Multi Choice Type Question In NUnit, what attribute is used to mark a method as a test method? Marks : 1 Negative Marks : 0 image Answer here [TestMethod] [Test] [TestCase] [Fact]

The correct answer is:

**[Test]** ✅

Explanation:  
In **NUnit**, the [Test] attribute is used to mark a method as a test method.

* [TestMethod] → Used in **MSTest** framework.
* [TestCase] → Used in NUnit to provide parameterized test cases.
* [Fact] → Used in **xUnit** framework.

**You said:**

Multi Choice Type Question How are test methods recognized and executed by NUnit in an ASP.NET Core project? Marks : 1 Negative Marks : 0 image Answer here By running all public methods in the test project By inspecting the method names for keywords like "Test" or "Should" By explicitly defining test methods in the NUnit configuration file Test methods are automatically discovered and executed by NUnit

The correct answer is:

**Test methods are automatically discovered and executed by NUnit** ✅

Explanation:  
NUnit automatically **discovers test methods** by scanning for methods marked with the [Test] attribute (or other relevant NUnit attributes like [TestCase], [Theory]). You don’t need to list them manually or rely on method names.

**You said:**

Multi Choice Type Question In unit testing, what is the purpose of an assertion? Marks : 1 Negative Marks : 0 image Answer here To define the test setup and tear-down methods To specify the test cases to execute To verify the expected behavior of a test method To group multiple test methods together

The correct answer is:

**To verify the expected behavior of a test method** ✅

Explanation:  
In unit testing, an **assertion** checks whether the actual result of a method matches the expected result. If the assertion fails, the test fails, indicating that the code does not behave as intended.

* Defining setup/tear-down → [SetUp] / [TearDown] in NUnit.
* Specifying test cases → [TestCase].
* Grouping test methods → [TestFixture].

**You said:**

Multi Choice Type Question Which of the following is a commonly used assertion method in NUnit to check for equality between two values? Marks : 1 Negative Marks : 0 image Answer here Assert.True() Assert.False() Assert.AreEqual() Assert.IsNotNull()

The correct answer is:

**Assert.AreEqual()** ✅

Explanation:  
In **NUnit**, Assert.AreEqual(expected, actual) is used to verify that two values are equal.

* Assert.True() → Checks if a condition is true.
* Assert.False() → Checks if a condition is false.
* Assert.IsNotNull() → Checks that an object is not null.

**You said:**

Multi Choice Type Question In NUnit, which attribute is used to mark a method as an ignored or skipped test case? Marks : 1 Negative Marks : 0 image Answer here [Ignore] [Exclude] [Skip] [Ignored]

The correct answer is:

**[Ignore]** ✅

Explanation:  
In **NUnit**, the [Ignore] attribute is used to mark a test method so that it is **skipped during test execution**. You can also provide an optional reason, e.g., [Ignore("Not implemented yet")].

* [Exclude], [Skip], [Ignored] → Not valid NUnit attributes.

**You said:**

Multi Choice Type Question What is the purpose of using the [SetUp] attribute in NUnit test methods? Marks : 1 Negative Marks : 0 image Answer here To group multiple test methods together To define the test case data for parameterized tests To specify the test setup method to run before each test method To set up the test environment before running the tests

The correct answer is:

**To specify the test setup method to run before each test method** ✅

Explanation:  
In **NUnit**, the [SetUp] attribute is used to mark a method that should **run before each test method** in a test class. It is commonly used to initialize objects or set up the environment required for the tests.

* Grouping test methods → [TestFixture]
* Defining test case data → [TestCase]
* Setting up environment once for all tests → [OneTimeSetUp]

**You said:**

Multi Choice Type Question In NUnit, which assertion method is used to check that a condition is false? Marks : 1 Negative Marks : 0 image Answer here Assert.True() Assert.False() Assert.AreEqual() Assert.IsNull()

The correct answer is:

**Assert.False()** ✅

Explanation:  
In **NUnit**, Assert.False(condition) is used to verify that a given condition evaluates to **false**.

* Assert.True() → Checks that a condition is true.
* Assert.AreEqual() → Checks that two values are equal.
* Assert.IsNull() → Checks that an object is null.

**You said:**

Multi Choice Type Question What happens when an assertion fails in a test method during NUnit execution? Marks : 1 Negative Marks : 0 image Answer here The test method is automatically marked as ignored. The test method is retried with different parameters. The test method execution stops, and the test is considered failed. The test method execution skips to the next test.

The correct answer is:

**The test method execution stops, and the test is considered failed** ✅

Explanation:  
In **NUnit**, when an assertion fails, the current test method **immediately stops executing**, and the test is marked as **failed**. Other tests in the test suite continue to run independently.

* Test is not marked as ignored.
* NUnit does not automatically retry the test unless explicitly configured.
* Execution does not skip silently; failure is reported.

**You said:**

Multi Choice Type Question What is the benefit of using parameterized tests in NUnit for ASP.NET Core unit testing? Marks : 1 Negative Marks : 0 image Answer here It allows running the same test method multiple times with different input values. It reduces the execution time of test cases. It automatically generates test case data based on the code complexity. Parameterized tests are not supported in NUnit.

The correct answer is:

**It allows running the same test method multiple times with different input values** ✅

Explanation:  
In **NUnit**, parameterized tests (using attributes like [TestCase]) let you **run the same test logic multiple times** with different sets of input values, improving test coverage and reducing code duplication.

* It does not automatically generate test data.
* It may or may not reduce execution time.
* NUnit **does support** parameterized tests.

**You said:**

Multi Choice Type Question What is the purpose of using the Assert.Throws method in NUnit assertions? Marks : 1 Negative Marks : 0 image Answer here To verify if a test method throws an exception To check if a condition is true during test execution To assert equality between two objects To validate if a value is null

The correct answer is:

**To verify if a test method throws an exception** ✅

Explanation:  
In **NUnit**, Assert.Throws<ExceptionType>(delegate) is used to **verify that a specific exception is thrown** by a block of code or method during execution.

* Checking a condition → Assert.True() / Assert.False()
* Asserting equality → Assert.AreEqual()
* Validating null → Assert.IsNull()

**You said:**

Multi Choice Type Question What is the purpose of the [MaxTime] attribute in NUnit? Marks : 1 Negative Marks : 0 image Answer here To define the maximum time a test should take to execute. To specify the maximum number of test cases to run. To set a maximum time for all tests in a fixture. To log the execution time of tests.

The correct answer is:

**To define the maximum time a test should take to execute** ✅

Explanation:  
In **NUnit**, the [MaxTime(milliseconds)] attribute is used to **fail a test if it takes longer than the specified time** to execute. This helps identify performance issues in unit tests.

* It does not limit the number of test cases.
* It applies to individual tests, not necessarily all tests in a fixture.
* It does not just log execution time; it enforces a timeout.

**You said:**

Multi Choice Type Question Which NUnit attribute allows a test to expect a certain type of exception to be thrown? Marks : 1 Negative Marks : 0 image Answer here [ExpectedException] [Throws] [Assert.Throws] [Expected]

The correct answer is:

**[ExpectedException]** ✅

Explanation:  
In older versions of **NUnit**, the [ExpectedException(typeof(ExceptionType))] attribute was used to indicate that a test **expects a specific exception** to be thrown.

* [Assert.Throws] is a **method**, not an attribute, used inside the test body to assert exceptions.
* [Throws] and [Expected] are not valid NUnit attributes.

Note: In modern NUnit versions, it is recommended to use Assert.Throws<ExceptionType>() inside the test method instead of [ExpectedException].

**You said:**

Multi Choice Type Question How can you categorize tests in NUnit so they can be run selectively? Marks : 1 Negative Marks : 0 image Answer here By using the [TestCase] attribute By using the [Category] attribute By using the [Group] attribute By using the [Tag] attribute

The correct answer is:

**By using the [Category] attribute** ✅

Explanation:  
In **NUnit**, the [Category("CategoryName")] attribute is used to **categorize tests**, allowing you to run a specific group of tests selectively based on their category.

* [TestCase] → Used for parameterized tests.
* [Group] and [Tag] → Not valid NUnit attributes.

**You said:**

Multi Choice Type Question Which NUnit assertion is used to verify that an object reference is null? Marks : 1 Negative Marks : 0 image Answer here Assert.IsTrue() Assert.IsNull() Assert.IsFalse() Assert.AreEqual()

The correct answer is:

**Assert.IsNull()** ✅

Explanation:  
In **NUnit**, Assert.IsNull(object) is used to **verify that a given object reference is null**.

* Assert.IsTrue() → Checks if a condition is true.
* Assert.IsFalse() → Checks if a condition is false.
* Assert.AreEqual() → Checks if two values are equal.

**You said:**

Multi Choice Type Question What NUnit attribute can be used to run tests in a specific order? Marks : 1 Negative Marks : 0 image Answer here [Order] [Priority] [Sequence] [Rank]

The correct answer is:

**[Order]** ✅

Explanation:  
In **NUnit**, the [Order(int)] attribute is used to **specify the order in which test methods are executed** within a test fixture.

* [Priority], [Sequence], [Rank] → Not valid NUnit attributes.