

**Document Generated: 11/13/2024**

**Learning Style: Virtual Classroom**

**Provider: Java**

**Difficulty: Beginner**

**Course Duration: 5 Days**

## **Introduction to Programming, OO and Java 8 Essentials for Non-Programmers (TT2000)**



## About this course:

This is a beginner level course for the people having no or very little prior programming skills. The **Introduction to Programming, OO and Java 8 Essentials** course helps you learn the best practices and skills of OOP (object-oriented programming) specifically focusing on Java 8. When enrolled you will get to learn the basic principles and skills that help throughout the application development phase. You will be learning in depth object oriented concepts and the best practices used in the market. You will be taught how to utilize these skills and how to enhance your code quality as well.

A senior Java developer these days earn anywhere between \$80,000 to \$100,000.

This course can help you and guide you towards that direction. As this is a basic training course which can act as bedrock towards that goal of becoming a senior Java developer. People who already have Java skills can also take this course as it helps refresh their long forgotten or unused basic concepts of Java.

## Course Objective:

- To provide a basic understanding of Java concepts
- To provide a basic understanding of object oriented programming and its benefits
- To showcase implementation of object oriented (OO) concepts in real world applications
- To provide understanding of encapsulation, abstraction, inheritance and polymorphism
- To build logical mindset that all programming languages require
- Understand error exception handling in Java
- Understand collections, classes, generics, enumerations and autoboxing
- Understand Java tooling and take full advantage of it

## Audience:

This course is for beginner level students who have little to none knowledge about Java or any other programming language. IT engineers who are looking to start their career in Java, or people who already have skills in other programming languages and wanting to switch to Java. Even the people who are already familiar with Java, this course can help boost their knowledge or refresh their fundamental concepts.

## **Prerequisite:**

People who are taking this course need only to be aware of using a computer system properly, and have basic know how of windows environment. Most people taking this course are on the path of becoming a Java developer. This course helps and guides you in becoming a well practiced Java developer.

## **Course Outline:**

### **Module 1: Introduction to Computer Programming**

#### **Lesson: Introduction to Programming**

- What is a Computer Program?
- Problem Solving using Programs
- The Spectrum of Programming Languages

#### **Lesson: Requirements to Production**

- Starting with Requirements
- Designing a Program
- Implementing the Program
- Converting Source Code to Machine Language
- Executing Machine Language
- Testing to Requirements
- Deploying to Production
- Exercise: Business Use Case: Payroll

#### **Lesson: Programming Tools**

- Design Tools
- Development Tools
  - Code Editors
  - Documentation
  - Compilers
  - Debuggers
- Execution Tools
  - Runtime Environments
  - Interpreters
  - Libraries
- Testing Tools

### **Module 2: Programming Fundamentals**

#### **Lesson: Thinking About Objects**

- Real-World Objects
- Programming Objects

- State and Functionality
- Classes as a Template for Instances
- Working with a Class
  - A Class Diagram
  - Implementing the Class
  - Compiling the Class
  - Executing the Class
- Exercise: Modelling Employees for Payrol

### **Lesson: Program Basics**

- Different Languages, Different Syntax
- Features of Any Program
  - Entry Point
  - Instructions
  - Exit Points
- Why Break a Program Apart?

### **Lesson: Programming Constructs**

- Handling Data:
  - Variables and constants
  - Arrays, Lists and Queues
- Handling Functionality
  - Methods
  - Pre-defined Functionality
- Handling Flow Control
  - Decision structures
  - Repetition (loops)
- Exercise: Implementing Payroll

## **Module 3: 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 ColorPicker and MemoryViewer

## **Lesson: The Eclipse Paradigm**

- Workbench and Workspace
- Views
- Editors
- Perspectives
- Projects
- Tutorial: Working with Eclipse Neon

## **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

## **Module 4: 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 5: Getting Started with Java**

### **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

## **Lesson: Language Statements**

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

## **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

## **Module 6: 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

- Exercise: Field Test

## **Lesson: Using Arrays**

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

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

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

## **Module 7: Advanced Java Programming**

### **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

### **Lesson: Exceptions**

- Exception Architecture
- Handling Multiple Exceptions
- Automatic Closure of Resources
- Creating Your Own Exceptions
- Throwing Exceptions

- Checked vs. Unchecked Exceptions
- Exercise: Exceptions

## **Module 8: Java Developer's Toolbox**

### **Lesson: Utility Classes**

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

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

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

### **Lesson: Formatting Strings**

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

## **Module 9: Collections and Generics**

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

- Generics and Subtyping
- Bounded Wildcards
- Generic Methods
- Legacy Calls to Generics
- When Generics Should Be Used
- Exercise: ShoppingCart

### **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
- Exercise: Collections Poker

## 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](#)