

What Is XUnit?

XUnit

- Open-source framework for DotNet unit testing
- Can be run from the command-line
- Tests can run in Parallel
- Supports shared test **fixtures**
- Is cross-platform (support for DotNet Core)
- Is installable via NUGET

Basic Test

XUnit

```
public class TestClass1 //each class is a test collection
{
    [Fact] //each test uses the Fact attribute
    public void Test1( )
    {
        Thread.Sleep(500);
    }
}
```

Test Anatomy

Tests consist of

- Test Fixture Setup Step
 - Setting up the scenario
- Exercise the System Under Test (SUT)
 - Execute the function you are testing
- Assertions step
 - Verify the outcome
- Test Fixture teardown
 - Clean

Test Anatomy

```
public class TestClass1 //each class is a test collection
{
    public TestClass1() //Constructor sets up the fixture
    {
    }

    [Fact] //each test uses the Fact attribute
    public void Test1()
    {
    }

    ~TestClass() { //Finalizer teardown the fixture
    }
}
```

Shared Context

Tests can share the setup/teardown code

```
public class ModelFixture : IDisposable
{
    public ModelFixture()
    {
        new ClearFactory().Then(new ModelFactory()).Run();
    }

    public void Dispose()
    {
        FactoryGirl.Clear();
    }
}
```

Shared Context

Tests can share the setup/teardown code

```
public class TestClass1 : ITextFixture<ModelFixture>
{
    [Fact]                //each test uses the Fact attribute
    public void Test1( )
    {
    }
}
```

Test Collections

Tests can be grouped into Collections

- Classes within the collection are **not** run in parallel

[Collection("My Test Collection")]

```
public class TestClass1 {  
    [Fact]                //each test uses the Fact attribute  
    public void Test1()  
    {  
    }  
}
```

Fixture Setup

Setup any data for a given test scenario

- Class constructor acts as setup

```
public class TestClass1 {  
    private IDbConnection connection;  
  
    public TestClass1() {  
        connection = new SqlConnection("server=local");  
        SetUpTestData(connection);  
    }  
    ...  
}
```


Exercise the System Under Test (SUT)

Setup any data for a given test scenario

```
public class TestClass1 {  
    .... setup...  
    [Fact]  
    public void TestAddCustomer( ) {  
        var santa =new Customer("Ho Ho Ho");  
        customerRepository.Save(santa);  
        ...  
    }  
}
```

Verification

Setup any data for a given test scenario

```
public class TestClass1 {  
    ... setup ...  
    [Fact]  
    public void TestAddCustomer( ) {  
        ... add customer ...  
        var foundSanta = customerRepository.Find("Santa");  
  
        foundSanda.Should( ).NotNull( );  
        ...  
    }  
}
```

Verification

Fluent Assertions

- Open source assertion framework
- Fluent interface makes assertions easy to read

```
1.Should( ).Be(1);
```

```
true.Should( ).BeTrue( );
```

```
result.Should( ).Be(expectedResult);
```

<https://fluentassertions.com/>

XUnit Workshop

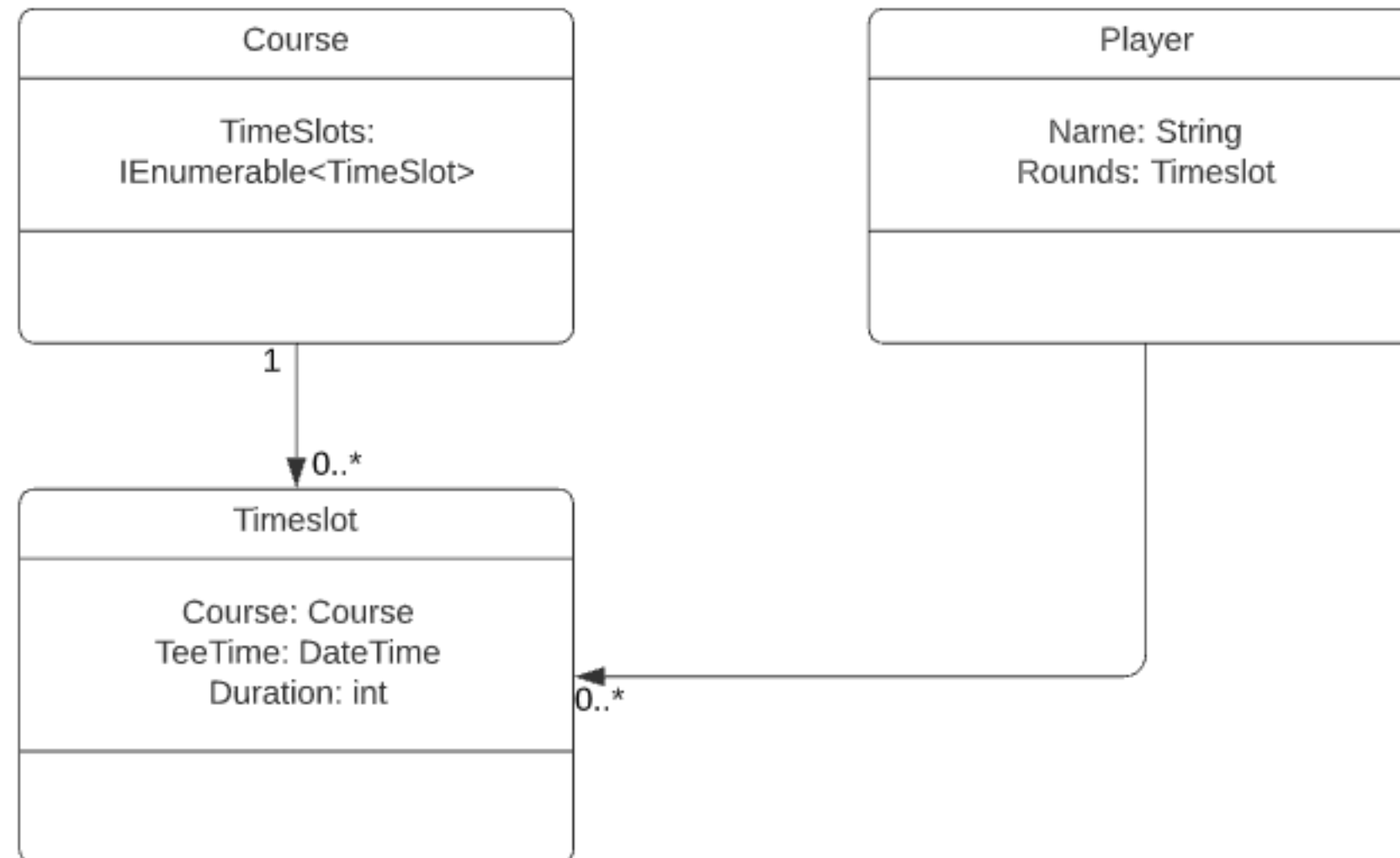
XUnit.101 DotNet Project

- Load the XUnit.101 project
- Write unit tests for Math.cs
 - Add
 - Subtract
 - Divide
 - Multiply

Golf Score App

Create a program to:

- Book a Tee time



XUnit Workshop

Objective

- Add TimeSlots to Course