

# Specialized - Introduction to Java 9 / 10 for Developers New to OO Programming

---

<b>Code:</b>	TT2120-J9X
<b>Length:</b>	5 days
<b>URL:</b>	<a href="#">View Online</a>

---

Java 9 (or 10) and OO Programming Essentials for Developers New to OO is a five-day, hands-on Java training course geared for developers who have little or no prior working knowledge of object-oriented programming languages (such as those working on (C, COBOL, 4GL, etc.) Throughout the course, students learn the best practices for writing great object-oriented programs in Java, using sound development techniques, new improved features for better performance, and new capabilities for addressing rapid application development. Special emphasis is placed on object oriented concepts and best practices.

- This course introduces new features in Java 9 and Java 10, including the Java Modular System and Local Variable Type Inference. Developers leaving this course will be able to participate in projects that are still on Java 8, while they are also ready to move onto projects using Java 10.

## Skills Gained

Our engaging instructors and mentors are highly-experienced practitioners who bring years of current "on-the-job" application development experience into every classroom. Working within in an engaging, hands-on learning environment, guided by our expert team, attendees will learn to:

- Understand what OO programming is and what the advantages of OO are in today's world
- Work with objects, classes, and OO implementations
- Understand the basic concepts of OO such as encapsulation, inheritance, polymorphism, and abstraction
- Understand not only the fundamentals of the Java language, but also its importance, uses, strengths and weaknesses
- Understand the basics of the Java language and how it relates to OO programming and the Object Model
- Work with the Modular system (Project Jigsaw)
- Understand and use classes, inheritance and polymorphism
- Understand and use collections, generics, autoboxing, and enumerations
- Process large amount of data using Lambda expressions and the Stream API
- Abstract, static and private methods in interfaces
- Use the JDBC API for database access
- Take advantage of the Java tooling that is available with the programming environment being used in the class

## Who Can Benefit

This is an introductory -level course designed for attendees with prior development experience in another language, such as COBOL, 4GL, Mainframe or other non-object oriented languages. This course is not geared for non-developers.

## Course Details

Specific Java 9 features that are covered in the course include:

- The Modular system (Project Jigsaw)
- JShell
- Private methods in interfaces

Optional: Specific Java 10 features that are covered in the course include:

- Local Variable Type Inference

## 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
- Exercise: Exploring MemoryViewer
- Exercise: Exploring ColorPicker

Lesson: The Eclipse Paradigm

- Workbench and Workspace
- Views
- Editors
- Perspectives
- Projects
- Tutorial: Setup Projects in Eclipse

## Session: Getting Started with Java

Lesson: Writing a Simple Class

- Classes in Java
- Class Modifiers and Types
- Class Instance Variables
- Primitives vs. Object References
- Creating Objects

- Exercise: 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
- Exercise: Create a Class with Methods

## **Session: OO Concepts**

Lesson: Object-Oriented Programming

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

Lesson: Inheritance, Abstraction, and Polymorphism

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

## **Session: Essential Java Programming**

Lesson: Language Statements

- Operators
- Comparison and Logical Operators
- Looping
- Continue and Break Statements
- The switch Statement
- The for-each() Loop
- Exercise: Looping
- Exercise: Language Statements

Lesson: Using Strings

- Strings
- String Methods
- String Equality

- StringBuffer
- StringBuilder
- Exercise: Fun with Strings
- Exercise: Using StringBuffers and StringBuilders

#### Lesson: Specializing in a Subclass

- Extending a Class
- Casting
- The Object Class
- Default Constructor
- Implicit Constructor Chaining
- Exercise: Creating Subclasses

#### Lesson: Fields and Variables

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

#### Lesson: Using Arrays

- Arrays
- Accessing the Array
- Multidimensional Arrays
- Copying Arrays
- Variable Arguments
- Exercise: Creating an Array

#### Lesson: Local-Variable Type Inference

- Type inference
- Inferring Types of Local Variables
- The var Reserved Type name
- Benefits of Using var
- Backward Compatibility

#### Lesson: Java Packages and Visibility

- Class Location of Packages
- The Package Keyword

- Importing Classes
- Executing Programs
- Visibility in the Modular System
- Java Naming Conventions

## **Session: Object Oriented Development**

Lesson: Inheritance and Polymorphism

- Polymorphism: The Subclasses
- Upcasting vs. Downcasting
- Calling Superclass Methods From Subclass
- The final Keyword
- Exercise: Salaries - Polymorphism

Lesson: Interfaces and Abstract Classes

- Separating Capability from Implementation
- Abstract Classes
- Implementing an Interface
- Abstract Classes vs. Interfaces
- Exercise: Mailable - Interfaces

## **Session: Exception Handling**

Lesson: Introduction to exception handling

- Exception Architecture
- Throwing Exceptions
- Checked vs. Unchecked Exceptions
- Exercise: Exceptions

Lesson: Exceptions

- Handling Multiple Exceptions
- Automatic Closure of Resources
- Creating Your Own Exceptions
- Exercise: Exceptional

## **Session: Java Developer's Toolbox**

Lesson: Utility Classes

- Wrapper Classes
- Autoboxing/Unboxing
- Enumeration Syntax
- Using Static imports
- Exercise: Using Primitive Wrappers

- Exercise: Enumerations

Lesson: 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
- Exercise: Agenda

Lesson: Formatting Strings (Optional)

- String.format
- System.out.printf
- The Formatter class
- Using the formatting syntax

## **Session: JShell**

Lesson: JShell

- Introduction to JShell
- Running Expressions in JShell
- Importing packages
- Defining methods and types
- Using the JShell editor
- Save and loading state
- Exercise: Working With JShell

## **Session: Advanced Java Programming**

Lesson: Introduction to Generics

- Generics and Subtyping
- Bounded Wildcards
- Generic Methods
- Legacy Calls To Generics
- When Generics Should Be Used
- Exercise: DynamicArray
- Exercise: Adding Generics to Dynamic Array

Lesson: Lambda Expressions and Functional Interface

- Lambda Expression Syntax
- Functional Interfaces

- Type Inference in Java 8
- Method references
- Exercise: Using Lambda

## **Session: Working with Collections**

Lesson: Collections

- Characterizing Collections
- Collection Interface Hierarchy
- The Set, List and Queue Interfaces
- Map Interfaces
- Exercise: Create a simple Game using Collections

Lesson: Using Collections

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

## **Session: Stream API**

Lesson: Streams

- Processing Collections of data
- The Stream interface
- Reduction and Parallelism
- Filtering collection data
- Sorting Collection data
- Map collection data
- Find elements in Stream
- Numeric Streams
- Create infinite Streams
- Sources for using Streams
- Exercise: Working with Streams

Lesson: Collectors

- Creating Collections from a Stream
- Group elements in the Stream
- Multi-level grouping of elements
- Partitioning Streams
- Exercise: Collecting

## Session: The Java Module System

Lesson: Introduction to the Module System

- Introduce Project Jigsaw
- Classpath and Encapsulation
- The JDK internal APIs
- Java 9 Platform modules
- Defining application modules
- Define module dependencies
- Implicit dependencies
- Implied Readability
- Exporting packages
- Exercise: Defining Modules

## Session: Accessing Resources

Lesson: Java Data Access JDBC API

- Connecting to the Database
- Statement and PreparedStatement
- ResultSet
- Executing Inserts, Updates, and Deletes
- Controlling Transactions and Concurrency
- Exercise: Intro to JDBC

Lesson: Introduction to Annotations (Optional)

- Annotations Overview
- Working with Java Annotations
- Exercise: Using Annotations

---

## Schedule (as of 4 )

Date	Location	
Nov 9, 2020 – Nov 13, 2020	Virtual	<a href="#">Enroll</a>
Dec 14, 2020 – Dec 18, 2020	Virtual	<a href="#">Enroll</a>

---