

JUnit Testing

Length:

2 Days

Description:

Developers interested in using the new testing strategy - unit testing - which interleaves coding and testing in an integrated way. This has proven to be a powerful combination that results in better designed software with fewer defects and faster delivery cycles. This course explains and shows developers how to effectively use unit testing.

Course Objectives:

- Discover the benefits of a new testing strategy; unit testing using JUnit
- Learn about unit testing frameworks
- Explore and learn different kinds of software tests
- Includes running tests from Ant and Eclipse
- Study the following testing strategies:
 - Coarse-grained testing with stubs
 - Testing in isolation with mock objects
 - In-container testing with Cactus
- Learn about testing components with:
 - Unit-testing Servlets and filters
 - Unit-testing JSPs and taglibs
 - Unit-testing database applications
 - Unit-testing EJB

Audience:

Java developers.

Prerequisites:

Working knowledge of Java and enterprise technologies such as Servlets and EJB.

Course Content:**Chapter 1 - JUnit jumpstart**

- Understanding unit testing frameworks
- Setting up JUnit
- Testing with JUnit

Chapter 2 - Exploring JUnit

- Exploring core JUnit
- Launching tests with test runners
- Composing tests with TestSuite
- Collecting parameters with TestResult
- Observing results with TestListener
- Working with TestCase

Chapter 3 - Sampling JUnit

- Introducing the controller component
- Testing exception-handling
- Setting up a project for testing

Chapter 4 - Examining software tests

- Kinds of tests
- Test-driven development

Chapter 5 - Automating JUnit

- Running tests from Ant
- Running tests from Eclipse

Chapter 6 - Coarse-grained testing with stubs

- Introducing Stubs
- Stubbing the web server's resources
- Stubbing the connection

Chapter 7 - Testing in isolation with mock objects

- Introducing mock objects
- Using mock objects as a refactoring technique
- Practicing on a HTTP connection sample
- Using mocks as Trojan horses
- Deciding when to use mock objects

Chapter 8 - In-container testing with Cactus

- Testing components using mock objects
- Integration unit tests
- Introducing Cactus
- Testing components using Cactus
- How Cactus works

Chapter 9 - Unit-testing servlets and filters

- Writing servlet tests with Cactus
- Testing servlets with mock objects
- Writing filter tests with Cactus
- When to use Cactus, and when to use mock objects

Chapter 10 - Unit-testing JSPs and taglibs

- JSP unit testing
- Unit-testing a JSP in isolation with Cactus
- Unit-testing taglibs with Cactus
- Unit-testing taglibs with mock objects
- When to use mock objects, and when to use Cactus

Chapter 11 - Unit-testing database applications

- Database unit testing
- Testing business logic in isolation from the database
- Testing persistence code in isolation from the database
- Writing database integration unit tests
- Running the Cactus test using Ant
- Tuning for build performance

- Overall database unit-testing strategy

Chapter 12 - Unit-testing EJBs

- Defining a sample EJB application
- Using a facade strategy
- Unit testing JNDI code using mock objects
- Unit-testing session beans
- Using mock objects to test message-driven beans
- Using mock objects to test entity beans
- Choosing the right mock object strategy
- Using integration unit tests
- Using JUnit and remote calls
- Using Cactus