# Objectives

* This lab will help you become skilled at writing automated unit tests using the NUnit framework.
* Explain & Demonstrate TestCaseSource to minimize amount of code used for testing.

Create a Unit Test Project using NUnit Framework for the following requirement. Click [here](https://cognizantonline.sharepoint.com/:u:/r/sites/GTP-Solutions/Gencsharepath/Shared%20Documents/Internship2020/FSE/DotNet/02%20-%20NUnit,%20C%23%204.5,%20ASP.Net%20Core/Handson/FourSeasonsLib.zip?csf=1&web=1&e=VY3YQa) to download the source project **FourSeasonsLib**.

Following is the business scenario implemented in the source project.

* The function accepts a month name and returns what season in that month (both inclusive) based on certain logic. The criteria is elaborated in the matrix below.

|  |  |  |
| --- | --- | --- |
| **Seasons** | **Month** | **Climate** |
| Spring | February to March | Sunny and pleasant |
| Summer | April to June | Hot |
| Monsoon | July to September | Wet, hot and humid |
| Autumn | September to November | Pleasant |
| Winter | December to January | Very Cool |

Make sure that you are not writing multiple test methods that are equal to the number of execution paths. Your focus should be of writing minimum code while unit testing.

Try both straight forward and alternate ways of working with the TestCaseSource attribute for the above scenario.

**Recommendations:**

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

Test Class Name: *<SUT>Tests*

Test Method Name:  *UnitUnderTest\_Scenario\_ExpectedOutcome*

**Note:**

* *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 ConverterLib project to the test project.
4. Additionally add the reference of NUnit, NUnit3TestAdapter and Moq 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.