HW 05 - Static Code Analysis

By Vaibhav Vashisht

Summary : The changes made to the original program after running the code analyzer and adding one more function to the program made the code more readable and executable eradicating all the unnecessary indentation , spaces , variable renaming. Once this was done static value reached to a full. Hence ensuring in covering 99% coverage.

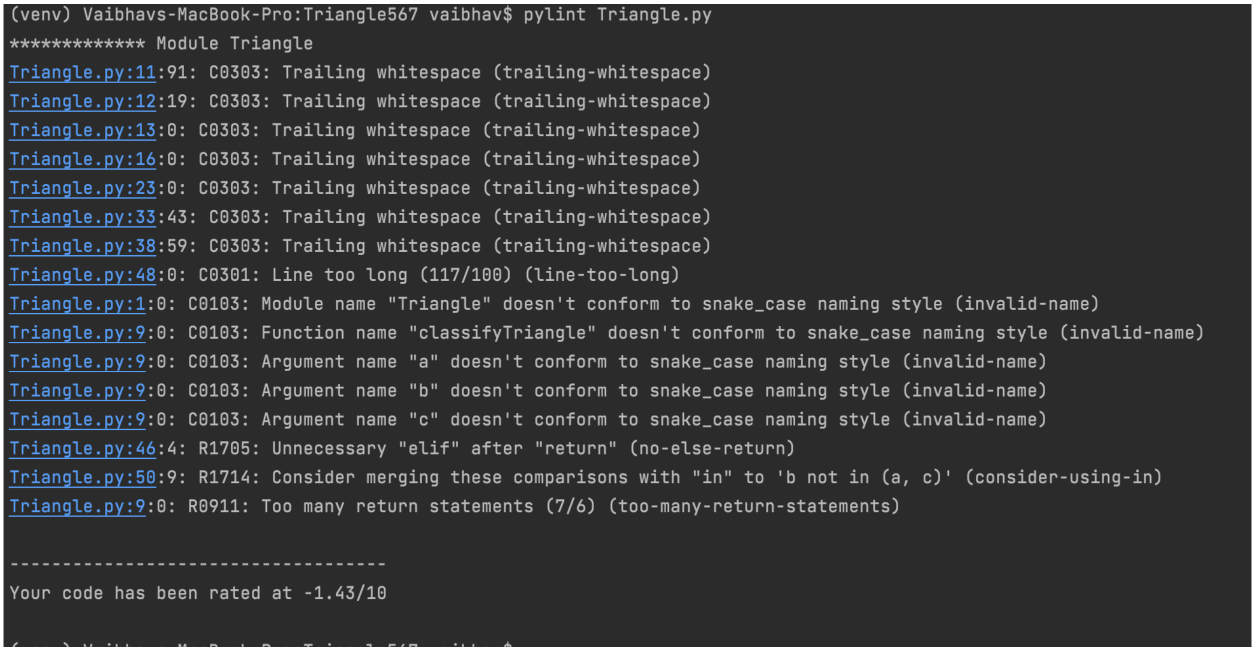
1.**The GitHub URL containing the code that was analyzed**

URL: https://github.com/vaibhavvashisht16/HW-05---Static-Code-Analysis/tree/main

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

The tool used for static code analyzer is Pylint

Initial Output (Before making the changes to the code)

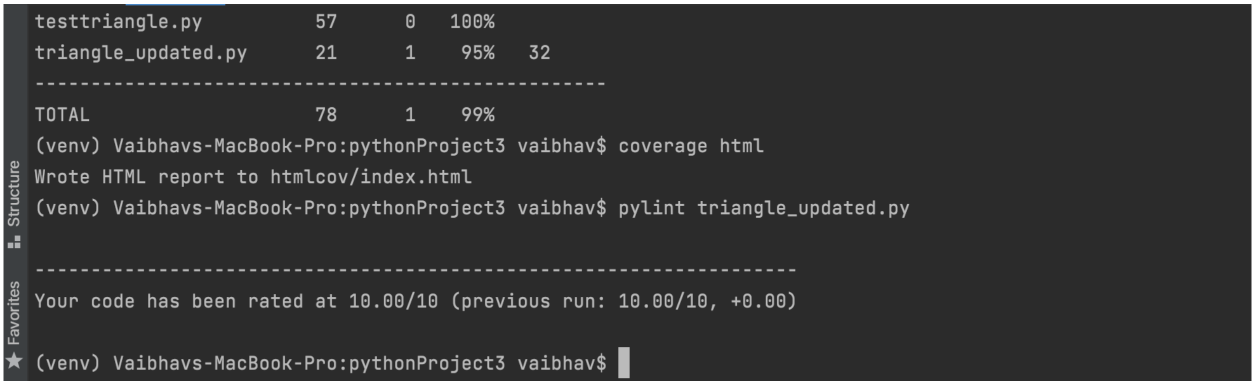


```
(venv) Vaibhavs-MacBook-Pro:Triangle567 vaibhav$ pylint Triangle.py
************* Module Triangle
Triangle.py:11:91: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:12:19: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:13:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:16:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:23:0: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:33:43: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:38:59: C0303: Trailing whitespace (trailing-whitespace)
Triangle.py:48:0: C0301: Line too long (117/100) (line-too-long)
Triangle.py:1:0: C0103: Module name "Triangle" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:9:0: C0103: Function name "classifyTriangle" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:9:0: C0103: Argument name "a" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:9:0: C0103: Argument name "b" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:9:0: C0103: Argument name "c" doesn't conform to snake_case naming style (invalid-name)
Triangle.py:46:4: R1705: Unnecessary "elif" after "return" (no-else-return)
Triangle.py:50:9: R1714: Consider merging these comparisons with "in" to 'b not in (a, c)' (consider-using-in)
Triangle.py:9:0: R0911: Too many return statements (7/6) (too-many-return-statements)

------------------------------------
Your code has been rated at -1.43/10

(venv) Vaibhavs-MacBook-Pro:Triangle567 vaibhav$
```

Final output : After the changes have been made.



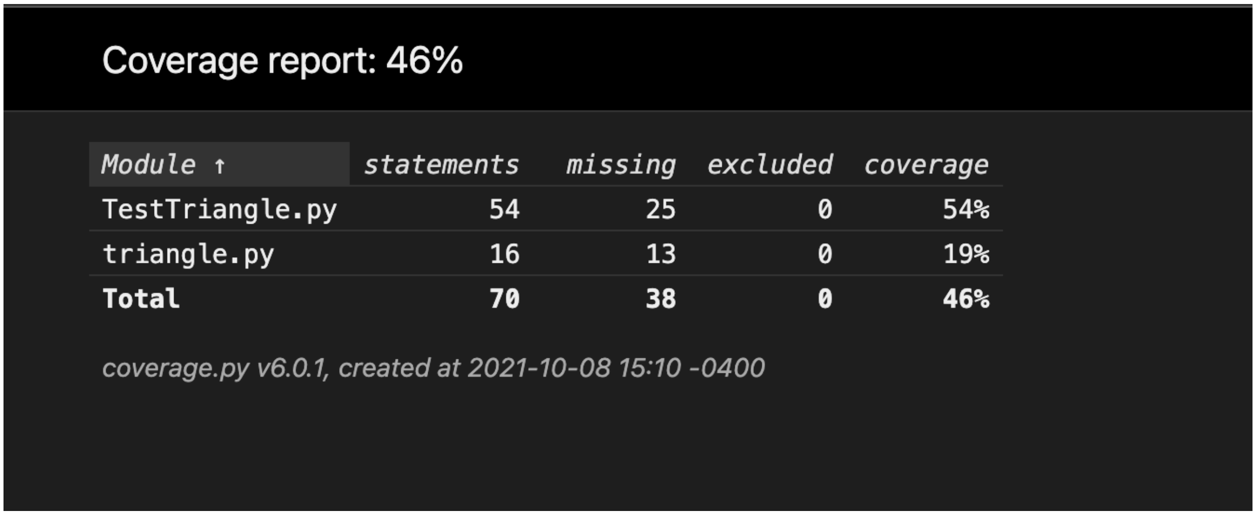
```
testtriangle.py          57      0   100%
triangle_updated.py      21      1    95%   32
---------------------------------------------------
TOTAL                    78      1    99%
(venv) Vaibhavs-MacBook-Pro:pythonProject3 vaibhav$ coverage html
Wrote HTML report to htmlcov/index.html
(venv) Vaibhavs-MacBook-Pro:pythonProject3 vaibhav$ pylint triangle_updated.py

------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

(venv) Vaibhavs-MacBook-Pro:pythonProject3 vaibhav$
```

3. The name and output of the code coverage tool you used : The tool used is coverage.py

Initial : The initial coverage was 46%.

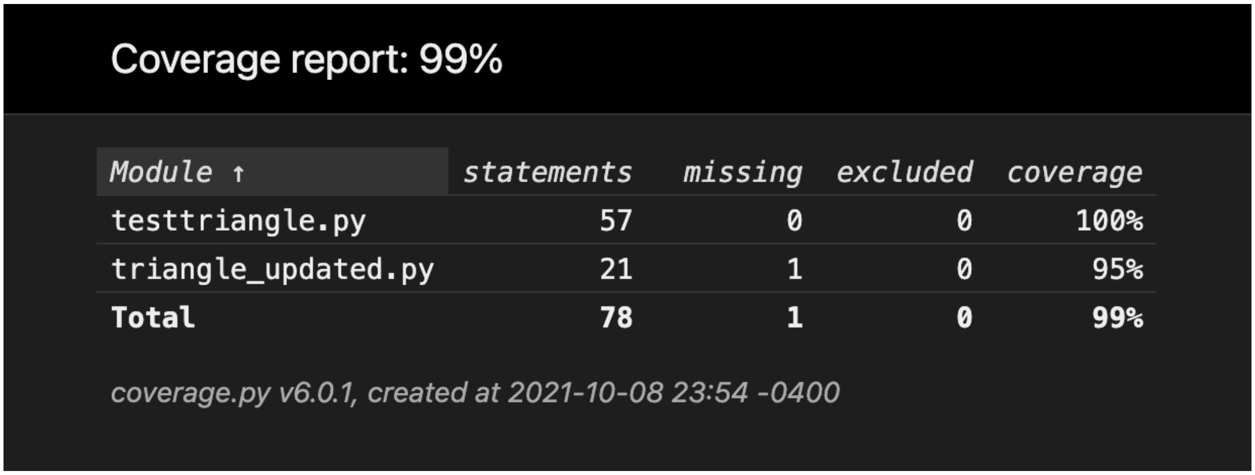


Coverage report: 46%

Module ↑	statements	missing	excluded	coverage
TestTriangle.py	54	25	0	54%
triangle.py	16	13	0	19%
Total	**70**	**38**	**0**	**46%**

coverage.py v6.0.1, created at 2021-10-08 15:10 -0400

Final : The final coverage is 99%, covering all the test cases.



Coverage report: 99%

Module ↑	*statements*	*missing*	*excluded*	*coverage*
testtriangle.py	57	0	0	100%
triangle_updated.py	21	1	0	95%
Total	**78**	**1**	**0**	**99%**

coverage.py v6.0.1, created at 2021-10-08 23:54 -0400

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

As a part of the initial request our aim was to make the code 100% , we fixed our code to an 100% efficiency and post that when we ran the test cases against the new code and was able to achieve coverage of more than 80%. We achieved an efficiency of 99%.
I tested the program with a lot of test cases in the Assignment and there was no need to add more test cases. The thing that worked for me was to make the correction to code and post that everything was working good.

5. 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:

I have attached the screenshot of the static code analysis and code coverage above before and after already above.Also uploaded the new versions of the code to git URL.