SSW 567 HW 2

1. **Assignment Description:**The objective of this assignment is to apply the techniques from the lecture to static testing of your Triangles program. Specifically:  
     
   You will run a static code analyzer on your code, e.g. Pylint, identify and fix any problems reported by the static code analyzer;  
     
   You will run a code coverage tool on your code, e.g. Coverage.py, and extend your test cases to demonstrate at least 80% code coverage;  
     
   In this assignment, you will need to download and install the tools that you will need for static code analysis and code coverage. You will then run those tools locally on your laptop to get the results.   
     
   Any changes that you make to your programs should be pushed up to GitHub.
2. **Author:** Adrian Garcia
3. **Summary:** 
   1. GitHub URL: https://github.com/adriang11/SSW-567
   2. Pylint:  
      Text

      Description automatically generated
   3. Coverage:  
      Text

      Description automatically generated
   4. No new test cases required. (100% coverage from original state).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Input** | **Expected Results** | **Actual Result** | **Pass or Fail** |
| testRightTriangleA | 3,4,5 | Right | InvalidInput | Pass |
| testRightTriangleB | 5,3,4 | Right | InvalidInput | Pass |
| testRightTriangleC | 7,24,25 | Right | InvalidInput | Pass |
| testRightTriangleD | 25,7,24 | Right | InvalidInput | Pass |
| testEquilateralTrianglesA | 1,1,1 | Equilateral | InvalidInput | Pass |
| testEquilateralTrianglesB | 8,8,8 | Equilateral | InvalidInput | Pass |
| testScaleneTrianglesA | 8,3,5 | Scalene | InvalidInput | Pass |
| testScaleneTrianglesB | 7,3,7 | Scalene | InvalidInput | Pass |
| testIsoscelesTrianglesA | 8,8,3 | Isosceles | InvalidInput | Pass |
| testIsoscelesTrianglesB | 6,4,6 | Isosceles | InvalidInput | Pass |
| testInvalidInputA | 500,10,50 | InvalidInput | InvalidInput | Pass |
| testInvalidInputB | 6,4.75,6 | InvalidInput | InvalidInput | Pass |
| testNotATriangleA | 80,4,6 | NotATriangle | InvalidInput | Pass |
| testNotATriangleB | 3,98,6 | NotATriangle | InvalidInput | Pass |

Defined Test Cases:

* 1. Final Pylint output:  
     **Text

     Description automatically generated**

1. **Reflection:**Overall, it seems it was worth it to thoroughly test my triangle.py in my original file as test coverage was not an issue for this assignment. Pylint proved to be a very useful tool to clean up my code and make it more legible and less cluttered. For my final output, I was able to achieve a rating 10/10 by removing trailing whitespaces, combing functions with the same return statements and optimizing line length among other things.
2. **Honor pledge:**  
   I pledge my honor that I have abided by the Stevens Honor System – Adrian Garcia
3. **Detailed results, if any:**Progression of fixing pylint errors:
4. Text

   Description automatically generated
5. Text

   Description automatically generated
6. Text

   Description automatically generated
7. Text

   Description automatically generated