https://raygun.com/blog/unit-testing-frameworks-c/

**Creating Test Classes and Tests**

All test classes that contain tests should be public. Here are some reminders of the MS Test annotations...

* [TestClass()] - Makes the class a test class
* [TestMethod] - Marks the function as a test function
* [Description("Test the set up of command parameters")] - Add a description about the intention of the test
* [Ignore] - Don't run the test as default. Ignore the test if you want to keep the test code, but only want to test it occasionally. This is especially useful if it leaves a device in a undesired state, i.e., reboot on an ios device, or if the test takes ages to run.
* [ExpectedException] - Use this annotation if you expect your test to throw an exception. Note if your test throws more than one exception in different places, then see the tip below for using try and catch with Fail instead.

**Test Conditions**

When testing, use the Assert class. Remember you are making sure something is true. For example, Assert(This is true).

Note that you write an exception with what you expect as the first parameter and what you actually got as the second. The arguments are ordered, if you swap them around, you will get false information, i.e., Expected null, got a valid reference is not the same as Expected a valid reference and got a null!

Here are some Assert functions:

* IsTrue - Is the statement true
* IsFalse - Is the statement not true
* AreEqual - Are the two equal
* IsNull - Is the reference null