**Document Generated: 11/05/2026**

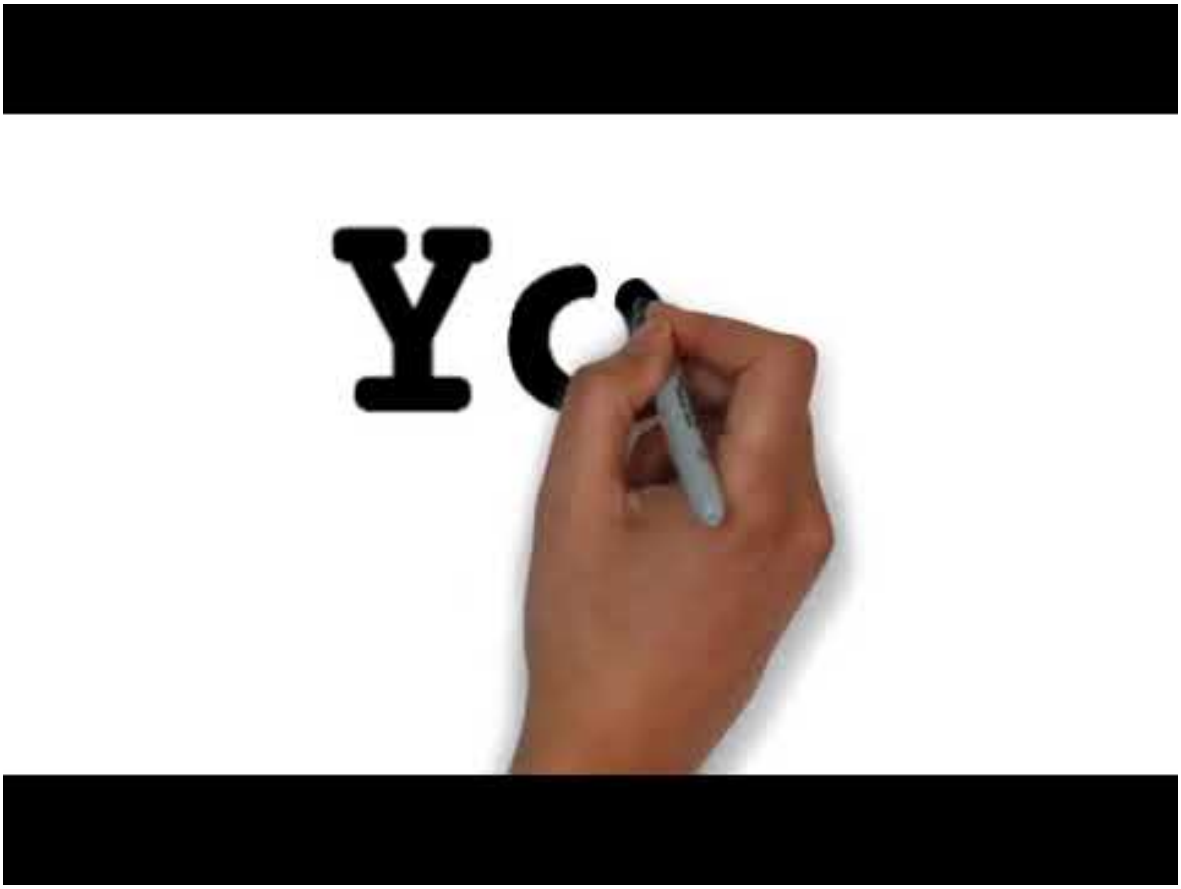
**Learning Style: On Demand**

**Technology: Java**

**Difficulty: Beginner**

**Course Duration: 17 Hours**

# From 0 to 1: Learn Java Programming - Live Free, Learn To Code



**About this course:**

This course will use Java and an Integrated Development Environment (IDE). Never fear, we have a detailed video on how to get this downloaded and set up.Hundreds of lines of source code, and hundreds of lines of comments - just download and open in your IDE! This is a Java course for everyone. Whether you are a complete beginner (a liberal arts major, an accountant, doctor, lawyer) or an engineer with

some programming experience but looking to learn Java - this course is right for you.The course is accessible because it assumes absolutely no programming knowledge, and quickly builds up using first principles aloneEven so, this is a serious Java programming class - the gradient is quite steep, and you will go from absolute beginner to an early intermediate level. The course is also quirky. The examples are irreverent. Lots of little touches: repetition, zooming out so we remember the big picture, active learning with plenty of quizzes. There's also a peppy soundtrack, and art - all shown by studies to improve cognition and recall.

The average salary for a Java developer is **$69,722** per year.

**Course Objectives:**

After completing this course, students will be able to:

- Write Java programs of moderate complexity and sophistication (at an early to middling intermediate level).
- Manage concurrency and threading issues in a multi-threaded environment.
- Use Reflection, Annotations, Lambda functions and other modern Java language features.
- Understand the Model-View-Controller paradigm, the Observer and Command Design patterns that are at the heart of modern UI programming.
- Understand the nuances of Java specific constructs in serialisation, exception-handling, cloning, the immutability of strings, primitive and object reference types.
- Understand Object-Oriented programming concepts at the level where you can have intelligent design conversations with an experienced software engineer.
- Create and modify files (including Excel spreadsheets) and download content from the internet using Java.
- Build serious UI applications in Swing.
- Gain a superficial understanding of JavaFX and Properties and Bindings.

**Audience:**

Folks with zero programming experience - liberal arts majors, doctors, accountants, lawyers. Engineering students from non-CS majors looking to learn fairly serious programming. Computer Science students or software engineers with no experience in Java, but experience in Python, C++ or even C#. You might need to skip over some bits, but in general the class will still have new learning to offer you.

**Prerequisites:**

No prior programming experience needed.

**Suggested prerequisites courses:**

- Python Programming for Beginners
- C++ for Beginners

**Career path:**

App Development…

## Course Outline:

### You, Us & This Course

- Introduction

### Coding Is Like Cooking

- Coding is like Cooking
- Functions as Food Processors
- Installing Java and Setting up an IDE
- Variables - Primitives, Objects, Null
- Hello World! (Drill)
- Don't Jump Through Hoops, Use Loops
- Arrays and Lists - Drill
- Maps
- Maps - Drill
- Resource Material

### An Object-Oriented State of Mind

- The Evolutionary Arc from C, through C++, to Java
- Objects are like puppies
- Classes and Objects: An Object-Oriented State of Mind
- Resource Material
- Static Variables and Methods
- Resource Material
- Access Modifiers
- Resource Material
- Classes and Objects: A Simple Example I
- Resource Material
- Classes and Objects: A Simple Example II
- Is-A Inheritance - setting up a class hierarchy
- Resource Material
- Is-A Inheritance - parent class, child class relationship
- Resource Material
- Runtime Polymorphism
- Resource Material
- The Object Base Class
- Resource Material
- Interfaces : Introduction
- Resource Material
- Interfaces vs Abstract Base Class
- Resource Material
- Interfaces In Detail
- Resource Material

- Interface Default Methods: Avoid Backward Compatibility Nightmares
- Resource Material
- Interfaces and Inheritance in Action
- Resource Material

**Java Language Constructs: The Best Thing Since Sliced Bread**

- Exceptions
- Resource Material
- Object Identity: == and .equals()
- Resource Material
- Generics: Type Safety and Code Re-use
- Resource Material
- Collections: Containers for all purposes
- Resource Material
- Generic Containers - much cooler
- Resource Material
- Ordered Associative Containers - The Tree Map
- Resource Material
- Inner Classes: Horses for Courses - Static vs Non-Static
- Resource Material
- Inner Classes: Horses for Courses - Anonymous and Local
- Resource Material
- A File is like a Barrel
- A Serious Java Application: Our First
- Resource Material
- A Serious Java Application: Parsing Stock Ticker Data - I
- Resource Material
- A Serious Java Application: Parsing Stock Ticker Data - II
- A Serious Java Application: Parsing Stock Ticker Data - III
- A Serious Java Application: Parsing Stock Ticker Data - IV
- A Serious Java Application: Parsing Stock Ticker Data - V

**Threading and Concurrency: A Lot Going On All At Once**

- Threading, and Why It Matters
- Resource Material
- Threading: Old school vs New school
- Resource Material
- Traditional Threading Support
- Resource Material
- Threading and Concurrency: A lot going on at once
- Resource Material
- Synchronized Methods, Livelock, Deadlock and Starvation
- Resource Material
- Working with threads - and our first synchronization bug!
- Resource Material
- Threads: Object locking and static variables in action
- Digging Deeper into Callables and Futures
- Resource Material

- Threading: New school fun in action
- Resource Material
- Threading: Executors are pretty cool

**Functional Programming: Crossover Hits**

- Functional Programming: A State of Mind
- Resource Material
- Lambda Functions: A Crossover Hit
- Resource Material
- Lambda Functions: No more boiler-plate code
- Resource Material
- Lambda functions for real - in code
- Resource Material

**Recursion, Reflection, Annotations, Dates, Packages and Jars**

- Recursion
- Reflection: Not a Dark Art
- Resource Material
- Reflection: A real use case
- Resource Material
- Annotations: Notes to Self can be Super-Useful
- Resource Material
- Working With Dates in Java
- Resource Material
- Jars: Not As Boring as They Might Seem
- Resource Material
- Packages
- Resource Material
- Packages and Jars in Action
- Serialisation
- Resource Material
- Cloning
- Resource Material
- The Immutability of Strings
- Resource Material

**UI Programming: Swing, MVC, and JavaFX**

- Frameworks - A State of Mind
- Resource Material
- The Basic Idea of Swing
- Resource Material
- Swing and the Model-View-Controller Paradigm
- Resource Material
- The Mechanics of Swing - I
- Resource Material
- The Mechanics of Swing - II
- Resource Material

- A Serious Swing App: News Curation Browser - I
- Resource Material
- A Serious Swing App: News Curation Browser - II
- A Serious Swing App: News Curation Browser - III
- A Serious Swing App: News Curation Browser - IV
- Observers, Events and Listeners
- Resource Material
- Actions, And The Command Design Pattern
- Resource Material
- A Peek at JavaFX - I
- Resource Material
- A Peek at JavaFX - II
- Resource Material
- Properties And Bindings
- Resource Material
- Properties And Bindings In Action
- Resource Material

**Some Interview Problems For Practice!**

- Starting up - palindromes and points within a distance
- Resource Material
- Play the Game Of Life and Break A Document Into Chunks
- Resource Material
- Implement paint fill to color a region on screen

## 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