2.12 Static Fakes



This section will guide you to:

* Create a Windows Class library project for creating static fakes in target classes to test.

**Development Environment**

* Windows 10
* Visual Studio 2019 Community Version

This guide has eleven subsections, namely:

2.12.1 Creating a Windows Class library project for creating target classes to test

2.12.2 Adding a static class with a static method for testing purposes

2.12.3 Adding a wrapper class for the static class

2.12.4 Creating another Windows Class library project for running NUnit tests

2.12.5 Setting up NUnit as part of a Visual Studio project

2.12.6 Setting up NUnit3TestAdapter as part of a Visual Studio project

2.12.7 Setting up Moq as part of a Visual Studio project

2.12.8 Writing a test to test the static method of the static class

2.12.9 Building the project

2.12.10 Running all the tests in Test Explorer

2.12.11 Pushing the code to your GitHub repositories

**Step 2.12.1:** Creating a Windows Class library project for creating target classes to test

* Open Visual Studio.
* From the top menu, click **File->New->Project**
* Select **(Class Library (.NET Framework)** from the displayed project types
* Click **Next**
* Input **Project Name** as Phase4Section2.5 and click **Create**
* This will create the Windows Class library project

**Step2.12.2:** Adding a static class with a static method for testing purposes

* From the **Solution Explorer,** right click **Phase4Section2.5** and click **Add->Class**
* Input name as SCalculator.cs and click **Add**
* Add the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Text;

**using** System.Threading.Tasks;

**namespace** Phase4Section2.\_5

{

**static** **class** SCalculator

{

**static** **public** **int** add(**int** x, **int** y)

{

**return** x + y;

}

}

}

**Step 2.12.3:** Adding a wrapper class for the static class

* From the **Solution Explorer,** right click **Phase4Section2.5** and click **Add->Class**
* Input name as SCalcWrapper.cs and click **Add**
* Add the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Text;

**using** System.Threading.Tasks;

**namespace** Phase4Section2.\_5

{

**public** **class** SCalcWrapper

{

**public** **int** add(**int** x, **int** y)

{

**return** SCalculator.add(x, y);

}

}

}

**Step 2.12.4:** Creating a Windows Class library project for running NUnit tests

* In **Solution Explorer,** right click the Solution item and click **Add->New Project**
* Select **(Class Library (.NET Framework)** from the displayed project types
* Click **Next**
* Input **Project Name** as Phase4Section2.5.Tests and click **Create**
* This will create the Windows Class Library project for using NUnit

**Step 2.12.5:** Setting up NUnit as part of the project

* From the **Solution Explorer,** right click **Phase4Section2.5.Tests** and click **Manage Nuget Packages**
* Click on **Browse** tab and search for NUnit
* Click on the NUnit item and click **Install**
* This will set up NUnit for the project

**Step 2.12.6:** Setting up NUnit3TestAdapter as part of the project

* From the **Solution Explorer,** right click on **Phase4Section2.3** and click **Manage Nuget Packages**
* Click on **Browse** tab and search for NUnit3TestAdapter
* Click on the NUnit3TestAdapter item and click **Install**

**Step 2.12.7:** Setting up Moq as part of the project

* From the **Solution Explorer,** right click **Phase4Section2.5.Tests** and click **Manage Nuget Packages**
* Click on **Browse** tab and search for Moq
* Click on the Moq item and click **Install**
* This will set up Moq for the project

**Step 2.12.8:** Writing a test to test the static method of the static class

* From the **Solution Explorer,** expand **Phase4Section2.5.Tests** and double click **Class1.cs**
* Add the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.IO;

**using** System.Linq;

**using** System.Text;

**using** System.Threading.Tasks;

**using** Moq;

**using** NUnit.Framework;

**namespace** Phase4Section2.\_5.Tests

{

[TestFixture]

**public** **class** Class1

{

[Test]

**public** **void** StaticFake()

{

**int** x = 10, y = 20;

**var** wrapper = **new** SCalcWrapper();

Assert.That(wrapper.add(x, y), Is.EqualTo(x+y));

}

}

}

**Step 2.12.9:** Building the Project

* From the top menu, choose **Build->Build Solution**
* If any compilation errors are shown, fix them as required

**Step 2.12.10:** Running all the tests in Test Explorer

* From the top menu, choose **Test->Windows->Test Explorer**
* In Test Explorer, click on **Run All**
* This will execute the tests and shows the results in Test Explorer

**Step 2.12.11:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master