Introduction to Java

Duration: 5 Days | NTJVA102

Note: Available in Java 8, 11, 14, 17, or 21
Class can be done in Eclipse or IntelliJ

Description

Our introduction to Java course is intended for programmers with experience in languages other than Java, but who may or may not have any previous Java experience. It focuses on procedural-coding skills first, and then offers meticulous, in-depth coverage of object-oriented concepts and how to apply them to Java software design and development. The latter part of the course moves from these basic skills into key parts of the Core API, including collections, exception handling and object serialization.

Students come to Java from a wide range of backgrounds, and this course is designed to be as flexible as possible for that reason.

Objectives

- Chiefly, learn to program effectively in the Java language
- Understand the Java software architecture and the design decisions that make Java software portable, efficient, secure and robust
- Learn how to configure a simple Java development environment
- Know the grammar, data types and flow control constructs of the Java language for simple procedural programming
- Understand Java as a purely object-oriented language, and implement software as systems of classes
- Implement and use inheritance and polymorphism, including interfaces and abstract classes
- Design appropriate exception handling into Java methods
- Understand the structure of streams in Java, and learn how to use streams to manage file I/O
- Learn how to use Java Serialization to internalize and externalize potentially complex graphs of objects

Prerequisites

No prior Java experience is required, but students should be experienced programmers in another third-generation (high-level) language. This course can be customized to fit any skill level.

Outline

- The Java Environment
    - Overview of Architecture
    - Java Virtual Machine
    - The Core API
    - Java Runtime Environment

    - Java SDK
    - Java Class Path
    - Portability and Efficiency

- Language Fundamentals
    - Source File Format
    - Application Classes
    - Code Grammar and Expressions
    - Identifiers
    - Literals
    - Operators
    - Expressions
    - Calling Methods

- Data Types
    - Primitive Types
    - Type Conversion
    - Numeric Types
    - Characters and Booleans
    - Enumerations
    - Object References
    - Comparing and Assigning References
    - Strings
    - Arrays

- Flow Control
    - The main Method
    - Calling and Returning from Methods
    - Conditional Constructs
    - Looping Constructs
    - The For-Each Loop
    - Processing Arrays

- Object-Oriented Software
    - Complex Systems
    - Abstraction
    - Classes and Objects
    - Responsibilities and Collaborators
    - Lightweight Design
    - Relationships and Dependencies
    - Visibility

- Classes and Objects
    - Java Classes
    - Constructors and Garbage Collection
    - Naming Conventions and JavaBeans
    - Packages and Modules
    - Relationships Between Classes
    - Visibility
    - Overloading Methods

    - JARs

- Inheritance and Polymorphism in Java
    - Extending Classes
    - Using Derived Classes
    - Polymorphism
    - Overriding Methods

- Using Classes Effectively
    - Class Loading
    - Statics
    - Static Initializers
    - Strings and StringBuffers
    - Controlling Object Creation

- Interfaces and Abstract Classes
    - Separating Interface and Implementation
    - Designing Interfaces
    - Implementing and Extending Interfaces
    - Abstract Classes

- Collections
    - Dynamic Collections
    - Collections vs. Arrays
    - The Collections API
    - Abstraction: The Collection Interface
    - Frequently Used Collections
    - Reading Elements and Downcasting
    - Collecting Primitive Values
    - Algorithmic Programming
    - Iterators
    - Maps
    - Generics and Auto-Boxing

- Exception Handling
    - Exception Handling
    - Throwing Exceptions
    - Declaring Exceptions per Method
    - Catching Exceptions
    - The finally Block
    - Catch-and-Release
    - Chaining Exceptions