

Price: R5,900.00 excl. VAT
Duration: 2 days
Delivery: Virtual classroom or
on-site training

Automated Testing for Java Developers

Available on request for group bookings only

Description

This course introduces Java developers to the concepts of test-driven development and automated testing using JUnit, Mockito and Selenium.

Objectives

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

- Understand test-driven development and automated testing.
- Understand the concept of assertions.
- Write test methods using JUnit.
- Mock out objects using Mockito.
- Write web-based tests using Selenium.
- Create and organise tests and test suites.

Intended Audience

You should attend this course if:

- You are a programmer or developer and you need to understand, use and write automated tests.
- You are an analyst or system architect and you want to understand test-driven development.
- You are a project or programme manager, and you want to improve the quality of your team's code through a test-driven development methodology.

Prerequisites

You should already have some experience programming in Java, or have attended the Incus Data Java Programming course.

Course Contents

Overview of Unit Testing

- Types of testing: unit, integration, functional, system.
- Why is unit testing needed?
- Properties of good tests.
- Advantages of automating tests.
- Planning and structuring unit tests.
- Improving test quality and effectiveness.
- Continuous testing and integration.

JUnit Features

- JUnit versions - JUnit 4 vs. JUnit 5.
- JUnit framework and test skeletons.
- Test fixtures, cases, suites and runners.
- Exception testing.
- Parameterised tests.
- Rules, theories and categories.
- Other assertion libraries: Hamcrest, AssertJ, Truth.

Developing JUnit Tests

- Writing test methods using assertions and the assertXxx() methods.
- Basic annotations: @Test, @Before, @After, @BeforeClass, @AfterClass, @Ignore.
- Creating and organizing test fixtures with @Before/@After.
- Testing for exceptions.
- Composing test cases into test suites.
- Organising tests and combining test suites.
- Executing tests and showing test results.

Selenium

- Selenium WebDriver as a language-neutral interface for controlling web browsers.
- Writing Selenium scripts for automated web application testing.
- Selenium browser commands: Starting a browser session, navigating to a web page, requesting browser info, setting a waiting strategy, finding a page element, sending an element action, requesting element info, ending a session.
- Selenium 3 vs Selenium 4.

Using Mock Objects

- Why use mocking?
- Dummy vs fake objects vs stubbing vs mocking.
- Mocking to test the functionality of a class in isolation.
- Mock objects as proxies for class implementations.
- Mocking frameworks: Mockito, JMockit, EasyMock.

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