1. 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.

1. Sean Hodges
2. Summary:

This homework assignment was pretty cool to see how to use coverage.py. I think that is way more useful than pylint because it shows you how much testing you have done. When I made the test cases for this program initially I thought I had considered every possible case but coverage showed me that I missed 2.

* 1. https://github.com/austinh48/SSW567/tree/main/Homework5
  2. Text

     Description automatically generatedText

     Description automatically generatedpylint.py
  3. Coverage.py

A picture containing graphical user interface

Description automatically generated

* 1. Original test cases had 94% coverage

Text

Description automatically generated

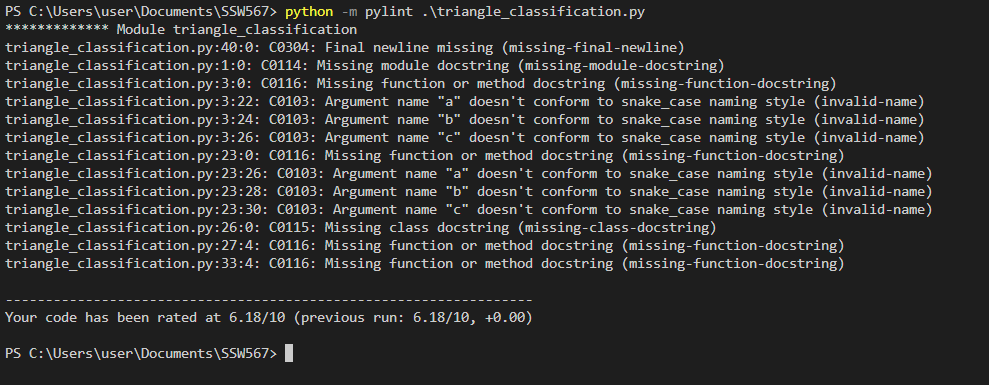
* 1. Output

Graphical user interface, text

Description automatically generated

A picture containing graphical user interface

Description automatically generated



A picture containing graphical user interface

Description automatically generated

I didn’t make any changes to the test cases since 94% coverage is pretty good.

1. Reflection:

I think that this was a cool assignment, I don’t know how useful pylint is, although I guess it is good for refactoring code and checking to make sure that there aren’t variables that have been created but not used although I think VS code already has that feature built in. However, I think coverage.py is super helpful for checking to see how much of your program you’ve testing with test cases.