

## Java Programming 1 / Java Developer

**Course Duration** : 5 days

### Course Overview

Java is everywhere and is as popular as ever. This hands-on course, delivered using Java 11, endeavours to provide the fundamental skills you need to become a Java Developer. We'll have you build a series of realistic apps focusing on object-oriented principles from the start.

Note that this is not a beginners course. You are expected to be familiar with programming concepts and have a little coding experience. If not you should consider our Learn to Code with Java course instead.

Exercises and examples are used throughout the course to give practical hands-on experience with the techniques covered.

### Target Audience

The Java Programming 1 / Java Developer course is aimed at anyone with a little coding experience seeking to become a Java Developer, and existing developers seeking to migrate to Java.

### Prerequisites

Delegates attending this course should be able to define general programming concepts including compilation and execution, variables, arrays, sequence, selection and iteration, navigate the filesystem (on the command line ideally), edit and save text files and browse the web. This knowledge can be obtained by attendance on the pre-requisite Introduction to Programming course.

### Objectives

- Construct a Java app; perform basic IO
- Declare classes; instantiate and use objects
- Differentiate between primitives and references; exploit the standard data types
- Code decisions and loops
- Group data using ArrayList and HashMap
- Make the case for and code static fields and methods
- Manage numbers and perform computations using the Math and BigDecimal classes
- Process Strings and perform pattern matching
- Manage exceptions
- Hide an object's data and its internal workings
- Build objects that are composed of others
- Build objects that inherit state/behaviour from others
- Build flexible and scalable apps by exploiting abstraction and polymorphism
- Read from and write to files and databases
- Build a standalone executable from one or more modules



## Course Content

### DAY 1

#### Session 1: PLATFORM, APP STRUCTURE, & BASIC IO

The Java language  
The Java platform  
App structure incl. modules, packages, and classes (brief)  
The main method  
Reading from stdin and writing to stdout  
Reading from and writing to a text file  
Compilation and execution

#### Session 2: CLASSES & OBJECTS

What is an object  
What is a class  
Object-oriented apps  
Classes  
Fields  
Constructors  
Methods  
Objects  
Instantiation  
Getting and setting fields  
Calling methods

#### Session 3: VARIABLES & STANDARD DATA TYPES

Instance fields  
Default values  
Local variables incl. parameters  
Primitives  
Literals  
Default types  
Casting  
References  
Arrays (brief)  
Strings (brief)  
Primitive wrappers  
Constants  
Type inference  
Varargs  
Naming conventions

### DAY 2

#### Session 4: DECISIONS & LOOPS

Relational and conditional operators  
If statements  
Switch statements  
The ternary operator  
While loops  
For loops (both kind)  
Branching statements (break, continue, return)

### Session 5: ESSENTIAL COLLECTIONS

What is an array  
List vs. array  
Map vs. list  
Generic classes (brief)  
Arrays  
Declaration and initialisation  
Setting and getting elements  
Sizing  
Iterating  
Multi-dimensional arrays  
Command line arguments  
The ArrayList and HashMap classes  
Imports  
Adding elements incl. autoboxing  
Membership testing  
Accessing elements  
Changing elements  
Removing elements  
Sizing  
Iterating

### Session 6: STATIC MEMBERS & ENUMS

What does it mean to be static  
Static fields  
Static methods  
Static context  
Static imports  
Enums

### Session 7: NUMBERS & MATHS

Strings to numbers  
Arithmetic and unary operators  
The Math class  
The BigDecimal class  
Number formatting

### DAY 3

#### Session 8: STRINGS & REGEX

What is a String  
Immutable objects  
The String Pool  
Strings  
Creation  
Escape characters  
Concatenation  
Format Strings  
Length  
Transformative methods  
Search methods  
Comparative methods  
Java 11 methods  
StringBuilder  
Pattern matching  
Regular expressions  
The Pattern and Matcher classes

### Session 9: EXCEPTION HANDLING

- What is an exception
- Checked vs. unchecked exceptions
- Exceptions and the call stack
- Try, catch, and finally
- Handling multiple exception types
- Try with resources
- Throwing exceptions
- Custom exceptions (brief)

### Session 10: ENCAPSULATION

- What is encapsulation
- Access modifiers
- Private fields and public methods
- Getters and setters
- Constructors (for controlling instantiation)
- The this keyword

### DAY 4

#### Session 11: COMPOSITION & AGGREGATION

- What is composition
- What is aggregation
- Composite associations
- Aggregate associations
- Constructor injection (mandatory)
- Setter injection (optional)
- The problem with getter methods (refs. vs. copies)

#### Session 12: INHERITANCE, ABSTRACTION, & POLYMORPHISM

- What is inheritance
- The purpose of inheritance
- Extending a superclass
- Shadowing
- Overriding
- Subclass constructors
- The super keyword
- Inheritance polymorphism
- Upcasting and downcasting
- The instanceof operator
- Final classes and methods
- The Object class incl. toString, hashCode, and equals (brief)
- What is abstraction
- The purpose of abstraction
- Abstract classes and methods
- Extending an abstract class
- Interfaces
- Implementing one or more interfaces
- Extending and implementing
- Interface polymorphism
- Interface default, static, and private methods

### DAY 5

#### Session 13: MODERN FILE IO

- What is a stream
- The different types of stream
- The Path class
- Reading all bytes/lines from a file
- Writing all bytes/lines to a file
- Reading from a file using a buffered stream
- Writing to a file using a buffered stream
- Parsing a file using a Scanner

#### Session 14: MODULES

- What is a module
- Module descriptor
- Module requires and exports
- Compiling a module
- Using jdeps to determine a module's dependencies
- Compiling a module that requires other modules
- Executing a module
- Using jlink to build a custom JRE

#### Session 15: BASIC JDBC

- What is JDBC
- Loading the driver
- Establishing a connection
- AutoCommit
- Transaction management
- Creating a statement
- PreparedStatement
- CallableStatement
- Batch updates
- Executing a query/update
- Processing a ResultSet
- Type
- Concurrency
- Holdability
- Closing resources