Name: Hemanth Boinipally

Stevens Institute of Technology

HW05: Static Code Analysis

Course: SSW–567

Summary:

The code was changed to make it more legible and executable by eliminating any superfluous spaces, indentation, and variable renaming after the code analyzer was performed on the original program. Once this was finished, the static value was full. Consequently, 92% of the population was covered.

1. The GitHub URL of this code which is analyzed is:
   https://github.com/hboinippa/SSW567HW05.git

2. The name and output of the static code analyzer tool you used:
   The tool used for static code analyzer is **Pylint** and **Coverage**

Initial Output (Before making the changes to code)

```
************* Module Triangle
Triangle.py:7:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:11:0: C0301: Line too long (108/100) (line-too-long)
Triangle.py:18:0: C0304: Final newline missing (missing-final-newline)
Triangle.py:1:0: C0114: Missing module docstring (missing-module-docstring)
Triangle.py:1:0: C0103: Module name "Triangle" doesn't conform to snake_case naming style (in
Triangle.py:1:21: C0103: Argument name "a" doesn't conform to snake_case naming style (invali
d-name)
Triangle.py:1:24: C0103: Argument name "b" doesn't conform to snake_case naming style (invali
d-name)
Triangle.py:1:27: C0103: Argument name "c" doesn't conform to snake_case naming style (invali
d-name)
Triangle.py:8:4: R1705: Unnecessary "elif" after "return", remove the leading "el" from "elif
" (no-else-return)

-----------------------------------
Your code has been rated at 0.83/10
```

```
PS C:\Users\15513\Documents\GitHub\SSW567HW05> & C:/Users/15513/AppData/Local/Programs/Python/Python310/python.exe c:/Users/15513/Documents/GitHub/SSW567HW05/triangletest.py
....
----------------------------------------------------------------------
Ran 4 tests in 0.001s

OK
PS C:\Users\15513\Documents\GitHub\SSW567HW05> pylint triangletest.py
************* Module triangletest
triangletest.py:18:0: C0303: Trailing whitespace (trailing-whitespace)
triangletest.py:25:0: C0304: Final newline missing (missing-final-newline)
triangletest.py:1:0: C0114: Missing module docstring (missing-module-docstring)
triangletest.py:5:0: C0115: Missing class docstring (missing-class-docstring)
triangletest.py:7:4: C0116: Missing function or method docstring (missing-function-docstring)
triangletest.py:11:4: C0116: Missing function or method docstring (missing-function-docstring)
triangletest.py:15:4: C0116: Missing function or method docstring (missing-function-docstring)
triangletest.py:19:4: C0116: Missing function or method docstring (missing-function-docstring)

-----------------------------------
Your code has been rated at 3.85/10

PS C:\Users\15513\Documents\GitHub\SSW567HW05> pip3 install coverage
```

The name and output of the code coverage tool you used:

➔ The tool used is coverage.py Initial: The initial coverage was 8% for the actual code. For the command

```
PS C:\Users\15513\Documents\GitHub\SSW567HW05> & C:/Users/15513/AppData/Local/Programs/Python/Python310/python.exe c:/Users/15513/Documents/GitHub/SSW567HW05/Triangle.py
PS C:\Users\15513\Documents\GitHub\SSW567HW05> coverage run Triangle.py
PS C:\Users\15513\Documents\GitHub\SSW567HW05> coverage report -m
Name          Stmts   Miss  Cover   Missing
-----------------------------------------
Triangle.py      12     11     8%   2-18
-----------------------------------------
TOTAL            12     11     8%
PS C:\Users\15513\Documents\GitHub\SSW567HW05>
```

Final: The final coverage is 92%, covering all the test cases. For the command

```
PS C:\Users\15513\Documents\GitHub\SSW567HW05> coverage report -m
Name              Stmts   Miss  Cover   Missing
-------------------------------------------------
Triangle.py          12      2    83%   3, 6
triangletest.py      13      0   100%
-------------------------------------------------
TOTAL                25      2    92%
```

3. Identify both your original test cases and new test cases that you created to achieve at least 80% code coverage.

Ans: Our first request stated that we should make the code 100% efficient, therefore we fixed our code to be more efficient than 80%. After testing the new code against the test cases, we were able to achieve coverage of more than 92%. A 92% efficiency was attained. There was no need to develop additional test cases after I thoroughly tested the program with the Assignment's several tests' cases. Making the necessary code corrections and then posting that everything was functioning properly was what I found to be effective.

4.Attach screen shots of the output of the static code analyzer as well as code coverage. You should show a screenshot of the analysis results both before and after any changes that you make to your programs:

Ans: The before-and-after screenshots of the static code analysis and code coverage are sent as an attachment. Additionally, I used the git URL to publish this assignment to the new repository specified above.