

**Price:** R6,900.00 excl. VAT  
**Duration:** 5 days  
**Code:** J2EJB

# Enterprise JavaBeans Development

## Description

This course focuses on J2EE development using Enterprise JavaBeans (EJBs). The J2EE architecture is revised. The types, uses and problems of EJBs are discussed. J2EE patterns and best practices are presented.

## Objectives

Delegates who complete this course will be able to:

- Understand the various J2EE platform technologies.
- Write entity EJBs and deploy them on an application server.
- Write and deploy session EJBs.
- Write and deploy message-driven beans to receive JMS messages.

## Intended Audience

Experienced Java programmers and developers who need to develop scalable and robust enterprise applications using EJBs and the J2EE environment. This course is aimed at companies and individuals who are already doing J2EE development and need training in the specific EJB aspects of the J2EE architecture.

## Prerequisites

The Java Programming course and at least 1 year of Java programming experience. The J2EE overview course is recommended.

## Course Contents

**J2EE Technology Revision** • Enterprise JavaBeans. • Java Transaction API (JTA) and Java Transaction Service (JTS). • RMI and RMI-IIOP. • Java Naming and Directory Interface (JNDI). • JavaMail. • Java Message Service (JMS). • Java Database Connectivity (JDBC). • J2EE Connector Architecture (JCA). • Java Authentication and Authorization Service (JAAS).

**Enterprise JavaBeans** • J2EE application servers. • Enterprise applications, EAR files and portability issues. • EJBs as the core of a J2EE application. • BMP and CMP entity beans. • Stateless and stateful session beans. • Message Driven beans. • Timers. • EJB interfaces and components. • EJB lifecycle and bean pooling. • EJB2 versus EJB3. • Deployment descriptors. • Packaging and deploying EJBs.

**Databases and Transactions** • JDBC API and drivers. • JDBC Connection Pooling. • The ACID principle. • Programmatic vs Declarative transactions with JTA and JTS. • Bean-managed persistence (BMP) vs container-managed persistence (CMP). • EJB Query Language (QL). • Bean-managed relationships (BMR) vs container-managed relationships (CMR).

**Messaging with JMS** • Point to point vs Publish and subscribe messaging. • Queuing mechanisms. • Connection Factories, Connections, Sessions, Destinations. • Message types.

**Remote Objects** • Referencing remote objects with JNDI. • Serialization for data transfer. • RMI and RMI-IIOP.

**Patterns and Best Practices** • J2EE Design Patterns. • Best Practices in J2EE development.