



Training: Capstone Courseware
102 Introduction to Java Programming



TRAINING TERMS

2025-06-02 | 5 days | Virtual Classroom

TRAINING GOALS:

Version 8.0

The five-day timeline of this course explicitly targets less experienced programmers, providing them with a thorough step-by-step introduction to **Java programming**. It lays a firm foundation for further study of Java. There are a large number of example programs and many labs.

An important thrust of this course is to teach programming from an object-oriented perspective. It is often difficult for programmers trained originally in a procedural language to start "thinking in objects." This course introduces object-oriented concepts very early, and Java is developed in a way that leverages its object orientation. Most of the course emphasizes simple classes without inheritance. The last chapter introduces inheritance and polymorphism, along with interfaces and collections.

This revision of the course targets the Java 8 language and Core API.

Learning Objectives

- Learn the basic principles of object-oriented programming
- Learn the essentials of the Java programming language
- Acquire the skills needed to design, code and debug computer programs in the Java language

CONSPECT:

- What is Java?
 - Object Orientation
 - Java Language
 - Java Virtual Machine
 - Java Libraries
 - World Wide Web and Java
 - Java Platforms
 - Java as a First Programming Language





- First Java Programs
 - Hello, World
 - Program Structure
 - Output in Java
 - Variables and Expressions
 - Calculations Using Java
- Introduction to Objects
 - Object Models
 - Classes and Objects
 - State and Behavior
 - Input in Java
 - InputWrapper Class
 - Packages
- Data Types and Operators
 - Strong Typing
 - Integer Data Types
 - Floating Point
 - Conversions Between Types
 - Arithmetic Operators
 - Doing Math in Java
 - Precedence
 - Errors in Integer Arithmetic
- Booleans and Enumerations
 - Boolean Variables
 - Logical and Relational Operators
 - If Tests
 - Compound Statements
 - Switch Statement
 - Enumerated Types
- Loops and Program Flow
 - While Loops
 - Quitting Infinite Loops
 - Curly Braces and Indenting
 - Sentinels and Counters
 - For Loops
 - Loops and If Tests Together





- Nested If Statements
- Objects and Classes
 - Structured Data
 - Classes
 - References
 - Instantiating an Object
 - Assignment of Objects
 - Initialization
 - Garbage Collection
 - Methods
 - Public vs. Private
 - Encapsulation
 - Constructors
 - Formatted Output
- Characters and Strings
 - Char Data Type
 - Character Codes
 - ASCII and Unicode
 - String Class
 - String Input and Output
 - String Methods
- Modular Programming
 - Monolithic Programs
 - Static Variables and Methods
 - Functional Modularity
 - Object Modularity
 - Top-Down and Bottom-Up Development
 - Pass-By-Value and Pass-By-Reference
 - Nested Classes
- Arrays
 - One Dimensional Arrays
 - Subscripts
 - Initializing Arrays
 - Arrays and the For-Each Loop
 - Copy and Assignment
 - Arrays of Objects





- Multidimensional Arrays
- Searching
- Bit Operations
 - Bitwise Operators
 - Truth Tables
 - Hexadecimal Notation
 - Shifting and Masking
- Exception Handling and More Flow Control
 - Exceptions
 - Errors in Integer Arithmetic
 - Floating Point Operations
 - I/O Exceptions vs. Runtime Exceptions
 - Exception Methods
 - Using finally
 - try-with-resources
 - Break
 - Continue
 - Do
- Advanced Java Features
 - Reusable Software Components
 - Abstraction
 - Inheritance
 - Inheritance Hierarchies
 - Polymorphism
 - Abstract Classes
 - Interfaces
 - Collections
 - Iterators
 - Auto-Boxing

REQUIREMENTS:

Good general problem solving skills. Some previous experience programming in a procedural language is essential.





Difficulty level



CERTIFICATE:

The participants will obtain certificates signed by Capstone Courseware.

TRAINER:

Authorized Capstone Courseware Trainer.

