**Superset id:- 6363303**

**Exercise 1: NUnit-Handson**

**Code:-**

**namespace CalcLibrary.Tests**

**{**

**[TestFixture]**

**public class CalculatorTests**

**{**

**private SimpleCalculator calc;**

**[SetUp]**

**public void Setup()**

**{**

**calc=new SimpleCalculator();**

**}**

**[TearDown]**

**public void TearDown()**

**{**

**calc.AllClear();**

**}**

**[Test]**

**[TestCase(5,3,8)]**

**[TestCase(10,-5,5)]**

**[TestCase(2,3,5)]**

**[TestCase(-1,-2,-3)]**

**[TestCase(3,1,4)]**

**[TestCase(991,9,1000)]**

**public void Test\_Addition(double a,double b,double expected)**

**{**

**double result=calc.Addition(a,b);**

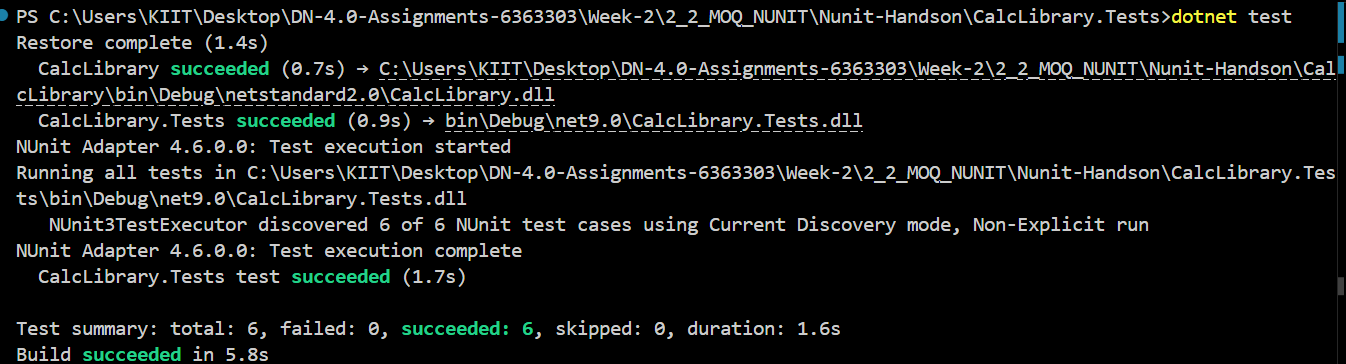
**Assert.That(result,Is.EqualTo(expected));**

**}**

**}**

**}**

**Output:-**

****

**Unit Testing:-**

Unit testing means checking a small part of your code, like one function or method, to see if it works correctly. It is usually done by developers during development.

**Difference Between Unit Testing and Functional Testing:-**

* Unit Testing checks one small piece of code like a method or function.
* Functional Testing checks a full feature to make sure the system behaves as expected when used.

Example:-

Unit testing = checking if Add(a, b) returns the correct result.

Functional testing = checking if the full “Login” feature works correctly.

**Types of Testing:-**

**Unit Testing** – Tests individual methods or small parts of the code.

**Functional Testing** – Tests complete features of the software.

**Automated Testing** – Tests that run automatically using tools or scripts.

**Performance Testing** – Checks how fast or stable the system is under load.

**Benefits of Automated Testing:-**

* **Saves time by running tests automatically.**
* **Makes it easy to find bugs early.**
* **You can test again and again without doing it manually.**
* **Helps developers make changes confidently.**

### **What is a Loosely Coupled and Testable Design?**

### A loosely coupled design means your code does not directly depend on other specific classes. This makes your code easier to test.

**Example of Tightly Coupled Code:**

public class Report {

private Database db = new Database();

}

**Example of Loosely Coupled Code:**

public class Report {

private IDatabase db;

public Report(IDatabase database) {

db = database;

}

}

**Meaning of [SetUp], [TearDown] and [Ignore]:-**

* **[SetUp] – Runs before every test. You can use it to prepare data or objects.**
* **[TearDown] – Runs after every test. You can use it to clean up.**
* **[Ignore] – Skip a test if you don’t want to run it right now.**

**Ayush Kumar (6363303)**