Assignment # 1

Name: Muhammad Fahad

ID: FA19-BSSE-0014

Course: Test Driven Development

Section: AM

Teacher: KAMRAN CS

Difference between MS Test / X-Unit / N-Unit:

- **Isolation of Tests:**

  xUnit framework provides much better isolation of tests in comparison to NUnit and MSTest frameworks. For each test case, the test class is instantiated, executed, and is discarded after the execution. This ensures that the tests can be executed in any order as there is reduced/no dependency between the tests. Executing each test as a separate instance minimizes the chances of one test causing the other tests to fail!

- **Extensibility:**

  When we do NUnit vs. XUnit vs. MSTest, extensibility plays an important role in choosing a particular test framework. The choice might depend on the needs of the project, but in some scenarios, extensibility can turn the tables around for a particular test framework. When compared to MSTest and NUnit frameworks, xUnit framework is more extensible since it makes use of [Fact] and [Theory] attributes.

- **Initialization and De-initialization:**

  The NUnit uses [SetUp], [TearDown] pairs whereas MSTest uses [TestInitialize], [TestCleanup] pairs for setting up the activities related to initialization & de-initialization of the test code. On the other hand, xUnit uses the class constructor for the implementation of steps related to test initialization and IDisposable interface for the implementation of steps related to de-initialization.

  xUnit starts a new instance per test, whereas, in NUnit & MSTest frameworks, all the tests execute in the same Fixture/Class.

- **Assertion mechanism:**

  xUnit framework makes use of Assert.Throws instead of [ExpectedException] which is used in NUnit and MSTest. The drawback of using [ExpectedException] is that the errors might not be reported if they occur in the wrong part of the code. For example, if assert has to be raised for Security Exception, but Authentication Exception occurs, [ExpectedCondition] will not raise assert.

- **Parallel test execution**

  All the three C# unit testing frameworks support parallel test execution and are well-suited for Selenium automation testing as throughput plays a major role in automation testing. Below are the ways in which parallelism can be achieved in each of the test frameworks.