Document Generated: 11/22/2024

Learning Style: Virtual Classroom

Provider: Microsoft

Difficulty: Beginner

Course Duration: 5 Days

Introduction to Java 9/10 Programming for Developers New to OO Programming (such as C, Mainframe, 4GL) (TT2120-J9/10)

About this course:

This training course of Java 9 and OO Programming Essentials is a hands-on, five days course designed for engineers who have practically zero earlier information on object-oriented programming dialects, (for example, those working on (C, 4GL, COBOL, and so forth.) Throughout the course, understudies gain proficiency with the prescribed procedures for writing object-oriented projects in Java 9, using sound advancement methods, new improved highlights for better execution, and new capacities for addressing quick application advancement. Exclusive accentuation is put on object-oriented ideas and best practices.

Note: Learners with earlier Object-Oriented introduction and foundation, (for example, C++, C#, Smalltalk, and so forth.) ought to consider the Programming of TT2100-J9 Java 9 for Object-Oriented (OO) Experienced Developers as another option.

The normal compensation of a Java Developer is $90,992 every year.

Course Objective:

- Comprehend about OO programming and what the benefits of OO are in this current time
- Work with classes, objects, and OO usage
- Work with the Java 9 particular framework (Project Jigsaw)
- Take benefit of the Java tooling that is accessible with the programming condition being utilized in the class.
- Comprehend the basics of the Java language, and its significance uses, weakness, and strength.
- Comprehend the essential ideas of OO, for example, polymorphism, inheritance, encapsulation, and abstraction.
- Comprehend the rudiments of the Java language and how it identifies with OO programming and the Object Model
- Comprehend and use generics, collections, autoboxing, and enumerations
- The Modular framework (Project Jigsaw)
- Figure out how to utilize Java exception handling highlights
- Comprehend and use inheritance, classes, and polymorphism
- Procedure a lot of information using the Stream API and Lambda expressions.
- Utilize the JDBC API for access to the database.
- Private techniques in interfaces

Audience:

This course is an introductory-level programming course of Java, intended for experienced engineers who wish to find a good pace with Java, or who need to reinforce sound Java coding practices.

Prerequisite:

Participants ought to have earlier handy programming involvement with another language.

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

Lesson: The Eclipse Paradigm

- Workbench and Workspace
- Views
- Editors
- Perspectives
- Projects

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

Module 3: OO Concepts

Lesson: Object-Oriented Programming

- Real-World Objects
- Classes and Objects
- Object Behavior
- Methods and Messages

Lesson: Inheritance, Abstraction, and Polymorphism

- Encapsulation
- Inheritance
- Method Overriding
- Polymorphism

Module 4: Essential Java Programming

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

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 5: Object Oriented Development

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

Module 6: Exception Handling

Lesson: Introduction to exception handling

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

Lesson: Exceptions

- Creating Your Own Exceptions
- Handling Multiple Exceptions
- Automatic Closure of Resources

Module 7: Java Developer's Toolbox

Lesson: Utility Classes

- Wrapper Classes
- Autoboxing/Unboxing
- Working with Dates
- Enumeration Syntax
- Using Static Imports

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

Lesson: Formatting Strings

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

Module 8: Advanced Java Programming

Lesson: Introduction to Generics

- Generics and Subtyping
- Bounded Wildcards
- Generic Methods

Lesson: Lambda Expressions and Functional Interfaces

- Functional vs OO Programming
- Lambda Expression Syntax
- Functional Interfaces
- Type Inference
- Method references

Module 9: Working with Collections

Lesson: The Collection API

- Characterizing Collections
- Collection Interface Hierarchy
- The Set, List and Queue Interfaces
- Map Interfaces

Lesson: Using Collections

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

Module 10: Stream API

Lesson: Streams

- Processing Collections of data
- Filtering and sorting collection data
- Find elements in Stream
- Numeric 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 11: The Java Module system (Jigsaw)

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

Module 12: 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

Lesson: Introduction to Annotations (optional)

- Annotations Overview
- Working with Java Annotations

Credly Badge:

Display your Completion Badge And Get The Recognition You Deserve.

Add a completion and readiness badge to your Linkedin profile, Facebook page, or Twitter account to validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

Find Out More or See List Of Badges