

# Introduction To Test Driven Development Training Class

**Salt Valley  
Software  
Services**



Gary Brown  
Salt Valley Software Services  
Lincoln, NE  
[glbrown@inebraska.com](mailto:glbrown@inebraska.com)

# Basic Test-Driven Development Training

## Background Information

Test-Driven Development (TDD) is an advanced object-oriented programming technique intended to help you write software that has better structure, in less time, with no defects. TDD achieves this goal by strongly encouraging the separation of business logic from presentation logic, minimizing the time spent doing repetitive manual debugging, and verifying that the code exactly meets the requirements.

TDD programmers use the xUnit<sup>1</sup> family of automated unit-testing frameworks to create test fixtures for each business object as they are writing their application code. The xUnit frameworks were developed by and for the Extreme Programming (XP) community. TDD is one of the core practices of XP, but experience has shown that TDD can help programmers of all stripes, not just the extremos.

## Class language options

The Test-Driven Development class can be presented using VB6-SP5, C#, or VB.Net (we will add Java, if there is sufficient interest). Each class will teach TDD within the context of only one of those programming languages. Please make sure to sign up for the correct class.

## What you will learn:

- Mechanics of Test-Driven Development
- How to install and use one the xUnit Frameworks
- How to write clean, simple code
- How to verify that requirements are met
- How to test everything that could possibly break
- How to refactor code to eliminate duplication and clearly reveal intent
- Experience the “Rhythm of Success”

## Course Pre-requisites

Test-Driven Development is an advanced object-oriented programming course. As such, it is important that each student have a solid understanding of the code constructs, libraries, and idioms of the target language.

---

<sup>1</sup> JUnit for Java, NUnit for C# and VB.Net, NUnitASP for ASP.Net, etc. See [www.junit.org](http://www.junit.org), [www.nunit.org](http://www.nunit.org), etc. for more information.

## **About the hands-on exercises**

There are three hands-on exercises used in this course, Hello World, USGA Handicap Differential, and The Bowling Game. Each of these exercises was carefully chosen to demonstrate the use of Test-Driven Development within the context of the class. The problem domains are widely understood, and contain just the right amount of complexity, given the time constraints of the class.

The ubiquitous Hello World exercise is used to demonstrate the installation of the xUnit framework, how to setup a project for TDD, and how to write a simple test fixture.

The USGA Handicap Differential exercise applies the skills gained from Hello World. We will add to our knowledge, by writing more complex tests, begin to learn about refactoring, and how to test everything that could possibly break.

The Bowling Game further extends our learning of TDD by writing many more complex test cases, doing extensive refactoring, and integrating our class library with an existing GUI component.

## **Basic Test-Driven Development Course Outline:**

### **Day One**

#### **TDD Overview**

- Introduction to TDD Presentation

#### **TDD Demo**

- USGA Handicap Differential

#### **Hello World Exercise**

- Installing xUnit
- TDD Mechanics Presentation
- Hello World Exercise

#### **Do The Simplest Thing That Could Possibly Work**

- DTSTTCPW Presentation
- USGA Handicap Differential Exercise, part 1 – Setting up a project for TDD
- USGA Handicap Differential Exercise, part 2 – Implementing the positive path

## **Test Everything That Could Possibly Break**

- TETCPB Presentation
- USGA Handicap Differential Exercise, part 3 – Begin to TETCPB

## **Introduction to Refactoring**

- Basic Refactoring Presentation
- USGA Handicap Differential Exercise, part 4 – Refactoring the code
- USGA Handicap Differential Exercise, part 5 – Finish up TETCPB
- USGA Handicap Differential Exercise, part 6 – Handling changing requirements

## **Day Two**

### **TDD Review**

- TDD Mechanics
- DTSTTCPW
- TETCPB
- Refactoring

### **The Bowling Game, part 1**

- Overview
- Project Setup
- Iteration 1 – Scoring open frames
- Iteration 2 – Scoring strikes
- Iteration 3 – Scoring spares
- Refactoring

## **Day Three**

### **TDD Review**

- TDD Mechanics
- DTSTTCPW
- TETCPB
- Refactoring
- The Bowling Game

### **The Bowling Game, part 2**

- Iteration 4 – Scoring complete games
- Iteration 5 – Incremental scoring

- Iteration 6 – Adding Frames
- Refactoring
- Iteration 7 – GUI Integration

## **Course Review**

- TDD Mechanics
- DTSTTCPW
- TETCPB
- Refactoring
- The Bowling Game
- Complete Course Rating Surveys