

Student name: Aldalton Choo Chien Khin

Student ID: 101212783

## Pass task 10.1: Unit Testing

### Part 1

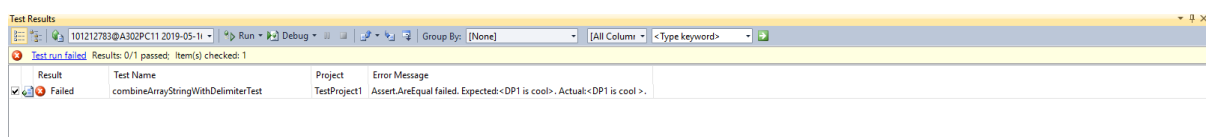
```
public void combineArrayStringWithDelimiterTest()
{
    StringUtilities target = new StringUtilities(); // TODO: Initialize to an appropriate value
    string[] stringArray = {"DP1","is","cool"}; // TODO: Initialize to an appropriate value
    string delimiter = " "; // TODO: Initialize to an appropriate value
    string expected = "DP1 is cool"; // TODO: Initialize to an appropriate value
    string actual;
    actual = target.combineArrayStringWithDelimiter(stringArray, delimiter);
    Assert.AreEqual(expected, actual);
    //Assert.Inconclusive("Verify the correctness of this test method.");
}

/// <summary>
/// A test for replaceSpacesInStringByHyphens
/// </summary>
[TestMethod()]
public void replaceSpacesInStringByHyphensTest()
{
    StringUtilities target = new StringUtilities(); // TODO: Initialize to an appropriate value
    string aString = "DP1 is cool"; // TODO: Initialize to an appropriate value
    string expected = "DP1-is-cool"; // TODO: Initialize to an appropriate value
    string actual;
    actual = target.replaceSpacesInStringByHyphens(aString);
    Assert.AreEqual(expected, actual);
    //Assert.Inconclusive("Verify the correctness of this test method.");
}
```

Figure (1) Both test method for combineArrayStringWithDelimiter function and replaceSpacesInStringByHyphens function

```
public string combineArrayStringWithDelimiter(string[] stringArray, string delimiter)
{
    string str = default(string);
    foreach (string item in stringArray)
    {
        str += item + delimiter;
    }
    return str;
}
```

Figure (2) combineArrayStringWithDelimiter function before fixing it to meet the test criteria.



The screenshot shows a 'Test Results' window with a table containing one row of test results. The table has four columns: Result, Test Name, Project, and Error Message. The 'Result' column shows a red circle with a white 'X' and the word 'Failed'. The 'Test Name' column contains 'combineArrayStringWithDelimiterTest'. The 'Project' column contains 'TestProject1'. The 'Error Message' column contains 'Assert.AreEqual failed. Expected:<DP1 is cool>. Actual:<DP1 is cool >.'.

Result	Test Name	Project	Error Message
Failed	combineArrayStringWithDelimiterTest	TestProject1	Assert.AreEqual failed. Expected:<DP1 is cool>. Actual:<DP1 is cool >.

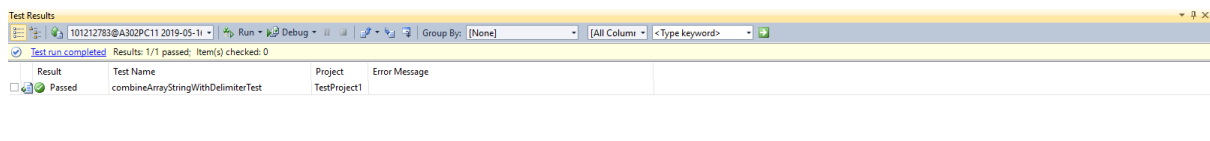
Figure (3) Test error for combineArrayStringWithDelimiter function

```

public string combineArrayStringWithDelimiter(string[] stringArray, string delimiter)
{
    string str = default(string);
    foreach (string item in stringArray)
    {
        str += item + delimiter;
    }
    return str.TrimEnd();
}

```

Figure (4) combineArrayStringWithDelimiter function after fixing it to meet the test criteria.



Result	Test Name	Project	Error Message
Passed	combineArrayStringWithDelimiterTest	TestProject1	

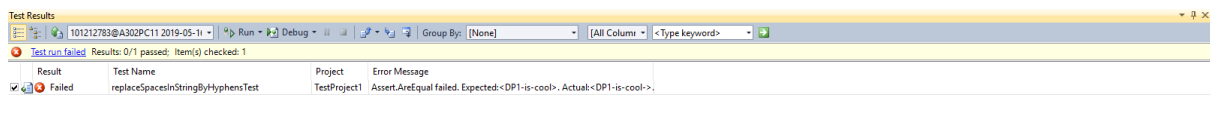
Figure (5) Test passed for combineArrayStringWithDelimiter function

```

public string replaceSpacesInStringByHyphens(string aString)
{
    string[] stringArray = aString.Split(null);
    return combineArrayStringWithDelimiter(stringArray, "-");
}

```

Figure (6) replaceSpacesInStringByHyphens function before fixing it to meet the test criteria.



Result	Test Name	Project	Error Message
Failed	replaceSpacesInStringByHyphensTest	TestProject1	Assert.AreEqual failed. Expected:<DP1-is-cool>. Actual:<DP1-is-cool>.

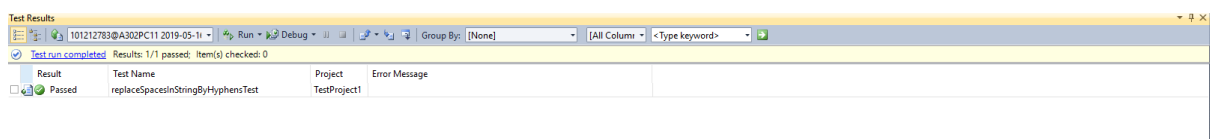
Figure (7) Test error for replaceSpacesInStringByHyphens function

```

public string replaceSpacesInStringByHyphens(string aString)
{
    string[] stringArray = aString.Split(null);
    return combineArrayStringWithDelimiter(stringArray, "-").TrimEnd('-');
}

```

Figure (8) replaceSpacesInStringByHyphens function after fixing it to meet the test criteria.



Result	Test Name	Project	Error Message
Passed	replaceSpacesInStringByHyphensTest	TestProject1	

Figure (9) Test passed for replaceSpacesInStringByHyphens function

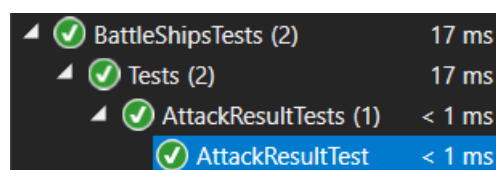
## Part 2

```
namespace Tests
{
    [TestClass()]
    0 references
    public class AttackResultTests
    {
        [TestMethod()]
        0 references
        public void AttackResultTest()
        {
            AttackResult attackResult = new AttackResult(ResultOfAttack.GameOver, "Game over", 0, 0);

            string result = attackResult.ToString();
            string outcome = "Game over";
            Assert.AreEqual(result, outcome);
        }
    }
}
```

Figure (10) Unit testing for AttackResult function

For the battleship testing I decided to test whether the AttackResult function is working as intended. In the unit testing class, I called the AttackResult function and assigned the values into a new variable called attackResult where the game is Over so ResultOfAttack.GameOver is passed into the function value and the string “Game over” will appear. I then passed on the value of attackResult into a new string variable called result and added the .ToString() function so that it will convert the values into a string. I then created another string variable called outcome and assigned the string value with “Game over” this is so that I can compare the result of the function and the expected outcome that the function should produce. I then use Assert.AreEqual function to compare both result and outcome. If both variables are equal, then the test is a success otherwise the test is a failure. The result of the test is a success as it is shown the figure (11).



The image shows a test runner interface with a tree view of test results. The tree is expanded to show the 'AttackResultTest' method, which has a green checkmark icon and a duration of '< 1 ms'. The parent nodes are 'AttackResultTests (1)' and 'Tests (2)', both also showing green checkmarks and durations. The root node is 'BattleShipsTests (2)' with a green checkmark and a duration of '17 ms'.

Test Name	Duration
BattleShipsTests (2)	17 ms
Tests (2)	17 ms
AttackResultTests (1)	< 1 ms
AttackResultTest	< 1 ms

Figure (11) Test success for AttackResultTest Unit testing.