

## **Java 8 Programming for OO Experienced Developers (C#, C++, etc.) (TT2100-J8)**

**Modality: Virtual Classroom**

**Duration: 5 Days**

Java 8 Programming for OO Experienced Developers

### **Course Outline:**

#### **Module 1: 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

#### **Module 2: 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

##### **Lesson: Adding Methods to the Class**

- Passing Parameters Into Methods
- Returning a Value From a Method
- Overloaded Methods
- Constructors
- Optimizing Constructor Usage

##### **Lesson: Language Statements**

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

### **Lesson: Using Strings**

- Strings
- String Methods
- String Equality
- StringBuffer
- StringBuilder

### **Lesson: Specializing in a Subclass**

- Extending a Class
- Casting
- The Object Class
- Default Constructor
- Implicit Constructor Chaining

## **Module 3: Essential Java Programming**

### **Lesson: Fields and Variables**

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

### **Lesson: Using Arrays**

- Arrays
- Accessing the Array
- Multidimensional Arrays
- Copying Arrays
- Variable Arguments

### **Lesson: Java Packages and Visibility**

- Class Location of Packages
- The Package Keyword
- Importing Classes
- Executing Programs
- Java Naming Conventions

## **Module 4: Advanced Java Programming**

### **Lesson: Inheritance and Polymorphism**

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

### **Lesson: Interfaces and Abstract Classes**

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

### **Lesson: Exceptions**

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

## **Module 5: Java Developer's Toolbox**

### **Lesson: Utility Classes**

- Wrapper Classes
- The Number Class
- Random Numbers
- Autoboxing/Unboxing
- The Date Class

### **Lesson: Enumerations and Static Imports**

- Enumeration Syntax
- When You Should Use Enumerations
- Using Static Imports
- When You Should Use Static Imports

### **Lesson: The Date/Time API**

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

## **Module 6: Collections and Generics**

### **Lesson: Introduction to Generics**

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

### **Lesson: Collections**

- Characterizing Collections
- Collection Interface Hierarchy
- Iterators
- The Set Interface
- The List Interface
- Queue Interface
- Map Interfaces
- Using the Right Collection
- Collections and Multithreading

## **Module 7: Java Lambda Expressions and Streams**

### **Lesson: Introduction to Lambda Expressions**

- Functional vs OO Programming
- Anonymous Inner-classes
- Lambda Expression Syntax
- Functional Interfaces
- Method references
- Constructor references

### **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

### **Lesson: Collectors**

- Creating Collections from a Stream
- Group elements in the Stream

- Multi-level grouping of elements
- Partitioning Streams

## **Module 8: Multithreading and Concurrency**

### **Lesson: Multithreading**

- Principles of Multithreading
- Creating a Threaded Class
- Basic Features of the Thread Class
- Thread Scheduling
- Thread Synchronization

### **Lesson: Concurrent Java**

- Concurrent Locks are Explicit and Flexible
- Executor Interfaces Provide Thread Management
- Challenges for Concurrent Use of Collections
- Concurrent Collections
- Atomic Variables Avoid Synchronization

## **Module 9: Java I/O**

### **Lesson: File System Access**

- The File Class
- File Utility Methods
- Lesson: Java I/O
- The Java I/O Mechanism
- Subclasses Accessing Real Data
- Filter Classes
- New File IO - NIO
- NIO Overview

## **Module 10: Java Application Development**

### **Lesson: Introduction to Annotations**

- Annotations Overview
- Working with Java Annotations

### **Lesson: Java Data Access JDBC API**

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

- Executing Inserts, Updates, and Deletes
- Controlling Transactions and Concurrency