# Date:15/09/2022

# LabSessionNo.:07

# Task 7:Design a test case for testing for Visual C# using NUNIT for Prime,Armstrong,Fibonocci Numbers

# Task 7: Testing visual c# using NUnit.

**Aim**: To perform testing visual c# using NUnit.

# Procedure:

**Steps**:

1. Creating Class Library.
   * Click on File and select New project
   * Click on Visual C# and rename it as Program1
   * Click on Finish
2. Add the following code

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Program2

{

public class primeFib

{

public Boolean prime(int n)

{

if (n == 1)

{

return false;

}

else if (n == 2)

{

return true;

}

else

{

for (int i = 2; i < n; i++)

{

if (n % i == 0)

{

return false;

}

}

}

return true;

}

public int fib(int n)

{

int number = n - 1;

int[] Fib = new int[number + 1];

Fib[0] = 0;

Fib[1] = 1;

for (int i = 2; i <= number; i++)

{

Fib[i] = Fib[i - 2] + Fib[i - 1];

}

return Fib[number];

}

public int armstrong(int n)

{

int r, sum = 0, temp;

temp = n;

while (n > 0)

{

r = n % 10;

sum = sum + (r \* r \* r);

n = n / 10;

}

if (temp == sum)

return 1;

else

return 0;

}

}

}

Add new class Library

* + Right click on Project in Solution explorer.
  + Click on Add and New Project(Class Library)
  + Type the following code

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using NUnit.Framework;

using Program2;

namespace Test

{

[TestFixture]

public class Class1

{

[Test]

public void prs()

{

primeFib p1 = new primeFib();

Assert.AreEqual(true, p1.prime(5));

Assert.AreEqual(false, p1.prime(8));

Assert.AreEqual(2, p1.fib(4));

Assert.AreEqual(8, p1.fib(7));

Assert.AreEqual(1,p1.armstrong(153));

}

[Test]

public void prs1()

{

primeFib p1 = new primeFib();

Assert.AreEqual(true, p1.prime(5));

Assert.AreEqual(false, p1.prime(8));

Assert.AreEqual(2, p1.fib(4));

Assert.AreEqual(8, p1.fib(7));

Assert.AreEqual(1,p1.armstrong(353));

}

}

}

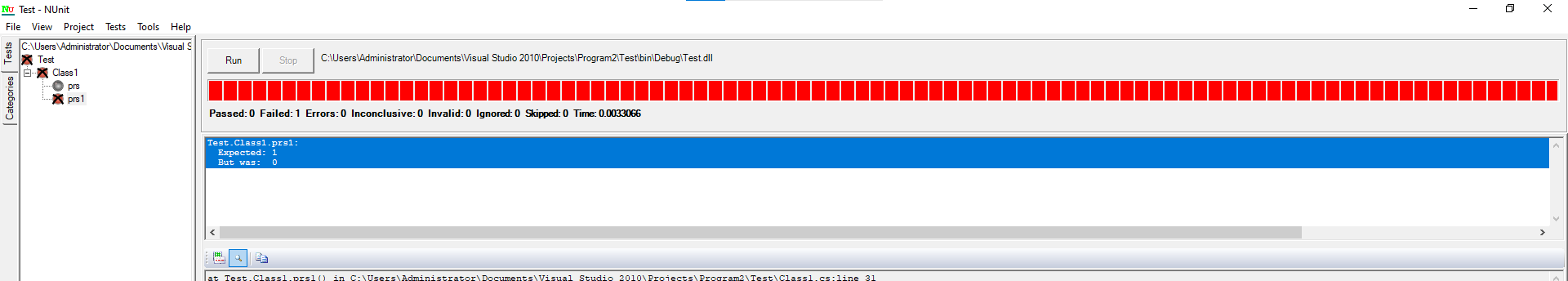
1. Add references
   * Right Click on Test and click on Add references
   * Add the Existing Project and also nunit.framework.dll
   * Click on OK
2. Build Solution
   * Click on Build
   * Now click on Build Solution
3. Testing using NUnit
   * Open NUnit and click on File
   * Click on Open project and select the Visual Project
   * Select the test code project.
   * Click on bin and then on debug and then on Test.nunit.dll

Click on Run

# Output: Pass Case

# 

**Fail Case**

****

# Test suite design:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Name**: NUnit Testing For Prime, Fibonacci, Armstrong | | | | | | |
| **Test case id**: ID\_7 **Test Designed by:T.Avinash**  **Test Priority**: low **Test Designed Date**:15/09/22  **Module Name**: Nunit Testing **Test Executed by**:**T.Avinash Test Title**: Blackbox testing  **Test Executed Date**: 15/09/22  **Description:** Test case for problem using NUnit Testing | | | | | | |
| **Pre-Condition**: User should give two input numbers and one expected output | | | | | | |
| **Stag e** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **status (Pass**  **/Fail)** | **Remarks** |
| 1 | Prime | 5 | true | true | Pass | Nil |
| 2 | Prime | 8 | false | false | Pass | Nil |
| 3 | Fibonacci | 4 | 2 | 2 | Pass | Nil |
| 4 | Fibonacci | 7 | 8 | 11 | Fail | Nil |
| 5 | Armstrong | 153 | 1 | 1 | Pass | Nil |
| 6 | Armstrong | 353 | 1 | 0 | Fail | Nil |
| **Post condition**: Expected result should match with value returned by function | | | | | | |

**Result:** Performing testing visual c# using NUnit has been implemented successfully.

**Evaluator’s Observation**

**Marks Secured: out of**

**Full Name of the Evaluator: Student’s Signature**

**Signature of the Evaluator: Date of Evaluation:**