

**Price:** R10,400.00 excl. VAT  
**Duration:** 4.5 days  
**Code:** SPRNG

## Spring Development

### Description

The Spring Framework is a Java platform that aims to make application development easier. This course will teach you how to develop Java applications in a consistent, productive way, using the Spring framework. You will learn about the various Spring technologies and how they work together. You will also learn how to use Inversion of Control (IoC), Dependency Injection (DI), and Aspect Oriented Programming (AOP).

### Objectives

After you have completed the Spring Development course, you will be able to:

- Understand the components and structure of the Spring framework.
- Use the Spring framework to develop Java applications.
- Integrate into the business layer with Spring DAO, transaction and ORM support.
- Integrate into the web layer of enterprise applications using Spring MVC and Web Flow.

### Intended Audience

You should attend the Spring Development course if:

- You are a Java programmer and you want to learn how to use Spring for development.
- You are a Java programmer and you need to support systems in an environment that uses Spring.
- You are a Java architect and you want a better understanding of the Spring framework.

### Prerequisites

Before you attend the Spring Development course:

- You must have attended our Java Programming course or already be comfortable with the fundamentals of the Java programming language.
- You should have some knowledge of basic XML.

### Course Contents

#### Spring Framework Overview

- Spring Philosophy.
- Easier Java application development using Spring.
- Decoupling components with Inversion of Control.
- Managing cross-cutting concerns with AOP.
- The Core Container.
- Comparison of the Spring and JEE containers.

#### Core Technologies - IoC and AOP

- The IoC container and Dependency Injection.
- Aspect Oriented Programming and AspectJ integration.
- Containers and Beans.
- The BeanFactory - a factory pattern implementation for decoupling.
- Dependency Injection.
- Autowiring.
- Bean scopes.

- Internationalization (I18N).
- Events 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 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 JPA.

### **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.
- Email using MailSender.

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