Assignment -01: Unit Testing using NUnit

Aparna Tomar – 7802150

PROG8170

Professor Firouzeh Sharifi Lotfabad

GITHUB:

#1 Source Code for Program.cs

```
1.  using System;
2.  using System.Collections.Generic;
3.  using System.Linq;
4.  using System.Text;
5.  using System.Threading.Tasks;
6.
7.  namespace Triangle
8.  {
9.      class Program
10.     {
11.         static void Main(string[] args)
12.         {
13.             Console.WriteLine("Welcome to Triangle Analyzer");
14.             string inputString;
15.             long choice, firstSide, secondSide, thirdSide;
16.             string fSide, sSide, tSide;
17.             do
18.             {
19.                 Console.WriteLine("1. Enter triangle dimensions.\n2. Exit");
20.                 inputString = Console.ReadLine();
21.             } while (!(long.TryParse(inputString, out choice) && (choice == 1 ||
    choice == 2)));
22.
23.             while (choice == 1)
24.             {
25.                 do
26.                 {
27.                     Console.Write("Enter length of First Side of Triangle(1-
    2147483647): ");
28.                     fSide = Console.ReadLine();
29.                 } while (!(long.TryParse(fSide, out firstSide) && (firstSide >= 1
    && firstSide <= 2147483647)));
30.
31.                 do
32.                 {
33.                     Console.Write("Enter length of Second Side of Triangle(1-
    2147483647): ");
34.                     sSide = Console.ReadLine();
35.                 } while (!(long.TryParse(sSide, out secondSide) && (secondSide >=
    1 && secondSide <= 2147483647)));
36.
37.                 do
38.                 {
39.                     Console.Write("Enter length of Third Side of Triangle(1-
    2147483647): ");
40.                     tSide = Console.ReadLine();
41.                 } while (!(long.TryParse(tSide, out thirdSide) && (thirdSide >= 1
    && thirdSide <= 2147483647)));
42.
43.                 string output = TriangleSolver.Analyze(firstSide, secondSide,
    thirdSide);
```

```
44.
45.                 Console.WriteLine(output);
46.                 System.Threading.Thread.Sleep(1000);
47.
48.                 do
49.                 {
50.                     Console.WriteLine("1. Enter triangle dimensions.\n2. Exit");
51.                     inputString = Console.ReadLine();
52.                 } while (!(long.TryParse(inputString, out choice) && (choice == 1
    || choice == 2)));
53.             }
54.
55.             if (choice == 2)
56.             {
57.                 Console.WriteLine("BBYE!");
58.                 System.Threading.Thread.Sleep(1000);
59.                 System.Environment.Exit(1);
60.             }
61.         }
62.     }
63. }
```

#2 Source Code of TriangleSolver.cs

```
1.  using System;
2.  using System.Collections.Generic;
3.  using System.Linq;
4.  using System.Text;
5.  using System.Threading.Tasks;
6.
7.  namespace Triangle
8.  {
9.      public static class TriangleSolver
10.     {
11.         public static string Analyze(long fSide, long sSide, long tSide)
12.         {
13.             //checking if sides are valid.
14.             string message = "";
15.             if ((fSide + sSide > tSide) && (fSide + tSide > sSide) && (sSide +
    tSide > fSide))
16.             {
17.
18.                 if (fSide == sSide && sSide == tSide)
19.                 {
20.                     message = "Given numbers ARE sides of an EQUILATERAL
    Triangle.";
21.                 }
22.                 else if (fSide == sSide || fSide == tSide || sSide == tSide)
23.                 {
24.                     message = "Given numbers ARE sides of an ISOSCELES Triangle.";
25.                 }
26.                 else
27.                 {
28.                     message = "Given numbers ARE sides of an SCALENE Triangle.";
```

```
                }
            }
            else
            {
                message = "Given numbers CAN NOT be sides of a Triangle.";
            }
            return message;
        }
    }
}
```

#3 Source Code of TriangleSolverTest.cs

```
using System;
using Triangle;
using NUnit.Framework;


namespace Triangle
{
    [TestFixture]
    public class TriangleSolverTest
    {
        [TestCase, Description("Valid equilateral triangle.")]
        public void Analyze_ifThreeSidesAreEqual_expectEquilateral()
        {
            Assert.AreEqual("Given numbers ARE sides of an EQUILATERAL Triangle.",
TriangleSolver.Analyze(5, 5, 5));
        }

        [TestCase, Description("Valid isosceles triangle.")]
        public void Analyze_ifTwoSidesAreEqual_expectIsosceles()
        {
            Assert.AreEqual("Given numbers ARE sides of an ISOSCELES Triangle.",
TriangleSolver.Analyze(10, 10, 15));
        }

        [TestCase, Description("Valid scalene triangle.")]
        public void Analyze_ifNoSideEqual_expectScalene()
        {
            Assert.AreEqual("Given numbers ARE sides of an SCALENE Triangle.",
TriangleSolver.Analyze(5, 7, 10));
        }

        [TestCase, Description("Sum of two sides is less than third.")]
        public void Analyze_ifSumOfTwoSidesIsLessThanThird_expectInvalidTriangle()
        {
            Assert.AreEqual("Given numbers CAN NOT be sides of a Triangle.",
TriangleSolver.Analyze(10, 5, 1));
        }

        [TestCase, Description("All three sides have MAX permissible length and
equilateral triangle.")]
```

```
36.         public void
    Analyze_ifThreeSidesAreEqualAndHoldBoundryValues_expectEquilateral()
37.         {
38.             Assert.AreEqual("Given numbers ARE sides of an EQUILATERAL Triangle.",
    TriangleSolver.Analyze(2147483647, 2147483647, 2147483647));
39.         }
40.
41.         [TestCase, Description("Two sides at maximum possible value.")]
42.         public void
    Analyze_ifTwoSidesAreEqualAndHoldBoundryValues_expectIsosceles()
43.         {
44.             Assert.AreEqual("Given numbers ARE sides of an ISOSCELES Triangle.",
    TriangleSolver.Analyze(2147483647, 2147483647, 15));
45.         }
46.
47.         [TestCase, Description("Tow sides at MIN and one side at MAX.")]
48.         public void
    Analyze_ifTwoSidesHoldMinBoundryValuesThirdSideHoldMaxValue_expectInvalidTriangle(
    )
49.         {
50.             Assert.AreEqual("Given numbers CAN NOT be sides of a Triangle.",
    TriangleSolver.Analyze(1, 1, 2147483647));
51.         }
52.
53.         [TestCase, Description("Sum of two sides is equal to the thrid, produces
    invalid triangle.")]
54.         public void
    Analyze_ifSumOfTwoSidesIsEqualToThirdSide_expectInvalidTriangle()
55.         {
56.             Assert.AreEqual("Given numbers CAN NOT be sides of a Triangle.",
    TriangleSolver.Analyze(12, 7, 5));
57.         }
58.     }
59. }
```

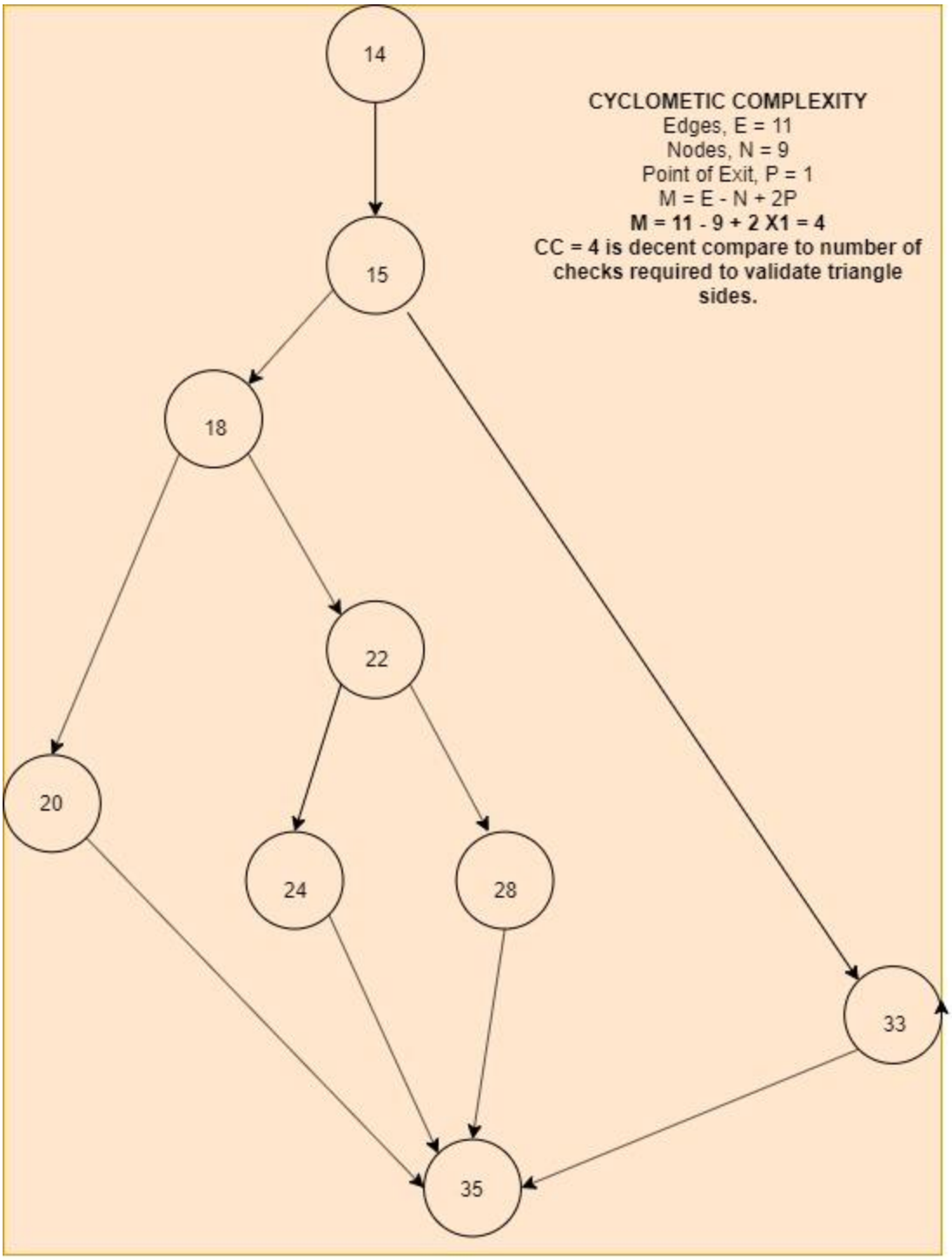
#4 CFG for Analyze Method

CFG is an important metrics being used in Static Unit Testing. Using CFG, we can track various paths that the code might follow.

Cyclometic Graphs tend to get complicated if routine gets bigger.

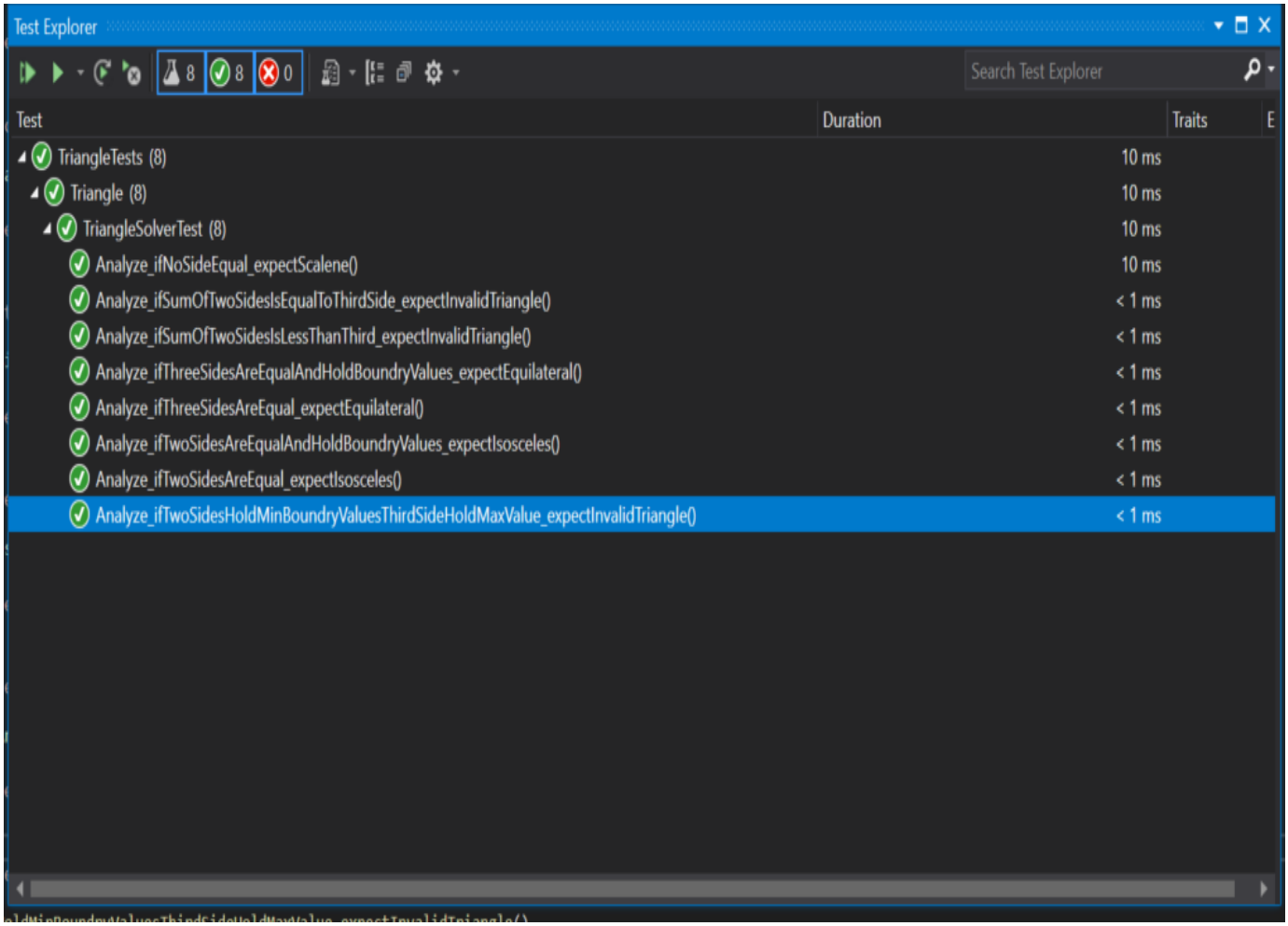
My Analyze method shows **CFG: 4, which I think is decent score which means that method which I wrote have Average Cyclometic Complexity.**

Assignment – 01: Unit Testing using NUnit



CFG DiAGRAM FOR TriangleSolver.Analyze(long x, long y, long

#5 NUnit Successful Test Runs Screenshot



#6 Git Hub Push

```
tomar@DESKTOP-3CKAGQR MINGW64 /c/Git/TriangleSolver (master)
$ git log
commit 216dc13d59799edd6c73f99e6735ad169a3ce9ef (HEAD -> master)
Merge: 08ffdad 2d3edfa
Author: ATomar2150 <56009926+ATomar2150@users.noreply.github.com>
Date:   Wed Oct 2 02:55:20 2019 -0400

    Merge branch 'master' of https://github.com/ATomar2150/TriangleSolver

commit 08ffdadf9951cb74b3d7600e959e434294a12bc7
Author: ATomar2150 <56009926+ATomar2150@users.noreply.github.com>
Date:   Wed Oct 2 02:51:22 2019 -0400

    Added code for Program.cs, TriangleSolver.cs, and TriangleSolverTest.cs"

commit 2d3edfa3d78de3ff8603d4b1f7bde182f4bb211c (origin/master, origin/HEAD)
Author: ATomar2150 <56009926+ATomar2150@users.noreply.github.com>
Date:   Wed Oct 2 02:51:22 2019 -0400

    Added code for Program.cs, TriangleSolver.cs, and TriangleSolverTest.cs"

commit 0d73c5c0aff1a4f19e115d30a5eff5cb1360f94f
Author: ATomar2150 <56009926+ATomar2150@users.noreply.github.com>
Date:   Wed Oct 2 02:33:11 2019 -0400

    Added screenshot displaying successful execution of test

commit 6c18b133bb74186735c9f2aaf75d0b7390c2f619
Author: ATomar2150 <56009926+ATomar2150@users.noreply.github.com>
Date:   Wed Oct 2 02:29:00 2019 -0400

    Update README.md

commit eacbe135f3e0f8c95b0fbece98de29cd5c9e00b1
Author: ATomar2150 <56009926+ATomar2150@users.noreply.github.com>
Date:   Mon Sep 30 22:53:36 2019 -0400

    Added assignment 02 of QA

commit 1470647f76d857c25972c33516ce755c28d63c4c
Author: ATomar2150 <56009926+ATomar2150@users.noreply.github.com>
Date:   Mon Sep 30 22:34:33 2019 -0400

    Trail Commit by Aparna

commit 5576ae8450150b0b6ca24432c8b70ee14b0cb102
Author: ATomar2150 <56009926+ATomar2150@users.noreply.github.com>
Date:   Mon Sep 30 22:12:38 2019 -0400

    Initial commit
```