

**Price:** R6,900.00 excl. VAT  
**Duration:** 3 days  
**Code:** SPRNG

## Spring Development

### Description

The Spring Development course focuses on the development of applications using the Spring framework. Topics include an overview, Inversion of Control (IoC) and Dependency Injection, Aspect Oriented Programming (AOP), JEE in Spring, the Spring Web MVC, ORM, Data Access (DAO) and transaction management, Object-relational mapping (ORM) with Hibernate and iBatis.

### Objectives

Delegates who complete the Spring Development course will be able to:

- Understand the Spring framework and the use of IoC and AOP.
- Integrate into the business layer - DAO, transactions, ORM.
- Integrate into the web layer of enterprise applications using Web MVC and Web Flow.

### Intended Audience

The Spring Development course is aimed at companies and individuals who are using or planning to use Spring, and require a detailed understanding of the relevant technologies. It is intended for experienced Java programmers who need to understand the Spring architecture.

### Prerequisites

Our Java Programming course and at least 1 year of Java programming experience, with knowledge of basic XML.

### Course Contents

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

**Spring Framework Overview** • Spring Philosophy. • Easier JEE application developments using Spring. • Decoupling components with Inversion of Control. • Managing cross-cutting concerns with AOP. • The Core Container. • Comparison of Spring and EJB.

**Core Technologies - IoC and AOP** • The IoC container. • Aspect Oriented Programming and AspectJ integration. • Containers and Beans. • The BeanFactory - a factory pattern implementation for decoupling. • Dependency Injection. • Autowiring. • Bean scopes.  
• Internationalization (I18N). • Event propagation. • Resources and resource loading.  
• Application contexts and transparent creation of contexts. • Validation and data binding.

**Web Development** • Spring MVC as the Model-View-Controller implementation. • The DispatcherServlet class as a "Front Controller" design pattern. • The Controller interface for handling requests and returning models and views - ModelAndView. • Handler mappings to map requests to appropriate handlers. • Interceptors for configurable handler mappings. • Servlet listeners for IoC container initialization. • Redirecting and resolving views. • Exception handling.  
• Integrating view technologies - JSP, JSTL, Tiles, XSLT.

**DAO and Transactions** • Spring DAO as a JDBC abstraction layer. • Programmatic and declarative transaction management. • Object-relational mapping (ORM). • Integration layers to external ORM systems. • Hibernate, JDO and iBatis.

**Integration with the JEE architecture** • Remoting with RMI, Spring HTTP invoker, JAX-RPC and JMS. • JMS for receiving and sending messages. • Web service support via JAX-RPC. • Accessing EJBs. • Integrating Spring into a JMX infrastructure. • Accessing enterprise information systems (EIS) via JCA (Java Connector Architecture). • Email using MailSender.