skillsoft global knowledge™

GETTING STARTED WITH PROGRAMMING, OO AND JAVA BASICS FOR NON-DEVELOPERS

Course Code: 101065

Learn to Think Like a Programmer: Jumpstart your Java coding skills in this engaging, skill-focused programming course

Getting Started with Programming, OO and Java Basics for Non-Developers is a fast-paced, five day course designed to provide you with a first look at how to code in Java to a very basic level. You'll gain light hands-on programming experience, while you begin your journey to develop a programmer's mindset.

Throughout the course you'll explore the intricacies of the application development cycle, program structure, and language syntax. You'll learn and practice core coding skills including fundamental OO concepts, vital programming constructs, string and character manipulation, dynamic memory allocation, standard input/output, and exception handling. You'll also explore the power of inheritance and polymorphism as you learn to define specialized object implementations. The course also covers processing command line arguments and environment variables, empowering you to set up your own development environment to create flexible, user-friendly programs. With a wealth of hands-on lab exercises, you'll practice and refine your newly acquired skills.

Becoming a modern software developer is like learning a new language; it requires study, practice, and dedication well beyond this course to apply your new skills effectively. While this five day program won't transform you into an experienced developer, it will lay a solid foundation in coding basics using Java, while teaching you to think like a programmer. Although this course is technical in nature, our instructors will guide you every step of the way, providing a supportive environment for you to explore, ask questions, and prepare for your next learning milestones.

NOTE: Although this course is geared for non-developers, it is helpful for attendees to have a somewhat technical background and to be comfortable working with computers, having the ultimate goal of becoming a Java software developer. This course uses Java 21, which also covers the fundamental concepts and techniques in Java 11 and 17. This course is suited for Java 11, Java 17 and Java 21 skills development. Earlier

versions are available. Please inquire for options.

What You'll Learn

Working in a hands-on learning environment, guided by our expert team, you'll explore:

- The basic programming constructs that all programming languages share
- Fundamental programming concepts, such as variables, data types, loops, and conditional statements.
- Object-oriented programming principles, including encapsulation, inheritance, and polymorphism.
- How to handle problems that might occur during the execution of a Java application.
- How to use Java libraries and APIs for common tasks, such as file I/O, data manipulation, and networking.
- Implementing exception handling and debugging techniques to ensure robust and reliable code.
- Utilize industry-standard tools and Integrated Development Environments (IDEs) to efficiently write, test, and deploy Java applications.
- Best practices for code organization, documentation, and version control to enhance collaboration and maintainability.

Who Needs to Attend

This course is designed for anyone new to coding and eager to learn how to program using Java. While it's tailored for beginner-level students, please note that it is technical in nature. If you're transitioning from a non-technical role to coding for the first time, feel free to reach out to us for additional guidance or pre-training suggestions that can better prepare you for this course. Our goal is to make your learning experience exciting, challenging, and valuable, without being overwhelmed.

Attendees might include:

- Technically minded attendees who want or who want to begin the process of becoming an OO application developer
- Technical team members from non-development roles, re-skilling to move into software and application development roles within an organization
- Recent college graduates looking to apply their college experience to programming skills in a professional environment, or perhaps needing to learn the best practices and standards for programming within their new organization
- Technical managers tasked with overseeing programming teams, or development projects, where basic coding knowledge and exposure will be useful in project oversight or communications needs

Prerequisites

- **Basic computer literacy:** Familiarity with computer operating systems, file management, and general navigation to ensure a smooth learning experience.

- **Foundational knowledge of IT concepts:** Understanding of essential IT terminologies and concepts, such as computer networks, software applications, and data storage.
- **Analytical thinking:** Ability to analyze problems and think critically to develop logical solutions, fostering a programmer's mindset.
- **Attention to detail:** A keen eye for detail, ensuring the ability to spot errors and maintain code quality throughout the learning process.

GETTING STARTED WITH PROGRAMMING, OO AND JAVA BASICS FOR NON-DEVELOPERS

Course Code: 101065

CLASSROOM LIVE	$3,374 CAD	5 Day

Classroom Live 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

Classroom Live Labs

This hands-on course focuses on ‘learning by doing’, combining expert lecture, practical demonstrations and group discussions with plenty of machine-based real-world programming labs and exercises. Student machines are required.

skillsoft global knowledge™

GETTING STARTED WITH PROGRAMMING, OO AND JAVA BASICS FOR NON-DEVELOPERS

Course Code: 101065

VIRTUAL CLASSROOM LIVE	$3,374 CAD	5 Day

Virtual Classroom Live Outline

1. **Overview of Computer Programming**
   - Explain what a program is
   - Explain why there are different types of languages
   - Explain what a compiler is
   - Explain what an interpreter is
2. **Features of a Program**
   - Understand what the entry and exit points of an application are
   - Explain what variables are
   - Explain what programming instructions are
   - Explain what errors and exceptions are
   - Understand what programming algorithms are
3. **Software Development Life Cycle**
   - Explain the purpose of the software development life cycle
   - Explain what each phase is for
   - Explain the difference between the software development life cycle and a methodology
4. **Thinking in Objects**
   - Understand the difference between a class and an object
   - Deconstruct an object into attributes and operations
   - Map an object to a class
   - Define inheritance
5. **The Java Platform**
   - Introduce the Java Platform
   - Explore the Java Standard Edition
   - Discuss the lifecycle of a Java Program

- Explain the responsibilities of the JVM
- Executing Java programs
- Garbage Collection
- Documentation and Code Reuse

6. **Using the JDK**
   - Explain the JDK's file structure
   - Use the command line compiler to compile a Java class
   - Use the command line Java interpreter to run a Java application class
7. **Using the IntelliJ IDE**
   - Introduce the IntelliJ IDE
   - The Basics of the IntelliJ interface
   - IntelliJ Projects and Modules
   - Creating and running Java applications
   - Tutorial: Working with your IDE IntelliJ 2023.2 (Community Edition) or Eclipse
8. **Writing a Simple Class**
   - Write a Java class that does not explicitly extend another class
   - Define instance variables for a Java class
   - Create object instances
   - Primitives vs Object References
   - Implement a main method to create an instance of the defined class
   - Java keywords and reserved words
9. **Adding Methods to the Class**
   - Write a class with accessor methods to read and write instance variables
   - Write a constructor to initialize an instance with data
   - Write a constructor that calls other constructors of the class to benefit from code reuse
   - Use the this keyword to distinguish local variables from instance variables
10. **Object-Oriented Programming**
    - Real-World Objects
    - Classes and Objects
    - Object Behavior
    - Methods and Messages
11. **Language Statements**
    - Arithmetic operators
    - Operators to increment and decrement numbers
    - Comparison operators
    - Logical operators
    - Return type of comparison and logical operators
    - Use for loops
    - Switch Expressions
    - Switch Expressions and yield
12. **Using Strings and Text Blocks**
    - Create an instance of the String class
    - Test if two strings are equal

- Perform a case-insensitive equality test
- Contrast String, StringBuffer, and StringBuilder
- Compact Strings
- Text Blocks
- Unicode support

13. **Fields and Variables**
    - Discuss Block Scoping Rules
    - Distinguish between instance variables and method variables within a method
    - Explain the difference between the terms field and variable
    - List the default values for instance variables
    - Final and Static fields and methods
14. **Specializing in a Subclass**
    - Constructing a class that extends another class
    - Implementing equals and toString
    - Writing constructors that pass initialization data to parent constructor
    - Using instanceof to verify type of an object reference
    - Overriding subclass methods
    - Pattern matching for instanceof
    - Safely casting references to a more refined type
15. **Using Arrays**
    - Declaring an array reference
    - Allocating an array
    - Initializing the entries in an array
    - Writing methods with a variable number of arguments
16. **Records**
    - Data objects in Java
    - Introduce records as carrier of immutable data
    - Defining records
    - The Canonical constructor
    - Compact constructors
17. **Java Packages and Visibility**
    - Use the package keyword to define a class within a specific package
    - Discuss levels of accessibility/visibility
    - Using the import keyword to declare references to classes in a specific package
    - Using the standard type naming conventions
    - Introduce the Java Modular System
    - Visibility in the Java Modular System
18. **Utility Classes**
    - Introduce the wrapper classes
    - Explain Autoboxing and Unboxing
    - Converting String representations of primitive numbers into their primitive types
    - Defining Enumerations

- Using static imports
- Introduce the Date/Time API
- LocalDate / LocalDateTime etc.
- Apply text formatting
- Using System.out.printf

19. **Inheritance and Polymorphism**
    - Write a subclass with a method that overrides a method in the superclass
    - Group objects by their common supertype
    - Utilize polymorphism
    - Cast a supertype reference to a valid subtype reference
    - Use the final keyword on methods and classes to prevent overriding
20. **Interfaces and Abstract Classes**
    - Define supertype contracts using abstract classes
    - Implement concrete classes based on abstract classes
    - Define supertype contracts using interfaces
    - Implement concrete classes based on interfaces
    - Explain advantage of interfaces over abstract classes
    - Explain advantage of abstract classes over interfaces
21. **Introduction to Exception Handling**
    - Introduce the Exception architecture
    - Defining a try/catch blocks
    - Checked vs Unchecked exceptions
22. **Introduction to Generics and Collections**
    - Introduce Generics and Subtyping
    - Explain Bounded Wildcard
    - Generics Methods
    - Provide an overview of the Collection API
    - Review the different collection implementations (Set, List and Queue)
    - Explore how generics are used with collections

Virtual Classroom Live Labs

This hands-on course focuses on ‘learning by doing’, combining expert lecture, practical demonstrations and group discussions with plenty of machine-based real-world programming labs and exercises. Student machines are required.

Jun 23 - 27, 2025 | 10:00 AM - 6:00 PM EST

Aug 11 - 15, 2025 | 10:00 AM - 6:00 PM EST

Sep 22 - 26, 2025 | 10:00 AM - 6:00 PM EST

Nov 3 - 7, 2025 | 10:00 AM - 6:00 PM EST

GETTING STARTED WITH PROGRAMMING, OO AND JAVA BASICS FOR NON-DEVELOPERS

Course Code: 101065

PRIVATE GROUP TRAINING	5 Day

Visit us at www.globalknowledge.com or call us at 1-866-716-6688.

Date created: 5/9/2025 1:25:51 AM