

Price: R14,800.00 excl. VAT
Duration: 5 days
Delivery: Virtual classroom or
on-site training

Java Programming

Description

The Java Programming course covers the essential concepts and syntax of Java, from OO concepts and syntax, to exception handling and collections. This course is for you if you already have programming experience in another language, or have limited Java experience.

The Java course will give you a solid foundation. It is essential for learning other Java technologies, like Spring, JEE, and Android.

Objectives

After you have completed the Java Programming course, you will be able to:

- Understand the basic concepts of object orientation and how they apply to Java.
- Write your own Java programs.
- Read and maintain Java programs.
- Debug Java code to find and correct mistakes.

Intended Audience

You should attend the Java Programming course if:

- You are a programmer and you want to learn the Java language.
- You need to support existing Java systems.
- You need to understand Java so that you can learn other Java-based technologies.
- You have already learnt some Java - perhaps on your own or at university - but struggle with some concepts or have gaps in your knowledge.

Prerequisites

Before you attend the Java Programming course:

- You must already be a programmer and have experience in programming. If you have no programming background, start with our Introduction to Java Programming course.
- If you are a mainframe programmer, please consider first attending our Object-Oriented Analysis and Design Course. It will make your move to Java faster and easier.
- You should know how to use the command line for simple instructions.

Course Contents

Overview

- Java language history.
- Java Standard Edition (JSE) vs Enterprise Edition (JEE)
- Portability and the Java Virtual Machine.
- The Java development environment and tools.

Java Applications

- Console-based applications.
- Graphical applications.
- Applets vs applications.
- Creating a simple "Hello world" application.

OO Concepts

- Object-oriented vs procedural programming.
- Classes, attributes, behaviours.
- Data encapsulation.
- Inheritance and code reuse.
- Polymorphism.
- Abstract classes and interfaces.
- An introduction to UML.
- UML class and package diagrams.

Language Fundamentals

- Lexical structure.
- Comments.
- Keywords, identifiers and literals.
- Naming conventions.
- Data types - primitive vs object reference data types.
- Variable declarations, initialization and scope.
- Expressions, operators and operator precedence.
- Selection and iteration statements.
- Modifiers.
- Exception handling.

Classes, Interfaces and Packages

- Creating and using classes.
- Encapsulation - fields and methods.
- Constructors and initializers.
- Polymorphism - overloading and overriding.
- Inheritance - superclasses and subclasses.
- Abstract classes and interfaces.
- Canonical classes and JavaBeans.
- Inner classes.
- Packages.

Strings, Arrays and Collections

- String methods and concatenation.
- String, StringBuffer and StringBuilder classes.
- Creating, initializing and using arrays.
- Collections API overview.

Exception Handling

- Errors vs exceptions.
- Checked vs unchecked exceptions.
- The try...catch statement.
- Creating custom exceptions.
- Try with resources (TWR).

Threads and Multitasking

- Multitasking overview.
- Using and creating threads.
- The Runnable interface.
- Thread lifecycle.
- Thread synchronization.

File Input/Output

- Standard System streams.
- Files, Streams, Readers and Writers.
- An introduction to serialization.

Advanced Language Features

- Java 5 features - annotations, generics, enums, enhanced for loop, auto-boxing/unboxing, varargs, static imports.
- Java 7 features - binary literals, try-with-resources, type inference, strings in switches.
- Java 8 features - lambdas, functional interfaces, default and static methods in interfaces, method and constructor references.

*** The lecturer reserves the right to modify the contents of the course to suit the needs of the delegates.*