**Q1. What is Unit Testing and how is it different from Functional Testing?**

- Unit Testing refers to testing the smallest units of code in isolation (typically individual methods). Functional Testing, on the other hand, verifies the system as a whole against functional requirements. Unit tests are white-box tests and are typically written by developers, while functional tests are black-box and often done by QA.

**Q2. What is the smallest unit to test in Unit Testing?**

- The smallest unit is usually a method or function.

**Q3. List various types of testing:**

- Unit Testing

- Functional Testing

- Integration Testing

- System Testing

- Regression Testing

- Acceptance Testing

- Performance Testing

- Load Testing

**Q4. What are the benefits of Automated Testing?**

- Faster execution

- Consistent results

- Reduces manual testing effort

- Enables frequent regression testing

- Supports CI/CD pipelines

- Improves software quality

**Q5. What is a loosely coupled & testable design?**

- A loosely coupled design means components have low dependency on each other. Testable design allows injecting dependencies (e.g., via constructor) to test components in isolation using mocking frameworks.

**Q6. What attribute is used to define a test class in NUnit?**

- [TestFixture]

**Q7. What attribute is used to define test methods in NUnit?**

- [Test]

**Q8. What attribute is used to define common setup code before each test?**

- [SetUp]

**Q9. What attribute is used to define common teardown code after each test?**

- [TearDown]

**Q10. What attribute is used to ignore a test?**

- [Ignore]

**Q11. What is the use of [TestCase]?**

- It allows you to define multiple sets of parameters for a single test method, enabling parameterized testing.