An emerging software engineering paradigm is test-driven development, where tests are written early and testing is performed continuously during the development process. Problems are discovered early and corrected when they are found. This one-day course for developers explains the methodology of test-driven development and the use of the unit testing framework that comes with Visual Studio 2013, including the free Express version. The course is practical, with many example programs and tests written in C#, including a cumulative case study. The goal is to quickly bring you up to speed in doing unit testing in your .NET development projects. The student will receive a comprehensive set of materials, including course notes and all the programming examples.

**Audience:** Developers needing to become familiar with unit testing framework and its standards.

**Prerequisites:** The student should have a basic knowledge of the .NET Framework and experience programming in C# with Visual Studio.

**Number of Days:** 1 day

---

1. **Test-Driven Development**
   - Test-Driven Development
   - Functional Tests
   - Unit Tests
   - Test Automation
   - Rules for TDD
   - Implications of TDD
   - Simple Design
   - Refactoring
   - Regression Testing
   - Test List
   - Red/Green/Refactor
   - Using the Unit Testing Framework
   - Testing with Unit Testing Framework
   - Unit Testing Framework Test Drive
   - IQueue Interface and Stub Class
   - Test List for Queue
   - A Second Test
   - More Queue Functionality
   - TDD with Legacy Code
   - Acme Travel Agency Case Study

2. **Visual Studio Unit Testing Fundamentals**
   - Structure of Unit Tests
   - Assertions
   - Unit Testing Framework
   - Unit Testing Framework Namespace
   - Assert Class
   - Assert.AreEqual()
   - More Assert Methods
   - CollectionAssert Class
   - StringAssert Class
   - Test Case
   - Test Methods
   - Test Class
   - Test Runner
   - Command Line Test Runner
   - Ignoring Tests
   - Using the Ignore Attribute
   - Test Initialization and Cleanup
   - Class Initialization and Cleanup

3. **More about Unit Testing Framework**
   - Expected Exceptions
   - Enqueue and Dequeue
   - Tests for Enqueue and Dequeue
   - ToArray()
   - Test of ToArray()
   - Exceptions Dialog
   - Custom Asserts
   - Implementing a Custom Assert
   - Playlists
   - Debugging Unit Tests
   - Running Selected Tests
   - Running Tests at the Command Line
   - MSTest.exe Command Line Options
   - Refactoring
Collection Class Implementation
Testing the New Version

4. Appendix A: Learning Resources