Corey Heckel

Dr. Eman AlOmar

SSW 567

27 September 2023

Assignment Description:

Sometimes you will be given a program that someone else has written, and you will be asked to fix, update and enhance that program. In this assignment you will start with an existing implementation of the classify triangle program that will be given to you. You will also be given a starter test program that tests the classify triangle program, but those tests are not complete.

- These are the two files: Triangle.py and TestTriangle.py
  - ***Triangle.py*** is a starter implementation of the triangle classification program.
  - ***TestTriangle.py*** **c**ontains a starter set of unittest test cases to test the classifyTriangle() function in the file Triangle.py file.

In order to determine if the program is correctly implemented, you will need to update the set of test cases in the test program. You will need to update the test program until you feel that your tests adequately test all of the conditions. Then you should run the complete set of tests against the original triangle program to see how correct the triangle program is. Capture and then report on those results in a formal test report described below. For this first part you should not make any changes to the classify triangle program. You should only change the test program.

Based on the results of your initial tests, you will then update the classify triangle program to fix all defects. Continue to run the test cases as you fix defects until all of the defects have been fixed. Run one final execution of the test program and capture and then report on those results in a formal test report described below.

Corey Heckel

Summary:

I found a variety of bugs with my test cases before and after trying to fix issues. Performing unit testing has allowed me to find my bugs more easily.

Summary Matrix

	Test Run 1	Test Run 2	Test Run 3
Tests Planned	12	12	12
Tests Executed	12	12	12
Tests Passed	3	10	12

Defects Found	9	2	0
Defects Fixed	7	2	0

I pledge my honor that I have abided by the Stevens Honor System.

Detailed Results:

Testing Program before fixing bugs:

Test ID	Input	Expected Results	Actual Result	Pass Or Fail
testRightTriangleA	3 4 5	Right	InvalidInput	Fail
testRightTriangleB	5 3 4	Right	InvalidInput	Fail
testEquilateralTriangleA	1 1 1	Equilateral	InvalidInput	Fail
testEquilateralTriangleB	3 3 3	Equilateral	InvalidInput	Fail
testScaleneTriangleA	2 3 4	Scalene	InvalidInput	Fail
testScaleneTriangleB	5 12 9	Scalene	InvalidInput	Fail
testIsocelesTriangleA	5 5 7	Isoceles	InvalidInput	Fail
testIsocelesTriangleB	2 3 2	Isoceles	InvalidInput	Fail
testInvalidInputA	200 200 200	InvalidInput	InvalidInput	Pass
testInvalidInputB	0 0 0	InvalidInput	InvalidInput	Pass
testInvalidInputC	1 2 1.2	InvalidInput	InvalidInput	Pass
testNotTriangle	1 2 3	NotATriangle	InvalidInput	Fail

After Fixing Bugs:

```
PS C:\Users\wheel\Downloads\SSW-567\HW02a> python -m unittest TestTriangle
............
----------------------------------------------------------------------
Ran 12 tests in 0.004s

OK
PS C:\Users\wheel\Downloads\SSW-567\HW02a>
```

Test ID	Input	Expected Results	Actual Result	Pass Or Fail
testRightTriangleA	3 4 5	Right	Right	Pass
testRightTriangleB	5 3 4	Right	Right	Pass
testEquilateralTriangleA	1 1 1	Equilateral	Equilateral	Pass
testEquilateralTriangleB	3 3 3	Equilateral	Equilateral	Pass
testScaleneTriangleA	2 3 4	Scalene	Scalene	Pass
testScaleneTriangleB	5 12 9	Scalene	Scalene	Pass
testIsocelesTriangleA	5 5 7	Isoceles	Isoceles	Pass
testIsocelesTriangleB	2 3 2	Isoceles	Isoceles	Pass
testInvalidInputA	200 200 200	InvalidInput	InvalidInput	Pass
testInvalidInputB	0 0 0	InvalidInput	InvalidInput	Pass
testInvalidInputC	1 2 1.2	InvalidInput	InvalidInput	Pass
testNotTriangle	1 2 3	NotATriangle	NotATrigangle	Pass

The Github repo can be found here:

https://github.com/coreyheckel3/SSW-567