**Document Generated: 12/22/2024**

**Learning Style: Virtual Classroom**

**Provider: Java**

**Difficulty: Beginner**

**Course Duration: 5 Days**

# Introduction to Programming & Coding | Java Basics for Non-Developers (TTCODE101-J)

## About this course:

This course is a beginner level course for students who are interested in diving into java development. This course teaches all the fundamentals and basic concepts of programming in general and also focuses on some key features of Java as well and enlightens the students as to why Java is still one of the most relevant programming languages.

The course focuses mainly on Object-oriented concepts in Java along with an emphasis on the structural integrity of applications and why it is important to get the structure of application right from the very beginning.

Some high-level concepts like memory allocation, standard I/O, algorithm building and application security are also highlighted in this course. Throughout this course, you will gain experience in developing real world applications using Java, and during this process you will face the real world problems that are faced by developers and you will also be taught how to handle them.

There is a lot of emphasis on the structure of the application in this course, as new developers should learn to get their coding patterns and development structure right before anything else. This also includes setting/configuring the environment and the environment variables as well.

The salary of a Java developer is around $90,000 in today's market and the need for Java developers is ever increasing.

## Course objective:

Here are the main objectives of this course and what this course plans to teach.

- Basic concepts of programming
- Fundamentals of Java
- Setting up and getting familiar with development environment, including IDE's
- Understanding object oriented development
- Learning how to make classes, methods and objects
- Learning loops, their importance and usage
- Use of conditions and regular expressions
- Learning importance of lists and arrays
- Basic GUI skills
- Error and exception handling
- Formatting output and formatter class

## Audience:

The ideal audience for this course are the people who want to gain programming skills. More specifically people who want to develop or enhance their skills in Java development.

Students are not age-specific, anyone of any age willing learn programming can enroll and get started.

## Prerequisite:

Not many technical prerequisites required, only these:

- Familiar with Windows environment
- Wants to learn how things work within any application
- Basic idea of number theory
- Is able to follow suggested steps
- Can think their way out of technical problems
- Eager to solve problems and able to visualize problem solving skills

## Course Outline:

### Fundamentals of the Program Development Cycle

- Computer Architecture
- The Notion of Algorithms
- Source Code vs. Machine Code
- Compile-Time vs. Run-Time
- Software Program Architecture
- Standalone
- Client/Server
- Distributed
- Web-Enabled
- IDE (Interactive Development Environment) Concepts

### Application Development Fundamentals

- Structure of a Java Program
- Memory Concepts
- Fundamental Data Type Declarations
- Fundamental I/O Concepts
- Fundamental Operators
- Arithmetic Operators

- Logical Operators
- Precedence and Associativity
- Building and Deploying a Java Program

## Introduction to Classes and Objects

- Classes, Objects and Methods
- Object Instances
- Declaring and Instantiating a Java Object
- Declaring Methods
- set and get Methods
- Initiating Objects with Constructors
- Primitive Types vs. Reference Types

## Flow Control

- Conditional Constructs
- Looping Constructs
- Counter-Controlled Repetition
- Sentinel-Controlled Repetition
- Nested Control Constructs
- break and continue Statements
- Structured Programming Best Practices

## Writing Methods (Functions)

- Static vs. Dynamic Allocation
- Declaring Methods
- Declaring Methods with Multiple Parameters
- Method-Call Stack
- Scope of Declarations
- Argument Promotion and Casting
- Designing Methods for Reusability
- Method Overloading

## Arrays

- Purpose of Arrays
- Declaring and Instantiating Arrays
- Passing Arrays to Methods
- Multidimensional Arrays
- Variable-Length Argument Lists
- Using Command-Line Arguments
- Using Environment Variables

## Deeper Into Classes and Objects

- Controlling Access to Class Members
- Referencing the Current Object Using this
- Overloading Constructors

- Default and No-Argument Constructors
- Composition of Classes
- Garbage Collection and Destructors
- The finalize Method
- Static Class Members

## Defining Classes Using Inheritance

- Superclasses and Subclasses
- Advantages of Using Inheritance
- protected Class Members
- Constructors in Subclasses

## Increasing Convenience by Using Polymorphism

- Purpose of Polymorphic Behavior
- The Concept of a Signature
- Abstract Classes and Methods
- final Methods and Classes
- Purpose of Interfaces
- Using and Creating Interfaces
- Common Interfaces of the Java API

## Files and Streams

- Concept of a Stream
- Class File
- Sequential Access
- Object Serialization to/from Sequential Access Files
- Additional java.io Classes

## Fundamental Searching and Sorting

- Introduction to Searching Algorithms
- Linear Search
- Binary Search
- Introduction to Sorting Algorithms
- Selection Sort
- Insertion Sort
- Merge Sort

## Fundamental Data Structures

- Dynamic Memory Allocation
- Linked Lists
- Stacks
- Queues
- Trees

## Exception Handling

- Types of Exceptions
- Exception Handling Overview
- Exception Class Hierarchy
- Extending Exception Classes
- When to Throw or Assert Exceptions

**Formatted Output**

- printf Syntax
- Conversion Characters
- Specifying Field Width and Precision
- Using Flags to Alter Appearance
- Printing Literals and Escape Sequences
- Formatting Output with Class Formatter

**Strings, Characters and Regular Expressions**

- Fundamentals of Characters and Strings
- String Class
- String Operations
- StringBuilder Class
- Character Class
- StringTokenizer Class
- Regular Expressions
- Regular Expression Syntax
- Pattern Class
- Matcher Class

**Fundamental GUI Programming Concepts**

- Overview of Swing Components
- Displaying Text and Graphics in a Window
- Event Handling with Nested Classes
- GUI Event Types and Listener Interfaces
- Mouse Event Handling
- Layout Managers

## 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