Benjamin Day Consulting, Inc.
17 Roberts St #2
Brookline, MA 02445
617-645-0188
http://benday.com

Unit Testing & Test-Driven Development with Visual Studio 2017

Unit testing enables you and your team to write high-quality software with fewer bugs. It also allows you say with confidence when something in your code is working and – more importantly – know when it is not working. Test-Driven Development (TDD) is a methodology for writing software that ensures that your application and your application code are testable and tested from the very start.

This course is designed to give the student hands-on experience and knowledge for writing and maintaining applications using Test-Driven Development. Throughout the course we will discuss the options, process, and motivations for unit testing and TDD and reemphasize these concepts using hands-on labs.

What you'll learn:

- What is a unit test?
- What is Test-Driven Development (TDD)? Why is TDD important?
- What is the TDD process?
- Common (bogus) excuses for NOT doing Unit Testing & TDD
- How do I sell my team on Unit Testing & TDD?
- Create and write unit tests
- Test types in Visual Studio 2017
- Architect your application for testability
- Testing user interfaces
- Test non-public methods
- Strategies for unit testing legacy code
- Fakes, Mocks, & Stubs
    - Using the Visual Studio 2017 Fakes Framework
    - Using Mocks & Stubs to avoid The Huge Integration Test pitfall
    - Dealing with databases and test data in your unit tests
    - Mocking web services, back-end systems, and database calls
    - Use Mocks & Stubs to reach those hard to test cases
- Design Patterns for testability: Repository, Adapter, N-Tier Architecture, Model-View-Presenter (MVP), Model-View-Controller (MVC), and Model-View-ViewModel (MVVM)
- Best Practices for fixing bugs & defects using TDD
- Refactor for testability
- What is Code Coverage and why do I care?

- What other non-TDD test types are available?
    - Web Performance Tests
    - Load Tests
    - Coded UI Tests
- Testing ASP.NET MVC
- Testing Entity Framework
- Where does testing fit with DevOps?

Details

Technologies: Visual Studio 2017
Programming language: C#
Duration: 2 days, 9am to 5pm

Lab Machine Requirements

Operating System: Windows 10, Windows Server 2016
Memory: 8GB or more
Required applications: Visual Studio 2017 with latest service packs applied
Other: Lab machines should be up to date with patches from Windows Update. Students should have administrator rights on their laptop.