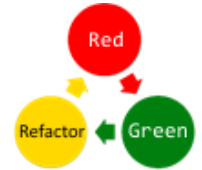


Unit Testing in Visual Studio 2026

UTVS2026 | 2 Days



Design, write, and run high-quality .NET unit tests using Visual Studio. This two-day course builds your fluency across unit testing frameworks, Test-Driven Development, code coverage, concurrent testing, and the techniques for testing difficult code with doubles and mocks. You'll work hands-on the way a real developer would. Note: This course targets Visual Studio 2026 and .NET 10.

Who should take this course?

This course is for software development professionals building high-quality .NET applications who want to design, write, and run effective unit tests. It's equally valuable for developers adopting TDD, refactoring, and techniques for testing difficult code. Experience with C#, Visual Studio, and writing and maintaining code is helpful but not required.

Course content

This course progressively builds your fluency with unit testing in Visual Studio, from .NET frameworks and Test-Driven Development to writing good tests, leveraging Visual Studio's testing features, and testing difficult code. You'll work hands-on throughout, the way a real developer would.

1. UNIT TESTING IN .NET

- The role of the developer
- Unit tests explained
- .NET unit testing frameworks (MSTest, NUnit, and xUnit)
- The anatomy of a unit test
- Writing and running your first unit test

2. UNIT TESTING IN VISUAL STUDIO

- Testing support and test projects
- Test Explorer and testing windows
- Writing and running unit tests
- Managing large numbers of tests and results
- Grouping, filtering, and playlists
- Continuous testing in Visual Studio

3. TEST-DRIVEN DEVELOPMENT (TDD)

- TDD overview and benefits
- Practicing TDD in Visual Studio
- Common objections to TDD
- Effective refactoring
- Working with legacy code
- Using CodeLens to support TDD and refactoring

4. WRITING GOOD UNIT TESTS

- Knowing your code and asking the right questions
- Path testing (happy, sad, evil)
- Right BICEP testing
- Testing for expected exceptions
- Naming conventions and organizing tests
- Maintaining high-quality test code

5. LEVERAGING VISUAL STUDIO

- Analyzing code coverage
- Using code coverage as a metric
- Data-driven (parameterized) unit tests
- Concurrent testing and Live Unit Testing
- AI-assisted unit testing with GitHub Copilot

6. TESTING DIFFICULT CODE

- Isolating code under test
- Doubles: dummies, stubs, fakes, and mocks
- Microsoft Fakes (stubs and shims)
- Comparing mocking frameworks
- Debugging and profiling slow-running tests
- AI-assisted testing of difficult code

Course Designer

This course was designed by Richard Hundhausen, Microsoft's first Visual Studio ALM/DevOps MVP, a Professional Scrum Trainer, and co-creator of the Nexus framework, with decades helping teams build and ship software. To see other Visual Studio courses, visit www.accentient.com.