# Objectives

* This lab will help you become skilled at writing automated unit tests using the NUnit framework.
* Explain & demonstrate various NUnit custom attributes to identify tests
* Explain & Demonstrate on CollectionAssert

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/CollectionsLib.zip?csf=1&web=1&e=LQC8Tc) to download the source project.

You have been given a source project called **CollectionsLib** that deals with set of collection objects. Write test methods for the below scenarios. Make sure that your tests pass. You may modify the collection values in the source project in order to make the test passed. Use the appropriate assert functionalities.

The method called **GetEmployees** returnsa collection of Employee object.

*Scenario 1*

* Ensure that there is no null value in the collection

*Scenario 2*

* Verify whether the employee having his/her id 100 exists in the collection.

*Scenario 3*

* Check whether the GetEmployees function returns only unique employees. If employee id is different in every employee object, then the list is considered as unique list. You may modify the source project so that you can achieve the goal here.

**Hint:** Override Equals() & GetHashCode() methods in Empoyee class.

*Scenario 3*

* Both **GetEmployees()** and **GetEmployeesWhoJoinedInPreviousYears()** return a set of employee object. Verify whether all items in both the collections are same or not.

Try both Classic Model as well as Constraint Model of Assertions while writing test cases.

**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 UtilLib 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.