

Unit Testing for C#

Duration: 3 Days

Day 1

- **Getting Started**
 - What is Automated Testing
 - Benefits of Automated Testing
 - Types of Tests
 - Test Pyramid
 - The Tooling
 - Source Code
 - Writing Your First Unit Test
 - Testing All the Execution Tests
 - Refactoring with Confidence
 - Using NUnit in Visual Studio
 - What is Test-Driven Development

- **Fundamentals of Unit Testing**
 - Introduction
 - Characteristics of Good Unit Tests
 - What to Test and What Not to Test
 - Naming and Organizing Tests
 - Introducing Rider
 - Writing a Simple Unit Test
 - Black-box Testing
 - Set Up and Tear Down
 - Parameterized Tests
 - Ignoring Tests
 - Writing Trustworthy Tests
 - Developers Who Don't Write Tests

Day 2

- **Core Unit Testing Techniques**
 - Introduction
 - Testing Strings
 - Testing Arrays and Collections
 - Testing the Return Type of Methods
 - Testing Void Methods
 - Testing Methods that Throw Exceptions

- Testing Methods that Raise an Event
- Testing Private Methods
- **Code Coverage**
- **Testing in the Real-world**
 - Exercises
- **Breaking External Dependencies**
 - Introduction
 - Loosely-coupled and Testable Code
 - Refactoring Towards a Loosely-coupled Design
 - Dependency Injection via Method Parameters
 - Dependency Injection via Properties
 - Dependency Injection via Constructor
 - Dependency Injection Frameworks

Day 3

- **Mocking Frameworks**
 - Creating Mock Objects Using Moq
 - State-based vs. Interaction Testing
 - Testing the Interaction Between Two Objects
 - Fake as Little As Possible
 - An Example of a Mock Abuse
 - Who Should Write Tests
 - Exercises

Project - Creating and Testing CustomerOrderService

Project - Creating and Testing GestBook Web Application

1. Test Framework Selection
2. Writing sample Test Cases
3. Testing Best Practices
4. Advanced Testing Scenarios
5. Testing Documentation