**WEEK-2**

Task-4

* Explain the meaning of Unit testing and its difference on comparison with Functional testing
  + Smallest unit to test mocking dependencies
* List various types of testing
  + Unit testing, Functional testing, Automated testing, Performance testing
* Understand the benefit of automated testing
* Explain what is loosly coupled & testable design
  + Write code that is NOT dependent on the class for data.
* Write your first testing program to validate a calculator addition operation
  + TestFixture, Test
* Understand the need of [SetUp], [TearDown] & [Ignore] attributes.
* Explain the benefit of writing parameterised test cases.
  + TestCase

Sol:

**Unit testing** involves testing the smallest parts (units) of an application, like individual methods or functions, often using mocks to isolate dependencies. It differs from **functional testing**, which tests the software against business requirements as a whole. Common testing types include **unit**, **functional**, **automated**, and **performance** testing. **Automated testing** saves time, ensures consistency, and catches regressions early. A **loosely coupled, testable design** avoids tight dependency between components, making code more flexible and easier to test. Using frameworks like NUnit

public class CalculatorTests {

[Test] public void Add\_TwoNumbers\_ReturnsSum() {

Assert.AreEqual(5, new Calculator().Add(2, 3));

}

}

Name:Ayushman Upadhyay

Email:2230164@kiit.ac.in

SuperSet Id: 6358125