

This two-day, instructor-led course provides students with the knowledge and skills to effectively use Visual Studio 2008 to perform test driven software development. The course focuses on applicable features and capabilities of Visual Studio and Visual Studio Team System as related to Unit Testing and Test Driven Development (TDD). In addition, this course introduces other, industry-standard, open-source tools and how they integrate with Visual Studio.



### AT COURSE COMPLETION

This course teaches students how to leverage the unit testing capabilities of Visual Studio 2008 in order to properly implement Test Driven Development. Specifically, attendees will learn how to:

- Create and execute unit tests in Visual Studio
- Implement Test Driven Development (TDD)
- Apply TDD as a design tool
- Refactor existing code to bring it under test
- Test boundary conditions
- Appropriately identify and apply advanced TDD technologies such as mock objects
- Understand and apply code coverage tools as part of an overall test strategy
- Use data-driven unit tests

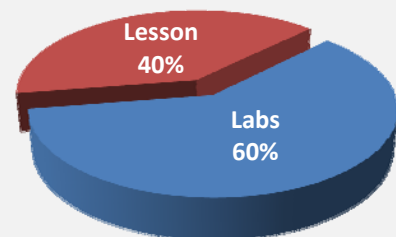
### AUDIENCE

This course is intended for current software development professionals, including architects, developers, and testers who are involved in building Windows or Web-based .NET applications. Regardless of the student's role, he or she will learn and get hands-on experience performing TDD, from varying points of view, using Visual Studio Team System.

### PREREQUISITES

Before attending this course, students should have experience creating, maintaining, and debugging software using Visual Studio and Visual C#, or another .NET object-oriented language. Knowledge of software development lifecycle is beneficial.

### LECTURE / HANDS-ON LAB BALANCE



## MODULE 1: UNIT TESTING IN .NET

This module introduces the fundamental concepts of unit testing and how they are supported by various unit testing frameworks for .NET, including XUnit, MSUnit, and Visual Studio 2008.

### Lessons

- Unit tests explained
- Unit testing frameworks for .NET
- XUnit
- MSUnit
- Unit Testing in Visual Studio

### Labs/Activities

- Creating and running simple unit tests in Visual Studio

## MODULE 2: TEST DRIVEN DEVELOPMENT

This module introduces Test Driven Development (TDD) and explores the objections and benefits. Refactoring and approaches to legacy code are also covered.

### Lessons

- TDD explained
- TDD objections vs. benefits
- Refactoring
- Legacy Code
- Groking TDD

### Labs/Activities

- The Bowling Kata

## MODULE 3: TDD PATTERNS

This module continues the discussion of TDD by introducing patterns and preferred practices around planning and writing your tests. Boundary and performance considerations are also explored.

### Lessons

- Know my code
- Testing the Sad Path
- Organizing Test Libraries

### Labs/Activities

- Testing the SAD path

## MODULE 4: DOUBLES AND MOCKING

This module introduces the various types of doubles and how they can be used to properly test your system. Rhino Mocks, a popular, free (BSD licensed) framework is introduced and used in the hands-on lab.

### Lessons

- Doubles
- Dummies, fakes, and stubs
- Mock objects
- *Rhino Mocks*

### Labs/Activities

- Advanced testing using Rhino Mocks

## MODULE 5: MORE UNIT TESTING FEATURES

This module examines three separate, but important capabilities of unit testing within Visual Studio Team System: data-driven unit tests, code coverage, and how to unit test ASP.NET web applications.

### Lessons

- Data-driven unit testing
- Code coverage
- Testing ASP.NET
- Model View Presenter (MVP)

### Labs/Activities

- Creating a data-driven unit test
- Computing code coverage
- Increasing code coverage

## COURSE DESIGNER

This course was designed by David Starr of Elegant Code and licensed to Accentient. David is a software architect specializing in Microsoft technologies and has been teaching and presenting on technology subjects for over 15 years.