NUnit Handson Question and Answers

Q1. Explain the meaning of Unit testing and its difference on comparison with Functional testing

Unit testing is the process of testing individual pieces of code, such as methods or functions, to ensure they work as expected in isolation. In contrast, functional testing checks whether the overall application features work as intended from the user's perspective. Unit testing is developer-focused, faster, and more granular, while functional testing validates complete user flows.

Q2. Smallest unit to test mocking dependencies

The smallest unit to test is usually a method. If that method depends on external systems (like a database or email server), those dependencies should be mocked to isolate the logic and run tests quickly.

Q3. List various types of testing

Common types of testing include unit testing, functional testing, automated testing, and performance testing.

Q4. Understand the benefit of automated testing

Automated testing saves time, avoids repetitive manual testing, provides quicker feedback on changes, and helps prevent bugs from reaching production. It also supports continuous integration and deployment.

Q5. Explain what is loosely coupled and testable design

Loosely coupled design means parts of the code are independent and interact through interfaces or abstractions. This makes it easier to test each part individually without relying on the full system. Testable design encourages separation of concerns and avoids hard dependencies.

Q6. Write code that is NOT dependent on the class for data

```
public interface IDataProvider {
   string GetData();
}
public class Service {
   IDataProvider _dp;
   public Service(IDataProvider dp) { _dp = dp; }
   public string Process() => _dp.GetData();
}
```

Q7. Write your first testing program to validate a calculator addition operation

```
public class Calculator {
   public int Add(int a, int b) => a + b;
}
```

Q8. Understand the need of [SetUp], [TearDown], and [Ignore] attributes

[SetUp] is used to prepare objects or variables before each test. [TearDown] cleans up resources after each test. [Ignore] is used to skip specific tests temporarily, such as when they are not yet implemented or under revision.

Q9. Explain the benefit of writing parameterised test cases

Parameterized tests let you run the same test logic with multiple sets of input values. This reduces code duplication and improves test coverage by testing more scenarios with fewer lines of code.