Testing

Nunit

Unit Testing

- Every software is composed of various modules.
- Each module is composed of various classes.
- Classes composed of various functions.
- Function is the smallest unit of code in the application.
- When we test individual function behavior without touching any other functions and determine whether it works exactly as per the requirements or not that is called Unit Testing.

Advantages of Unit Testing

- Defects found early in development life cycle
- Reliable Code
- Maintainable code
- Faster testing by only single click of action

NUnit

- NUnit is a unit testing framework for .NET. It is the most used framework for writing unit test cases.
- We can write testing code in either C# or VB.NET.
- It is suggested to write testing code in different assemblies called Test Assemblies.
- Test Runners are UI tool which run NUnit test cases and show the result of test cases whether they are passed or failed.

NUnit

- NUnit is very easy to use.
- It only provides some custom attributes and some static Assert classes.
- With the combination of custom attributes and static classes, unit test cases can be easily written.
- Some of the custom attributes are:
    - TestFixture
    - Setup
    - TearDown
    - Test
    - Category
    - Ignore
    - TestCase
    - Repeat
    - MaxTime

NUnit TestFixture

- NUnit TestFixture attribute is a class level attribute, and it indicates that this class contains NUnit Test Methods.

- Parameterized / Arguments TestFixtures
  - Sometimes our NUnit class needs arguments.
  - We can pass arguments to TestFixture class through constructors.
- NUnit TestFixture Inheritance
  - TestFixture attribute supports inheritance that means we can apply TestFixture attribute on base class and inherit from derived Test Classes.
  - A base class can be an Abstract class.

TestFixture Restrictions

- It can only place on class.
- If no arguments is provided in TestFixture attribute, then class must have default constructor.
- If arguments is provided in TestFixture attribute, then class must have matching constructor.
- We can place multiple TestFixture attributes on a single class.
- TestFixture attribute can be inherited
- We can provide generic arguments to TestFixture class.
- We can apply TestFixture attribute on abstract class.

NUnit TestCase

- NUnit TestCase Arguments / Parameters
  - TestCase arguments are used when we must use same test case with different data.
- Author Property
  - We can specify author name in the test method who has written the test case
- TestName property
  - TestName property is used when we must use different name than the specified test method name
- Ignore TestCase
  - Sometimes we need to ignore our test case reason may be code is not yet complete. So, we can use Ignore property to mark test case as ignore.

NUnit TestCase Array

- There is one restriction on array type.
- Array type must be a constant expression.
- Array types are limited to below types:
  - bool
  - byte
  - char
  - short
  - int
  - long
  - float
  - double
  - Enum
  - object
- For passing other data types like string, use either object type or can use NUnit TestCaseSource.

NUnit Assert

- NUnit Assert class is used to determine whether a particular test method gives expected result or not.
- In a test method, we write code the check the business object behavior.
- That business object returns a result.
- In Assert method we match the actual result with our expected result.
- If result comes according to our expected result, then our test case is passed else failed.

NUnit Assert

- Constraint Model
  - NUnit provides a new Constraint Model to improve the test method readability.
  - In constraint model, we use a single method "That" and specify constraints to check our expected response.
- That method applies a constraint to the actual value. If a constraint is satisfied our test case is succeeded else failed.
- Helper Classes
  - Below are helper classes to provide a constraint to assert the method.
    - Is
    - Has
    - Contains
    - Does
    - Throws

NUnit Assert

- Constraint Categories
  - These constraints can be divided into eight categories:
    - Comparison
    - String
    - Collection
    - Conditional
    - Compound
    - Directory/File
    - Type/Reference
    - Exceptions

Demo

Thank you

Innovative Services

Passionate Employees

Delighted Customers