# COMP-10066 – Assignment 4

Mark Yendt, Mohawk College 2019

## Assignment

Complete the following assignment and submit electronically using the assignment folder in **MyCanvas**. See the course Calendar for the exact date and time of the submission. This assignment must be done individually.

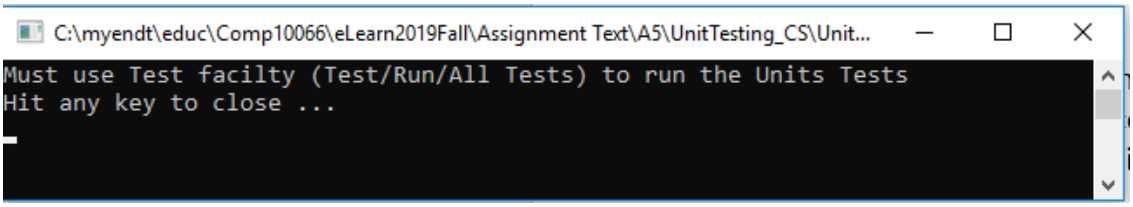
## Background

You have been assigned the task of writing unit tests to confirm the operation of all of the methods in the C# file included with this assignment (see the Assignment 4 folder on MyCanvas).

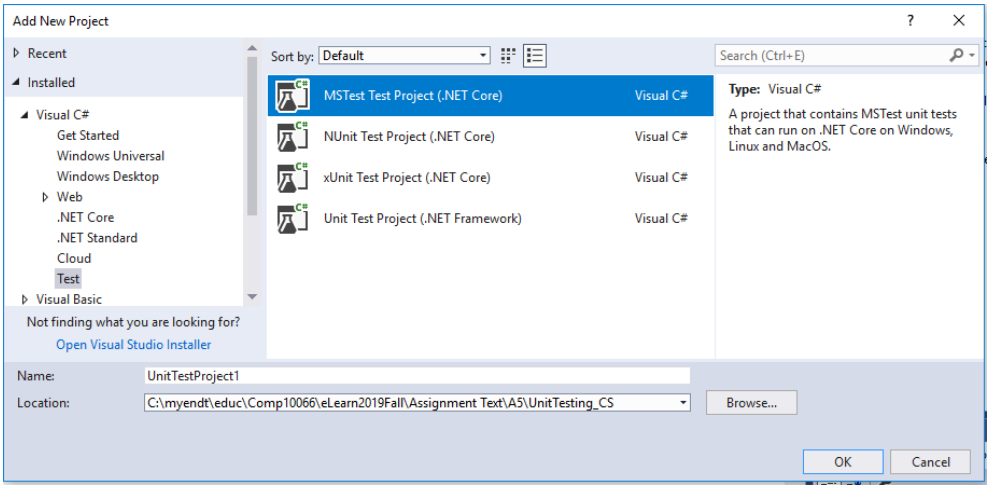
The **ArraySetUtilities** class contains five (5) static methods that require testing. **Your tests should be independent so that the results of one test will not impact any other test**. This is important as the testing harness will run individual tests in a random order by default. You can control the order if you must, but the general design rule in unit testing is to keep test cases isolated. All of the methods have been documented using C# documentation comments. Use this information to build appropriate unit tests for each of the methods.

## Assignment Set-Up

- You must use Visual Studio to create the solution for this assignment.
- Complete these steps:
    1. Start Visual Studio and Create a new project
    2. Download the code from the Assignment 4 folder (ArraySetUtilities.cs) and add the code to the project.
    3. Run the project. The output will appear in a console window that should look as follows:



4. In Visual Studio you will now need to add a new project to the solution. To do this go to the Solution Explorer / right click on the solution file and select Add/New Project, set the Project Type to Test / MSTest as shown below:




5. You will need to add a reference to the new project from the ArraySetUtilities project. On the Unit Testing Project right click then select Add/Reference and select the project (there should only be one project listed).
6. A default unit test file will be created for you. You should change the code in the default unit test file to the code provided below:

```
namespace COMP10066Lab5.Tests
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestIsUnique_unique()
        {
            List<int> arraySet = new List<int>() { 12, 22, 33, 44, 55 };
            bool expResult = true;
            bool result = ArraySetUtilities.IsUnique(arraySet);
            Assert.AreEqual(expResult, result,
            "Unique Array test returning false - Test array in unique");
        }
    }

}
```

7. **Ensure that the namespace of both classes is the same.**

8. Go to the Test menu and select Run/All Tests. You should see that the tests have passed. Make sure that this works before proceeding to the remainder of the assignment.

## Tasks

- There are bugs in the code presented. Any and all bugs that you uncover must be PROVEN with test case(s) that fail. Any failed test case is a bug.
- Summarize all bugs found in a comment at the top of the C# Unit Test file. Include a description of the bug, what test case shows the bug and a recommendation on how the problem should be corrected.
- Ensure that the test cases will provide 100% code coverage.
- Ensure that you consider large and small data sets in your test cases.
- Ensure all test cases developed will run in under 5 seconds (any individual test case).
- Confirm all of the return values possible for each of the methods. This will require that you create more than one test for each method.
- With each test case, include a one line comment above the method that provides the reason for the test. Each test case should have a unique (different from other test cases) reason.

## Submission / Evaluation

- Your submission for this assignment must be a single ZIP file with the entire Visual Studio Solution that will contain both projects (C# ArraySetUtilties Project and MSTest project).
- Unit test for each of the methods provided (75%)
    - 100 % code coverage is required for full marks
- Summary of bugs found in a comment at the top of the Unit Test C# file, with a recommendation on how the problem should be corrected (25%)