## **Objectives**

* Explain the benefit of writing parameterized test cases.

In this hands-on lab, you will learn how to:

* Implement TestCase attribute that serves the dual purpose of marking a method with parameters as a test method and providing inline data to be used when invoking that method.

## **Prerequisites**

The following is required to complete this hands-on lab:

* Visual Studio 2017 and above Community Edition
* NUnit
* NUnit3TestAdapter

## **Notes**

Estimated time to complete this lab: **30 minutes.**

Create a Unit Test Project using NUnit Framework for the following requirement.

Create **LeapYearCalculatorLib** application that tells the user whether the entered year is a leap year or not. Also, it checks the input value to make sure that the data given is a valid year.

* If the given year is a Leap Year, the program will output 1. If it’s not, then the program will result 0.
* Any value between 1753 and 9999 (both inclusive) will be a valid year. Violation of this rule will result -1.

**Recommendations:**

Test Project Name:*<ClassLib\_Project>.Tests*

Test Class Name: *<SUT>Tests*

Test Method Name:  *UnitUnderTest\_Scenario\_ExpectedOutcome*

After writing the above test methods, run the tests and assert the results with that of the success/failure messages.

**Constraints:**

* *Enforce the Single Assertion Rule*
* *Use Assert.That()*

**Steps to perform:**

1. Create a Class Library project in the same solution which is provided and name it as suggested.
2. Rename the class file name (<SUT>Tests.cs).
3. Add the assembly reference of the source project to the test project.
4. Additionally add the reference of both NUnit and NUnit3TestAdapter in the test project using NuGet Package Manager (NPM).
5. Write the suggested test methods.
6. Run your tests.
7. Break the test by modifying the source project functionality.
8. Rerun the test.
9. Observe the test result.