PROG2070 – Quality Assurance: Winter 2017

Assignment 02

[Maximum points: 45]

Due Date: Week of Feb. 13, 2016, in Class

This assignment should be done **individually**. Do your own work and **do not share** your work with others. Sharing work is an academic offense and is subject to penalty. Be aware that source code and documents are automatically checked by eConestoga against every other student's work in the course. Academic offenses will be reported to the College Registrar.

This assignment will be demonstrated in class. Any assignment not demonstrated in class will receive a penalty as outlined in the Assignment 02 Rubric.

For this assignment you must use a form of version control. You may use Git through Git Bash as in Assignment 01, but you are free to research and use any version control software (online or local) that you wish.

You are to create a C# console application that first presents the user with the following menu:

1. **Enter triangle dimensions**
2. **Exit**

If any incorrect input is given, the menu is shown again. Your program should never crash.

The exit option quits the program, and is the only way to exit the program (other than closing the window).

If the user selects the first option, the program asks the user to enter three integers. The program will use these three numbers and report back to the user if the numbers:

A. Form a triangle or not
B. If the numbers do form a triangle, the program reports if that triangle is scalene, isosceles, or equilateral.

You must create a public static class called TriangleSolver as a separate file. This class contains a static method called Analyze which accepts three integers as an input, and returns a string as an output.

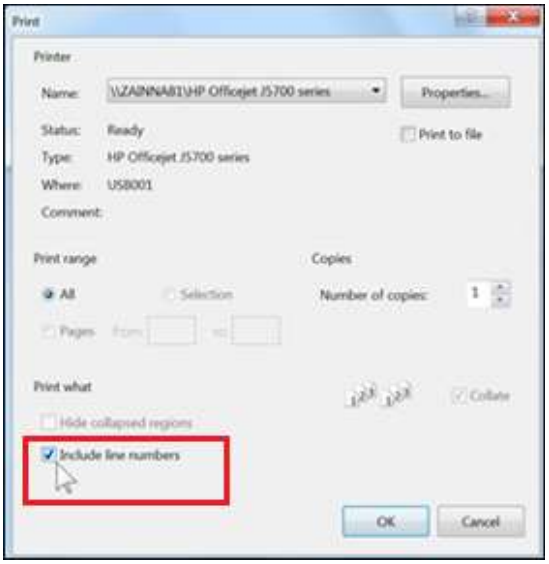
Write eight unit tests for the Analyze method. **Instructions for installing NUnit remain on eConestoga in the Assignment01 subfolder.** Each unit test should test some different condition or boundary. Provide a briefly (one or two sentences) explaining why you selected each input for your unit cases. Provide a screenshot of the Unit cases being run (can take a screenshot of the NUnit GUI after a run) [1 Mark]. All of your unit tests should pass.

Your test cases should be in their own project, but in the same solution as your console program, as you did in Assignment 01. Remember the naming conventions for the Test Project, the Test Class, and the Test Methods.

Draw Control Flow Graph for your Analyze method. The CFG should be drawn in an appropriate modeling software (such as Microsoft Visio). Label all of your nodes with the line numbers that match your source code.

Finally, compute the Cyclomatic Complexity of your CFG, showing your work. Provide a brief description of what the Cyclomatic Complexity number of your CFG tells you about your solution.

When printing your source code, please include the line numbers. This can be done in Visual Studio by checking the "Include line numbers" check box on the print prompt window, as shown in Figure 1 below.



Name: \\ZAINNAB1\HP Officejet J5700 series

Status: Ready

Type: HP Officejet J5700 series

Where: USB001

Comment:

□ Print to file

● All ○ Selection

□ Pages From: to:

Number of copies: 1

☑ Collate

□ Hide collapsed regions

☑ Include line numbers

Figure 1. How to Include Line Numbers on Print Out

The format for submitting the assignment is as follows:

1. **Demonstrate your program in class.**
2. **Printouts Handed in Class**:
    a. Assignment Cover sheet properly filled (found on eConestoga)
    b. Assignment Rubric left blank (found on eConestoga)
    c. Copy of Program source code
    d. Copy of TriangleSolver class source code
    e. Copy of Unit Test class source code
    f. Print out of your CFG for the Analyze method
    g. Print out of Doc showing the results of your unit tests being run, with an explanation to why you chose each case. This doc should also include your CFG, and a Cyclomatic Complexity calculation for it. A brief discussion of the CC value should be included. Finally a screenshot/output of your git repository log or any output from your version control showing commits.
3. **eConestoga Submission**: A single compressed (.zip format) archive file containing the solution folder of your source code (so I can run it) and the doc file with the screenshots, calculations and explanations.