**TITLE: ASSIGNMENT #1**

**NAME: GEETHU JOSEPH**

**STUDENT ID:8721314**

**SUBJECT: TESTING METHOLOGIES/SENG8040**

**DATE:8TH OCTOBER 2020**

Introduction:

In this assignment we are assigned to created a C# console application program that displays the triangle type and performed the unit testing on the code. After the unit testing gets completed, we have drawn the cyclomatic complexity diagram and find the value using below equation. Finally, we saved code to git repository and committed.

UNIT TESTING

Unit testing is a primary level of software testing where individual units / components of a software program is tested for finding the bugs/defects. The main aim is to validate that each unit of the software performs as designed. **UNIT TESTING** is also known as **COMPONENT TESTING.**

GITHUB

GitHub is a web-based version-control platform which is used by the software developers. Basically, it is used to store the source code of the project and track of all the changes to that made into the code. It allows developers to collaborate on a project more effectively and able to manage changes made by different programmers. GitHub assist developers to modify, adapt and improve quality of software through its public repositories with no cost effect.

Cyclomatic complexity

Cyclomatic complexity is calculated using the control flow graph of the program. The nodes of the graph correspond to indivisible groups of commands of a program. Nodes represents individual functions, modules, methods and classes within a program.

TriangleSolver

using System;

using System.Collections.Generic;

using System.Reflection.Metadata.Ecma335;

using System.Security.Cryptography.X509Certificates;

using System.Text;

namespace TestProgram\_Nunit

{

public static class TriangleSolver

{

public static string Analyst(int sideint1,int sideint2,int sideint3)

{

string result = string.Empty;

// if (sideint1 + sideint2 >= sideint3 || sideint1 + sideint3 <= sideint2 || sideint2 + sideint3 <= sideint1)

// {

// if ((sideint1 > 0) && (sideint2 > 0) && (sideint3 > 0))

//{

if ((sideint1 == sideint2) && (sideint2 == sideint3))

{

result = result + "Equilateral Triangle";

}

else if ((sideint1 == sideint2) && (sideint3 != sideint1) || (sideint2 == sideint3) && (sideint3 != sideint1) || (sideint3 == sideint1) && (sideint1 != sideint2))

{

result = result + "Isoceles Triangle";

}

else if (sideint1 != sideint2 && sideint2 != sideint3)

{

result = result + "scalene triangle";

}

// }

// }

// else

// {

Console.WriteLine("It given inputs doesnot form a triangle");

// }

Console.WriteLine("The arms of the triangle form a {0}", result);

return (result);

}

}

}

Main Code

using System;

using System.Security.Cryptography.X509Certificates;

using TestProgram\_Nunit;

namespace TestProgram\_Nunit

{

public class Program

{

static void Main(string[] args)

{

string cont;

string userInput = string.Empty;

bool Validmenuselect = false;

string resultString = string.Empty;

while (Validmenuselect == false)

{

do

{

Console.WriteLine("1.Enter the triangle dimensions");

Console.WriteLine("2.Exit");

Console.WriteLine("Please select an option by entering a valid number from the ablove list");

userInput = Console.ReadLine();

if (userInput != "1" && userInput != "2")

{

Console.WriteLine("Please enter a Valid option");

}

else if (userInput == "2")

{

Console.WriteLine("Exiting from the program");

System.Environment.Exit(0);

}

else

{

Validmenuselect = true;

string side1, side2, side3;

int sideint1, sideint2, sideint3;

Console.WriteLine("Enter the first side");

side1 = Console.ReadLine();

Console.WriteLine("Enter the second side");

side2 = Console.ReadLine();

Console.WriteLine("Enter the third side");

side3 = Console.ReadLine();

if ((int.TryParse(side1, out sideint1)) && (int.TryParse(side2, out sideint2)) && (int.TryParse(side3, out sideint3)) && (sideint1 > 0) && (sideint2 > 0) && (sideint3 > 0))

{

if (sideint1 + sideint2 >= sideint3 || sideint1 + sideint3 <= sideint2 || sideint2 + sideint3 <= sideint1)

{

Console.WriteLine("It given points forms arms of triangle");

TriangleSolver.Analyst(sideint1, sideint2, sideint3);

}

}

else

{

Console.WriteLine("It given inputs doesnot form a triangle!!");

}

}

Console.WriteLine("Do you want to continue?:(Y/N)");

cont = Console.ReadLine();

} while ((cont == "Y") || (cont == "y"));

}

}

}

}

Nunit code:

using System;

using TestProgram\_Nunit;

namespace TrianglSolverTest

{

public class TestTriangle

{

[TestFixture]

public class TestTriangle1

{

[Test]

public void CheckTriangle()

{

int Ftriangle = 20;

int Striangle = 30;

int Ttriangle = 10;

string expect= "Equilateral Triangle";

string actual = TriangleSolver.Analyst(Ftriangle, Striangle,Ttriangle);

Assert.AreEqual(expect, actual);

}

}

}

}

**References**

Kozier, B., Erb, G., Berman, A., Snyder, S. J., Buck, M., Yiu, L., & Stamler, L. L. (2014). *Fundamentals of Canadian Nursing: Concepts, Process, and Practice*. Toronto, Canada: Pearson Canada. (Delete this before you start your references. This example is just to set up the hanging indent feature.)

References are set up in a hanging indent paragraph format. If you enter your reference information following this paragraph and then delete these words, your paragraph will be formatted correctly. You can use the “enter” key to start a new paragraph, and then type your information.