goal to become a Java software developer.

Learning 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 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.

Working in a lab-intesive hands-on learning environment, guided by our expert team, you'll explore:

- The steps involved in the creation and deployment of a computer program
- What OO programming is and what the advantages of OO are in today's world
- To work with objects, classes, and OO implementations
- The basic concepts of OO such as encapsulation, inheritance, polymorphism, and abstraction
- The basic constructs that all programming languages share
- The basic Java constructs supporting processing as well as the OO orientation
- How to use Java exception handling
- About and how to use classes, inheritance and polymorphism
- About use collections, generics, autoboxing, and enumerations
- How to take advantage of the Java tooling that is available with the programming environment being used in the class

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. We offer additional Java, JEE and Java for Web application development, design, testing, services, application security and other related topics that may be blended with this course for a track that best suits your needs. 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

This course is intended for anyone who is new programming, and wants to start learning how to code using Java. **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

Before attending this course, you definitely should be able to:

- Use computers to start programs, open and save files, navigate application menus and interfaces
- Understand logical concepts such as comparisons
- Understand number theory
- Create, understand, and follow structured directions or step-by-step procedures
- Understand and apply abstract concepts to concrete examples

Follow On Courses: Our Java tracks include a wide variety of follow-on courses and learning paths for leveraging Java for next-level development, testing, security and more. Please see our **Java Developer Training Suite & Learning Paths** list of courses, or inquire for recommendations 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 will work with you to tune this course and level of coverage to target the skills you need most. Course agenda, topics and labs are subject to adjust during live delivery in response to student skill level, interests and participation.

Session: Introduction to Computer Programming

Lesson: Programming Tools

- Explain why we use various tools when programing
- Define the four major types of tools
- Lab: Setup the 'Java Development Environment'
- Lab: The HelloWorld 'Application' (optional)

Session: Programming Fundamentals

Lesson: Programming Basics

- Explain why different languages have different syntax
- Understand some basic features of any program
- Explain why we would break a program apart
- Lab: Using an Integrated Development Environment

Lesson: Thinking About Objects

- Define what an object is
- Explain the difference between an object and a class
- Design a class using a class diagram
- Understand how to define a basic class
- Explain how to compile and run a Java application

Lesson: Programming Constructs

- Explain what variables and constants are and when to use the
- Explain what arrays are and why they are used
- Explain what a method is
- Write if statements
- Write loops
- Lab: Modeling the Television Object

Session: Java: A First Look

Lesson: The Java Platform

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

Lesson: Using the JDK

- Setting Up Environment
- Locating Class Files
- Compiling Package Classes
- Source and Class Files
- Java Applications
- Lab: Exploring MemoryViewer
- Lab: Exploring ColorPicker (optional)

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

Lesson: Writing a Simple Class

- Classes in Java
- Class Modifiers and Types
- Class Instance Variables
- Primitives vs. Object References
- Creating Objects
- Lab: Create a Simple Class

Lesson: Adding Methods to the Class

- Passing Parameters into Methods
- Returning a Value from a Method
- Overloaded Methods
- Constructors
- Optimizing Constructor Usage
- Lab: Create a Class with Methods

Session: OO Concepts

Lesson: Object-Oriented

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

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

Session: Getting Started with Java

Lesson: Language Statements
- Operators
- Comparison and Logical Operators
- Looping
- Continue and Break Statements
- The switch Statement
- The for-each() Loop
- 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
- Contrast String, StringBuffer, and StringBuilder
- Lab: Fun with Strings
- Lab: Using StringBuffers and StringBuilders

Lesson: Specializing in a Subclass
- Extending a Class
- implementing equals and toString
- Using instanceof to verify the class type of an object reference
- The Object Class
- Default Constructor
- Implicit Constructor Chaining
- Safely casting references to a more refined type
- Lab: Creating Subclasses
- Lab: Defining the Student Subclass

Session: Essential Java Programming

Lesson: Java Coding and Debugging Tools
- Java Editing Assistance
- The Debug Perspective
- Breakpoints and the Debug View
- Debug Mode
- Refactoring Overview

Lesson: Fields and Variables
- Instance vs. Local Variables: Usage Differences
- Data Types
- Default Values
- Block Scoping Rules
- Final and Static Fields
- Static Methods
- Lab: Field Test

Lesson: Using Arrays
- Declaring an array
- Accessing the Array
- Multidimensional Arrays
- Copying Arrays
- Variable Arguments
- Lab: Creating an Array
- Lab: Defining the Student Array

Lesson: Java Packages and Visibility
- Class Location of Packages
- The Package Keyword
- Importing Classes
- Executing Programs
- Java Naming Conventions

Session: Advanced Java Programming

Lesson: Inheritance and Polymorphism
- Writing a subclass with a method that overrides a method in the superclass
- Grouping objects by their common supertype
- Utilize polymorphism by invoking subclass method implementations through superclass references
- Casting a supertype reference to a subtype reference
- Using the final keyword on methods and classes
- Lab: Salaries - Polymorphism

Lesson: Interfaces and Abstract Classes
- Define supertype contracts using interfaces
- Define supertype contracts using abstract classes
- Implement concrete classes using interfaces
- Implement concrete classes using abstract classes
- Explain the advantage of interfaces over abstract classes
- Explain the advantage of abstract classes over interfaces
- Lab: Mailable - Interfaces

Lesson: Exceptions
- Exception Architecture
- Handling Multiple Exceptions
- Automatic Closure of Resources
- Creating Your Own Exceptions
- Throwing Exceptions
- Checked vs. Unchecked Exceptions
- Lab: Exceptions

Session: Java Developer's Toolbox

Lesson: Utility Classes
- Wrapper Classes
- The Number Class
- Random Numbers
- Autoboxing/Unboxing
- The Date Class
- Lab: Using Primitive Wrappers

Lesson: Enumerations and Static Imports
- Enumeration Syntax
- When You Should Use Enumerations
- Using Static Imports
- When You Should Use Static Imports
- Lab: Enumerations (optional)

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

- Lab: Agenda

Session: Collections and Generics

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

Lesson: Collections

- Characterizing Collections
- Collection Interface Hierarchy
- Iterators
- The Set, List and Queue Interfaces
- The Map Interface
- Collections and Multithreading
- Lab: Using Hashtable and HashMap
- Lab: Collections Poker
- Lab: Writing a Collection (optional)

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.

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 • COURSEWARE LICENSING & DEVELOPMENT MENTORING • ASSESSMENTS • 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

Explore Trivera's Ways to Learn!