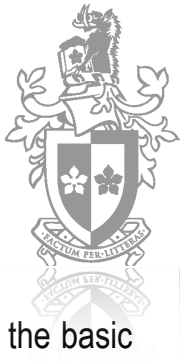
Faculty of Science, Engineering and Technology



Object Oriented Programming

Setting Up Unit Testing

Numerous tasks will ask you to implement unit testing. In this document we outline the basic procedures for installing and using unit testing in Visual Studio Community Edition.

Unit Testing setups

The unit testing package you use will probably depend on the operating system you are working with. On Mac, the NUnit framework is your best choice for unit testing. This is a NuGet package, installable from within Visual Studio across all platforms, but with some slight differences in installation which I will try and outline here. Note that Microsoft provides its own unit testing framework with Visual Studio, which if you installed the .NET desktop development Framework, is your easiest choice - though the syntax is slightly different to NUnit. Instructions for installing each are given below, noting that differences between setups and installations may mean your experience is a bit different. Remember, Google is your friend, and so are your tutors.

NUnit for Mac

The following has been tested and verified for Visual Studio for Mac Community (8.2.1), installed on Mac OS High Sierra.

1. Open up your project in Visual Studio for Mac Community.
2. Under the Project menu item, select Add NuGet package
3. Search for the NUnit package. Click on it and select “Add package” at the bottom right of the window.
4. Search for NUnit3TestAdapter. Click on it and install select “Add package” at the bottom right of the window. .
5. To run unit tests (assuming you have implemented some), under the “Run” menu item, select “Run unit tests”. If results of tests do not appear, click “Test Results” at bottom right of IDE.

MS Unit testing for Windows (using .NET Framework)

Assuming you have installed the .NET desktop development Framework (and not .NET core!), then the easiest way to go is to use the inbuilt unit testing. Microsoft provides its own unit testing framework, which you can add to your projects and use. This will require slightly different syntax to that shown in lectures, however it is quite similar, and intuitive.

For instructions on its installation and use, visit this page.

NUnit testing for Windows (using .NET core)

This is here for completeness only, but if you are keen to work with .NET core, then here the instructions for installing. Note the this has been tested and verified for Visual Studio Community 2017, installed on Windows 10. *Note also that your preferred search engine is always your best friend when things don't quite match.*

1. Open up your project in Visual Studio Community Edition.
2. Under the Tools menu item, select NuGet package manager and then select Manage NuGet Packages for Solution …
3. Select the "Browse" tab and then Search for the NUnit package. Click on it and install.
4. Under "Browse" again, search for NUnit3TestAdapter. Click on it and install.
5. Under "Browse" again, search for Microsoft.NET.Test.SDK, Click on it and install.
6. Close the project in the IDE and open up File Explorer
7. Locate the folder for the project. By default it is often in the source\repo\ folder inside your home folder.
8. Open the <project_name.csproj> file in a file editor (not in Visual Studio, but something like NotePad).
9. You will see marked-up text within the <PropertyGroup> block. You should ensure the following is in this block:

```
<PropertyGroup>
<OutputType>Exe</OutputType>
<TargetFramework>netcoreapp2.1</TargetFramework>
<RootNamespace>insert the project name</RootNamespace>
<GenerateProgramFile>false</GenerateProgramFile>
</PropertyGroup>
```

10. Once done, save file and re-open solution in Visual Studio. You should now be able to implement and run unit tests as per the lecture.
11. To run unit tests, assuming you have implemented some,