

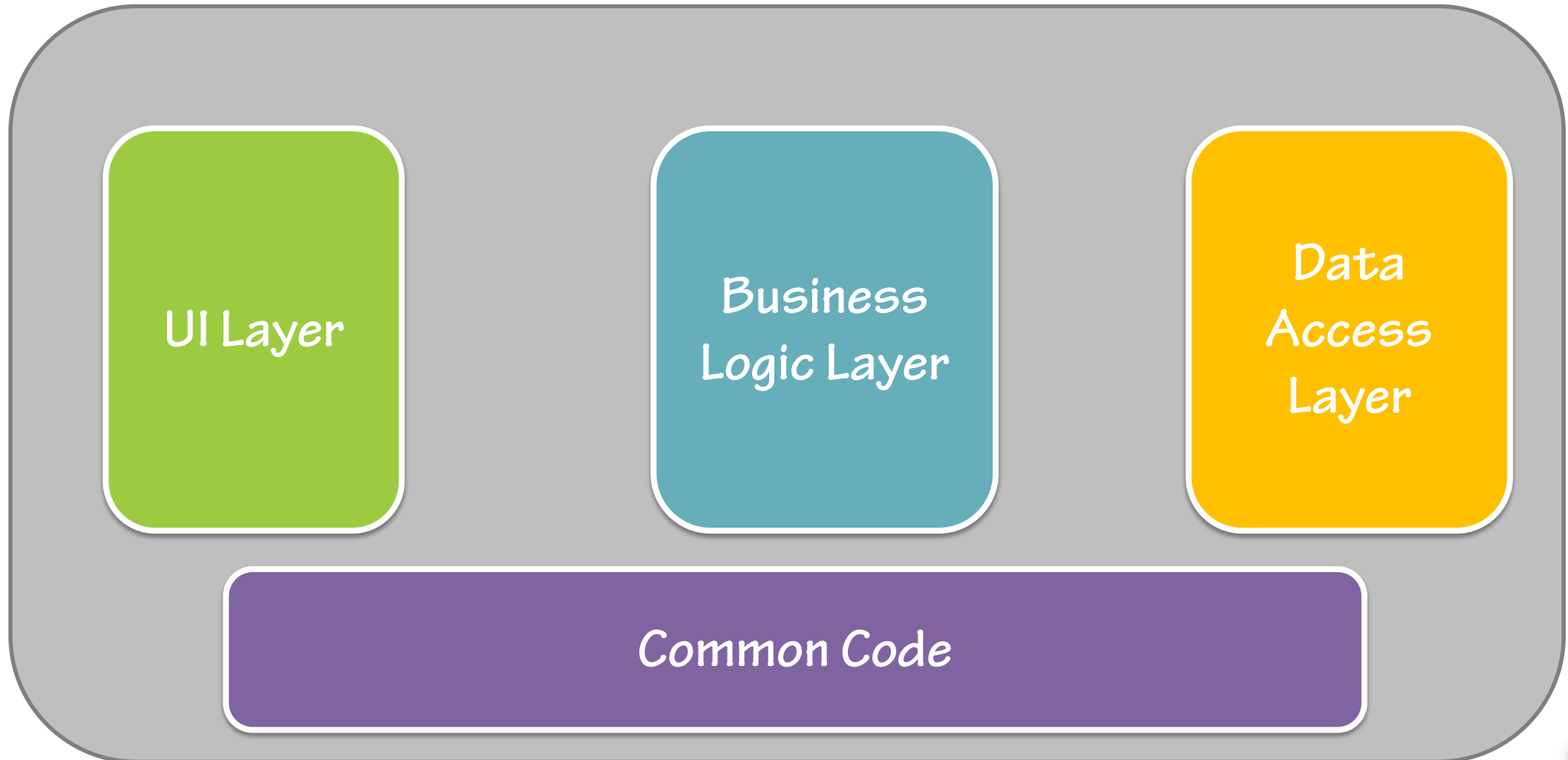
Defining Proper Classes



Deborah Kurata

@deborahkurata | blogs.msmvps.com/deborahk/

Application Architecture



Module Overview



Application Architecture

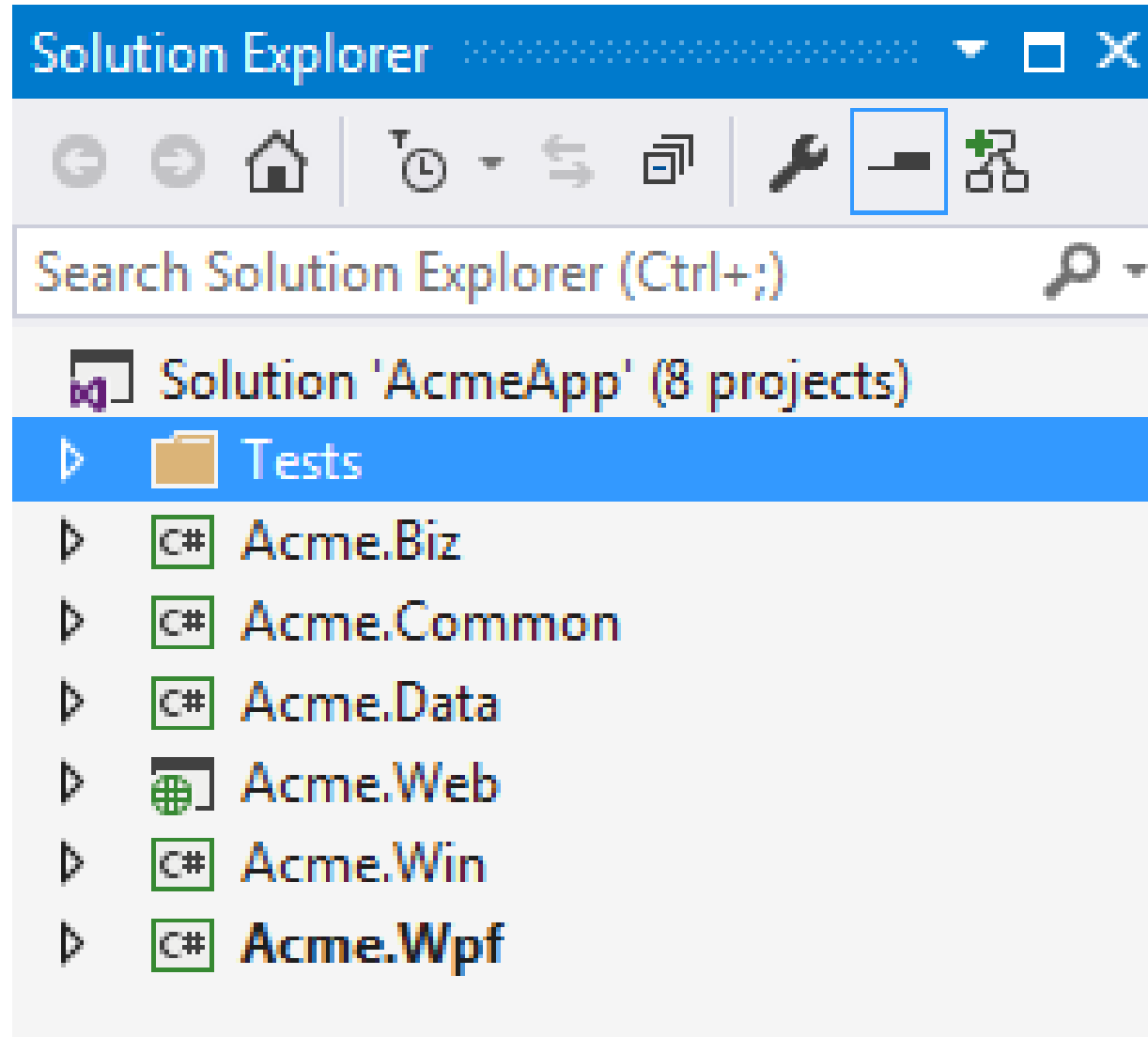
What is a Class?

Types of Classes

Unit Testing

FAQ

Application Architecture



Visual Studio Solution (AcmeApp)

UI Layer

Acme.Win

Acme.Wpf

Acme.Web

Business Logic
Layer

Acme.Biz

Data Access
Layer

Acme.Data

database

Acme.Common

Recommended Viewing

"Getting Started with Entity Framework 6"

Visual Studio Solution (AcmeApp)

UI Layer

Acme.Win

Acme.Wpf

Acme.Web

Business Logic
Layer

Acme.Biz

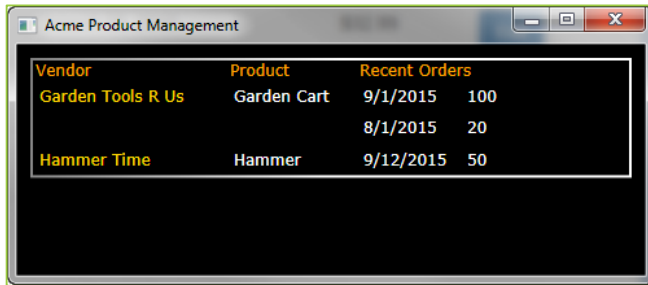
Data Access
Layer

Acme.Data

database

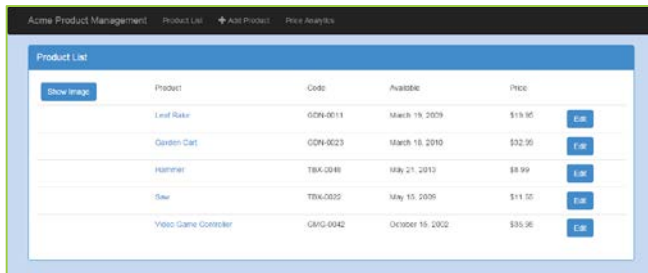
Acme.Common

C# Application "Things"



Acme Product Management

Vendor	Product	Recent Orders	
Garden Tools R Us	Garden Cart	9/1/2015	100
		8/1/2015	20
Hammer Time	Hammer	9/12/2015	50



Acme Product Management

Product	Code	Available	Price	
Loud Rake	GDN-0011	March 18, 2009	\$19.95	Edit
Garden Cart	GDN-0023	March 18, 2010	\$52.50	Edit
Hammer	TBX-0049	May 21, 2010	\$8.99	Edit
Saw	TBX-0032	May 15, 2006	\$11.50	Edit
Video Game Controller	CMG-0042	October 16, 2002	\$35.96	Edit

Products

Logging

DataSet

Vendors

Email
Generation

DbContext

Orders

What Is a Class?



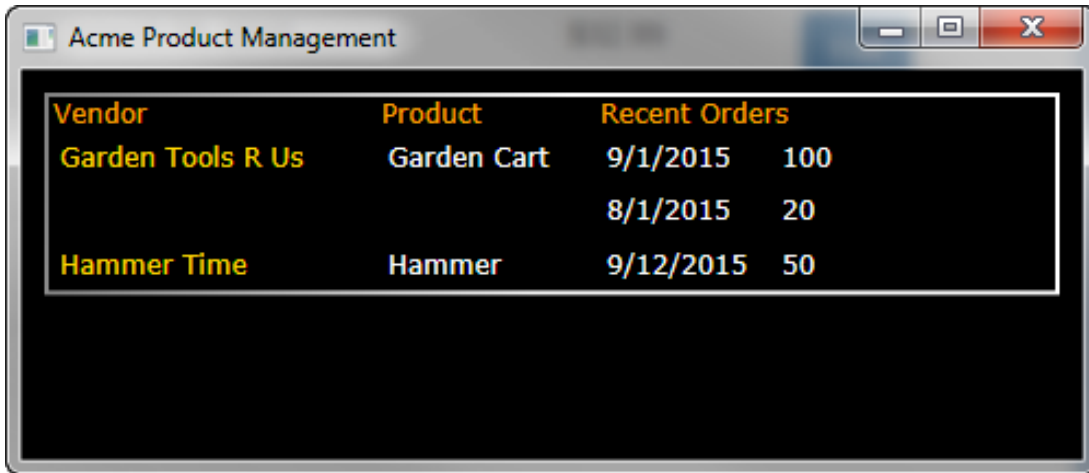
A template or mold

Specifies the traits or data

Specifies the behavior or operations

Reference: Wikipedia (German)
<https://de.wikipedia.org/wiki/Spritzgie%C3%9Fen>

What Is a Class?



A screenshot of a software window titled "Acme Product Management". Inside the window is a table with three columns: "Vendor", "Product", and "Recent Orders". The table contains two rows of data. The first row shows "Garden Tools R Us" as the vendor, "Garden Cart" as the product, and two orders: one on 9/1/2015 for 100 units and another on 8/1/2015 for 20 units. The second row shows "Hammer Time" as the vendor, "Hammer" as the product, and one order on 9/12/2015 for 50 units.

Vendor	Product	Recent Orders	
Garden Tools R Us	Garden Cart	9/1/2015	100
		8/1/2015	20
Hammer Time	Hammer	9/12/2015	50



Products

A template or mold

Specifies the traits or data

Specifies the behavior or operations

What Is a Class?

```
namespace Acme.Wpf.Views
{
    /// <summary>
    /// Interaction logic for VendorDetailView.xaml
    /// </summary>
    2 references
    public partial class VendorDetailView : Page
    {
        0 references
        public VendorDetailView()
        {
            InitializeComponent();
        }
    }
}
```

A template or mold

Specifies the traits or data

Specifies the behavior or operations

What Is a Class?

```
namespace Acme.Biz
{
    /// <summary>
    /// Manages the vendors from whom we purchase our inventory.
    /// </summary>
    9 references
    public class Vendor
    {
        1 reference
        public int VendorId { get; set; }
        7 references
        public string CompanyName { get; set; }
        4 references
        public string Email { get; set; }

        /// <summary>
        /// Sends an email to welcome a new vendor.
        /// </summary>
        /// <returns></returns>
        3 references
        public string SendWelcomeEmail(string message) ...
    }
}
```

A template or mold

Specifies the traits or data

Specifies the behavior or operations

Types of Classes

User
interface
classes

Domain
entity
classes

Library
classes

Visual Studio Solution (AcmeApp)

UI Layer

Acme.Win

Acme.Wpf

Acme.Web

Business Logic
Layer

Acme.Biz

Data Access
Layer

Acme.Data

database

Acme.Common

Visual Studio Solution (AcmeApp)

UI Layer

Acme.Win

Form

Acme.Wpf

Form

V/M

Acme.Web

Form

Business Logic
Layer

Acme.Biz

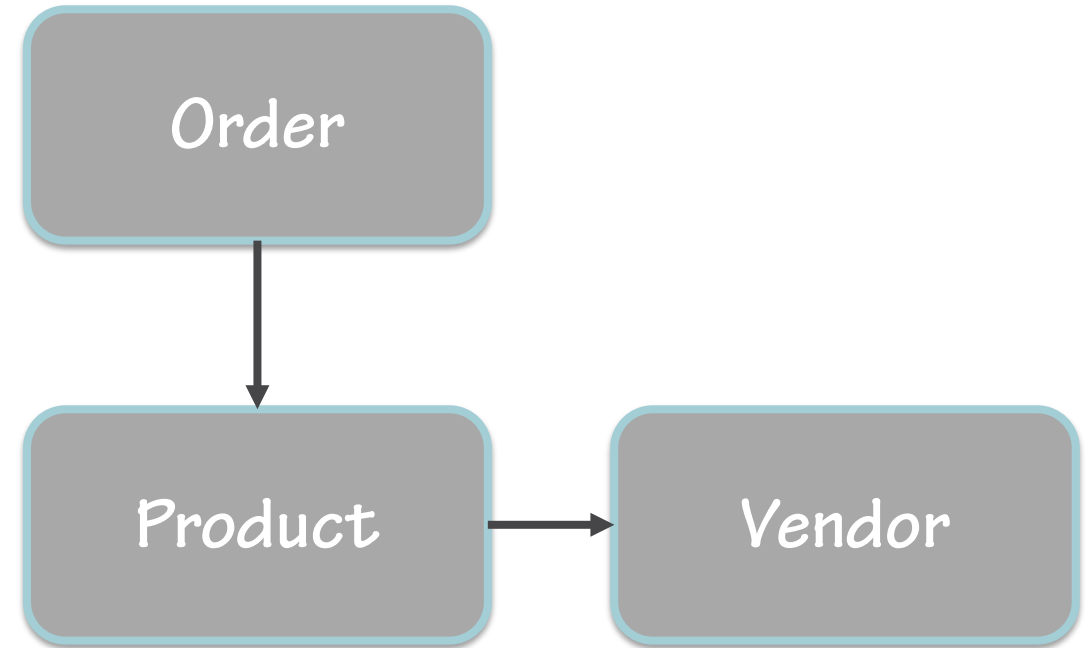
Data Access
Layer

Acme.Data

database

Acme.Common

Business Logic Component(s)



Business Domain Model

Visual Studio Solution (AcmeApp)

UI Layer

Acme.Win

Form

Acme.Wpf

Form

V/M

Acme.Web

Form

Business Logic
Layer

Acme.Biz

Order

Order
Repository

Product

Product
Repository

Vendor

Vendor
Repository

Data Access
Layer

Acme.Data

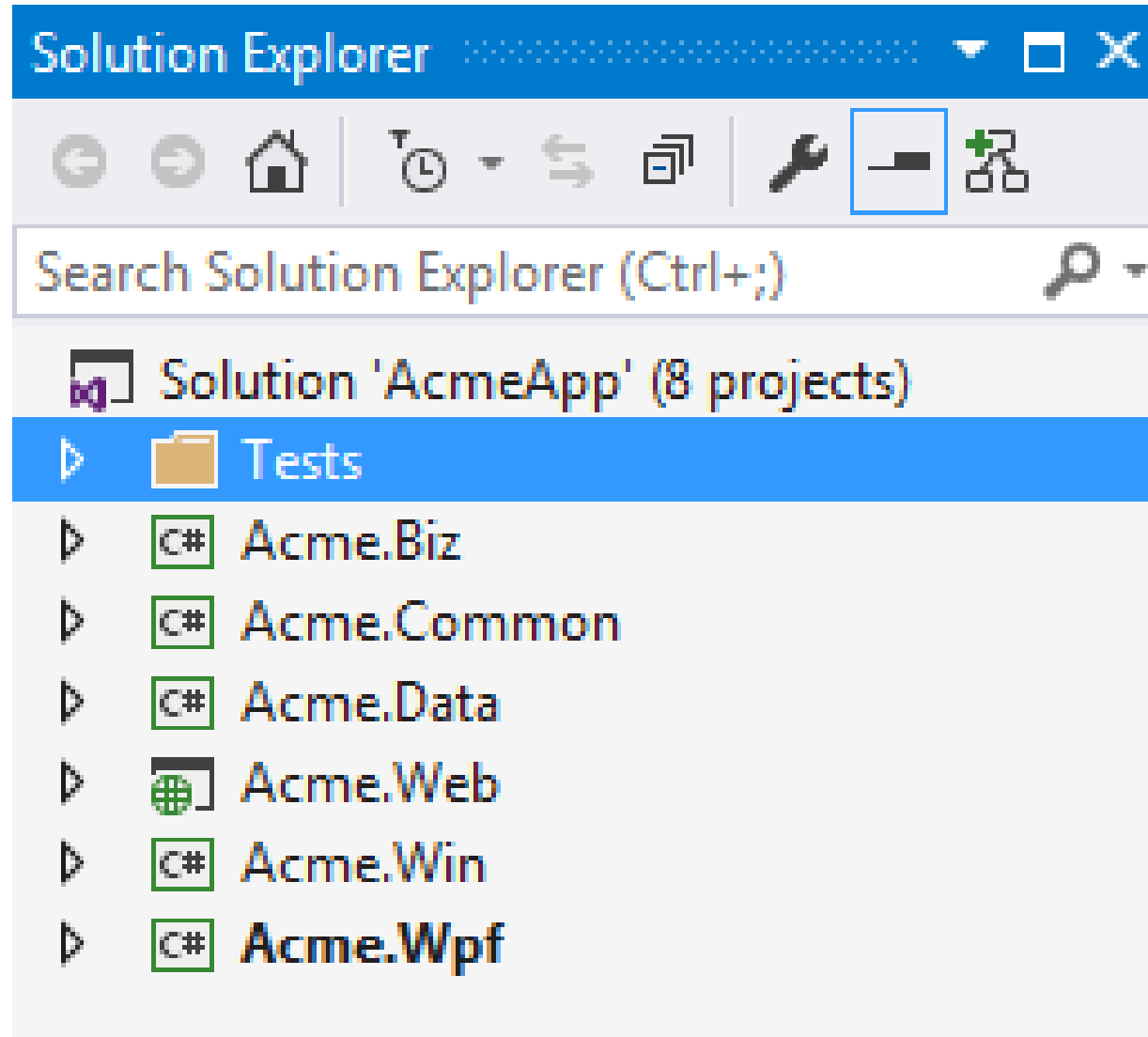
database

Logging

Acme.Common

Email

Application Architecture



Unit Testing

- Tests the behavior of a unit of code
 - Often a method
- Automated
- Defined with code
- Identifies errors

```
public int CalculateDiscount(int percent) {  
    return this.total - (this.total * percent/100)  
}
```

Tests call CalculateDiscount(x) where x is

- 10
- 100
- 200
- 0
- null

Higher Code Quality
Testable Methods
Test Cases Clarify Scenarios
Tests Identify Failures

Higher Code
Quality

Faster and Easier
Debugging

Higher Code
Quality

Faster and Easier
Debugging

Repeatable

Unit Testing in Visual Studio

- MSTest
- Steps:
 - Define the test scenarios
 - Generate the tests
 - Execute the tests

Defensive Coding in C#

This course will show you how to write clean, maintainable, and testable code, and how to keep that code great using defensive coding techniques.

Frequently Asked Questions

- Why is a **layered architecture** important?
 - Logical components are easier to create, change, extend, and maintain
 - Code is easier to reuse
- What is a **class**?
 - A template for the objects created at runtime
 - Specifies the data and operations for each entity
- What are the benefits of **unit testing**?
 - Higher quality code, faster and easier debugging, and they are repeatable over the life of the application

This Module Covered



Application Architecture

What is a Class?

Types of Classes

Unit Testing