

**Price:** R11,900.00 excl. VAT  
**Duration:** 5 days  
**Code:** AJAVA

# Advanced Java Programming

## Description

This course will take you to the next level as a Java programmer. You will learn advanced Java programming concepts, and techniques to improve your code. The course also covers some of the new features in Java.

## Objectives

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

- Understand advanced Java programming techniques.
- Understand the JEE architecture.
- Write code using the new Java language features.
- Develop database applications using Java technologies (JDBC and JPA).
- Develop distributed applications using the appropriate Java technology.
- Develop networked applications using sockets.
- Understand more about web services.
- Understand the Java Virtual Machine garbage collection and heap management.

## Intended Audience

You should attend the Advanced Java Programming course if:

- You are a Java programmer and you want to learn about the advanced aspects of the Java language.
- You are a Java programmer and you want to learn to write better Java code.

## Prerequisites

Before you attend the Advanced Java Programming course:

- You must have attended our Java Programming course or already be comfortable with the fundamentals of the Java programming language.
- You should have at least 6 months practical experience programming in Java.

## Course Contents

### **Overview**

- Review of the Java platform.
- Overview of JEE architecture.
- Overview of garbage collection and heap management.
- Introduction to native methods - interfacing to C with JNI.

### **Language Topics**

- Java 1.4 assertions.
- Java 5 features - annotations, generics, enums, enhanced for loop, auto-boxing/unboxing, varargs, static imports.
- Java 7 features - binary literals, try-with-resources, multi-catch, type inference, strings in switches.
- Java 8 features - functional/SAM interfaces, default methods, functional programming, lambda expressions.

### **Java Collections API**

- Sets, Lists, Maps, Queues.

- Binary trees and hash tables.
- Iterator and Enumeration interfaces.
- Big O notation.
- Sorting and searching.

#### **JavaBeans**

- JavaBeans architecture and the Reflection API.
- Canonical classes.
- MVC.

#### **Persistence**

- Serialization.
- JavaBean XML encoding.
- Java API for XML Binding (JAXB).
- Flat files.
- Java Database Connectivity (JDBC).
- Java Persistence API (JPA).

#### **Java Database Connectivity**

- Client/server methodology.
- JDBC API: Driver, Connection, Statement and ResultSet interfaces.
- Driver types and loading drivers.
- Connecting to a JDBC database.
- Executing queries and extracting data.

#### **Distributed Systems**

- Overview and introduction.
- RMI architecture vs Remote Procedure Calls (RPC).
- RMI interface definitions, stubs and skeleton classes.
- CORBA and IDL.
- Web Services.

#### **Web Services**

- Comparison to RMI/CORBA/EJB.
- SOAP vs RESTful Web Services.
- XML and WSDL.
- XML vs JSON.

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