Introduction to Java (5 Days)

This Java training course is intended for students without an extensive programming background. It covers most Java syntax elements, concentrating on fundamental and universally useful elements, while providing an overview of many more advanced elements. Students will learn to write useful Java classes, applying Object-Oriented concepts such as inheritance, and create Java programs that work with these classes.

This course has been updated to Java 10.

Early in the class the concept of objects is introduced so that later concepts can be discussed from the perspective of object orientation.

If you have a lot of object-oriented programming experience in a language such as C# or C++, you may want to consider our Java Programming Training for Experienced Programmers class instead.

What you will learn

After completing this course, the student should be able to:

- Learn how Java works.
- Understand the "write once, run anywhere" concept.
- Understand and learn how to create basic Java objects.
- Learn how to implement flow-control concepts in Java.
- Understand Java's package concept and create packages of Java classes.
- Understand Java arrays and write code to create and use arrays.
- Learn how to write Java classes using inheritance.
- Learn how to create and use interfaces.
- Understand and use the concept of polymorphism in an application
- Understand how Java's exception-handling mechanism works and learn how to apply exception-handling to Java applications.
- Understand and use inner classes.
- Learn how to use elements from the Java API library, including the Collections classes.

Prerequisites

Experience in the following *is required* for this Java Fundamentals class:

- Some prior programming experience in a procedural or object-oriented language.

Experience in the following *would be useful* for this Java Fundamentals class:

- Knowledge of Object-oriented Analysis and Design (OOAD) is extremely helpful in Java (and other) programming. We recommend learning OOAD, but you can do so at any stage of your learning. It will make you a better programmer, especially in a team environment.

Course Outline

Java Introduction

- The Java Environment - Overview
- Writing a Java Program
- Obtaining The Java Environment
- Setting up your Java Environment
- Creating a Class That Can Run as a Program
- Useful Stuff Necessary to go Further
- Using the Java Documentation

Java Basics

- Basic Java Syntax
- Data
- Mathematics in Java
- Creating and Using Methods

Java Objects

- Objects
- String, StringBuffer, and StringBuilder
- Creating Documentation Comments and Using javadoc

Comparisons And Flow Control Structures

- Controlling Program Flow
- Additional Loop Control: break and continue
- Classpath, Code Libraries, and Jar files

Arrays

- Arrays
- Enhanced for Loops - the For-Each Loop
- Multi-Dimensional Arrays
- Typecasting with Arrays of Primitives

Inheritance

- Inheritance
- Polymorphism
- Creating a Derived Class
- Example - Factoring Person Out of Employee and Dependent
- Typecasting with Object References
- Other Inheritance-Related Keywords
- Methods Inherited from Object

Interfaces

- Interfaces
- Creating an Interface Definition
- Implementing Interfaces
- Reference Variables and Interfaces
- Interfaces and Inheritance
- Some Uses for Interfaces

Exceptions

- Exceptions
- Attempting Risky Code - try and catch
- Guaranteeing Execution of Code - the finally Block
- Letting an Exception be Thrown to the Method Caller
- Throwing an Exception
- Exceptions and Inheritance
- Creating and Using Your Own Exception Classes
- Rethrowing Exceptions
- Initializer Blocks

Generics and Collections

- Fundamental Collections: Sets, Lists, and Maps
- Iterators
- Creating Collectible Classes
- Generics

Inner Classes

- Inner Class Syntax
- Instantiating an Inner Class Instance from Within the Enclosing Class
- Inner Classes Referenced from Outside the Enclosing Class
- Working with Inner Classes