

**Course 443:**  
**Mastering Test-Driven Development using JUnit :**  
**Hands-On**  
**(3 days)**

**Course Description...**

This in-depth workshop will help developers, product managers, and quality engineers hit the ground running with test-driven development skills. This is a comprehensive course on test-driven development that is 50% hands-on and 50% lecture, with multiple mini-projects built in to reinforce skills and concepts learned in the lectures. It covers unit testing, functionality testing and integration testing. It teaches developers how to write unit tests when using Java EE frameworks such as Struts and Spring.

**Learning Objectives...**

- Test business objects and services
- Follow best practices in test-first development
- Run unit tests during day-to-day development
- Best practices in maintaining unit tests
- Measuring test coverage
- Identifying design patterns and refactorings
- Testing container-managed objects using Spring Mock objects
- Testing objects that require dependency injection
- Testing data-driven classes using DBUnit
- Testing performance using JUnitPerf
- Testing GUIs and threaded applications

**Who should attend...**

This is an intermediate level Java programming course, designed for developers, product managers and quality engineers who wish to learn best practices for writing correct and testable programs in Java.

**Prerequisites...**

Attendees should be experienced Java programmers. Development experience using Java Enterprise technologies such as servlets, JSPs, EJB or Spring is helpful.

**See next page for a detailed course outline...**



## Course Outline...

### Chapter 1: Test-driven Development

- Why test-driven development?
- What to test
- Forms of software testing
- Tools for testing
- Continuous integration
- Agile Programming
- **Hands-On Exercise**

### Chapter 2: Unit Testing with JUnit

- Unit testing with JUnit
- Assertions
- Set up and tear down
- Testing exception throwing
- Simplifying tests with JUnit4 annotations
- TestNG & Java 5
- JUnit 4 Gotchas
- **Hands-On Exercises**

### Chapter 3: The Testing Lifecycle

- Unit testing patterns
- Measuring test coverage
- Maintaining unit tests
- Fixing bugs
- Refactoring
- Design patterns
- **Hands-On Exercises**

### Chapter 4: Functionality Testing

- Testing use cases
- Testing container-managed components
- Mock objects
- Spring support for mock objects
- Autowiring of mock objects
- **Hands-On Exercises**

### Chapter 5: Testing Data Access Objects

- Strategies for testing data-driven applications
- Transactional test cases
- DBUnit
- **Hands-On Exercises**

### Chapter 6: Integration Testing

- Measuring performance
- Timed and Load Tests
- Testing GUIs
- Concurrent tests
- **Hands-On Exercises**

*Please contact your ROI representative to discuss course customization!!!*