NUnit Hands-On Project

# 🧾 Introduction

This document provides a step-by-step guide to building and running unit tests in Visual Studio using NUnit. You'll create a simple calculator class (`CalcLibrary`) and write unit tests for it in a separate test project (`CalculatorTests`).

# 🛠 Tools Required

1. Visual Studio (Community Edition or above)

2. NUnit

3. NUnit3TestAdapter

4. Microsoft.NET.Test.Sdk

# 📁 Project Structure

Solution 'CalcLibrary'  
├── CalcLibrary  
│ ├── Calculator.cs  
│ └── Program.cs (auto-generated, not used)  
└── CalculatorTests  
 ├── CalculatorTests.cs  
 └── packages.config

# 📌 CalcLibrary - Calculator.cs

This file contains the class to be tested.

namespace CalcLibrary  
{  
 public class Calculator  
 {  
 public int Add(int a, int b)  
 {  
 return a + b;  
 }  
 }  
}

# 🧪 CalculatorTests - CalculatorTests.cs

This file contains NUnit unit test cases for the Calculator class.

using NUnit.Framework;  
using CalcLibrary;  
  
namespace CalculatorTests  
{  
 [TestFixture]  
 public class CalculatorTests  
 {  
 private Calculator calculator;  
  
 [SetUp]  
 public void Init()  
 {  
 calculator = new Calculator();  
 }  
  
 [TearDown]  
 public void Cleanup()  
 {  
 calculator = null;  
 }  
  
 [Test]  
 public void SimpleAddTest()  
 {  
 int result = calculator.Add(2, 3);  
 Assert.That(result, Is.EqualTo(5));  
 }  
  
 [TestCase(10, 20, 30)]  
 [TestCase(0, 0, 0)]  
 [TestCase(-5, -3, -8)]  
 public void ParameterizedAddTest(int a, int b, int expected)  
 {  
 int result = calculator.Add(a, b);  
 Assert.That(result, Is.EqualTo(expected));  
 }  
  
 [Test, Ignore("Skipping this test for now.")]  
 public void IgnoredTestExample()  
 {  
 Assert.Fail("This test is ignored.");  
 }  
 }  
}

