Basic Java 17 Programming for Developers New to OO (C, Mainframe, COBOL)

Days: *5*
Prerequisites: 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.
Audience: 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.

Description: Geared for experienced developers, 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.

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

Basic Java 17 Programming for Developers New to OO (C, Mainframe, COBOL)

- 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.
- Work with specific Java 17 features that are covered in the course include: Switch Expressions, Text blocks, Pattern matching for instanceof, and introducing records as carrier of immutable data.

OUTLINE:

LESSON 1: THE JAVA PLATFORM

- Introduce the Java Platform
- Discuss the lifecycle of a Java Program
- Explain the responsibilities of the JVM
- Documentation and Code Reuse

LESSON 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
- LAB: Ecploring MemoryViewer

LESSON 3: 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

Getting Started with Java

LESSON 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
- Lab: Create a Simple Class

LESSON 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 keyword to distinguish local variables from instance variables

Lab: Create a Class with Methods

OO Concepts

LESSON 6: OBJECT-ORIENTED PROGRAMMING

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

LESSON 7: INHERITANCE, ABSTRACTION, AND POLYMORPHISM

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

Essential Java Programming

LESSON 8: LANGUAGE STATEMENTS

Basic Java 17 Programming for Developers New to OO (C, Mainframe, COBOL)

- 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
- Lab: Looping
- Lab: Language Statements

LESSON 9: 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
- Lab: Fun with Strings
- Lab: Using StringBuffers and StringBuilders

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

LESSON 11: 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
- Local variable type influence
- Lab: Field Test

LESSON 12: 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 13: RECORDS

- Data objects in Java
- Introduce records as carrier of immutable data
- Defining records
- Lab: Record

LESSON 14: 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
- Lab: Defining Modules

Object Oriented Development

LESSON 15: INHERITANCE AND POLYMORPHISM

- Write a subclass with a method that overrides a method in the superclass

Basic Java 17 Programming for Developers New to OO (C, Mainframe, COBOL)

- 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

LESSON 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

LESSON 17: SEALED CLASSES

- Introduce Sealed classes
- The sealed and permits modifiers
- Sealed Interfaces

Exception Handling

LESSON 18: INTRODUCTION TO EXCEPTION HANDLING

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

LESSON 19: EXCEPTIONS

- Defining your own application exceptions
- Automatic closure of resources
- Suppressed exceptions
- Handling multiple exceptions in one catch
- Helpful Nullpointers
- Enhanced try-with-resources
- Lab: Exceptional

Java Developer's Toolbox

LESSON 20: DEVELOPING APPLICATIONS

- Introduce the wrapper classes
- Explain Autoboxing and Unboxing
- Converting String representations of primitive numbers into their primitive types
- Defining Enumerations
- Using static imports
- Deprecating methods
- Lab: Using Primitive Wrappers
- Lab: Enumerations (optional)

Advanced Java Programming

LESSON 21: INTRODUCTION TO GENERICS

- Generics and Subtyping
- Bounded Wildcards
- Generic Methods
- Legacy Calls To Generics
- When Generics Should Be Used
- Lab: DynamicArray

LESSON 22: LAMBDA EXPRESSIONS AND FUNCTIONAL INTERFACES

- Understanding the concept of functional programming
- Understanding functional interfaces
- Writing lambda expressions
- Lab: Using Labmda

Working with Collections

LESSON 23: 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

Basic Java 17 Programming for Developers New to OO (C, Mainframe, COBOL)

LESSON 24: USING COLLECTIONS

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

Bonus 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 25: STREAMS

- Understanding the problem with collections in Java
- 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
- takeWhile and dropWhile intermediate operations
- Lab: Working with Streams

LESSON 26: 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