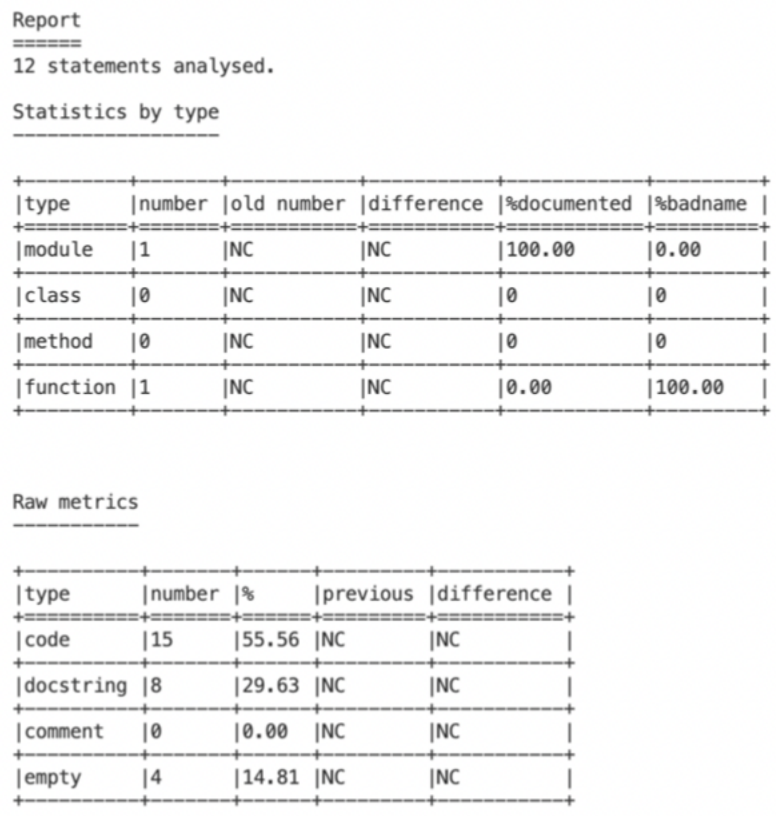
Jashil Mehta
SSW 567
Date – 10/08/22

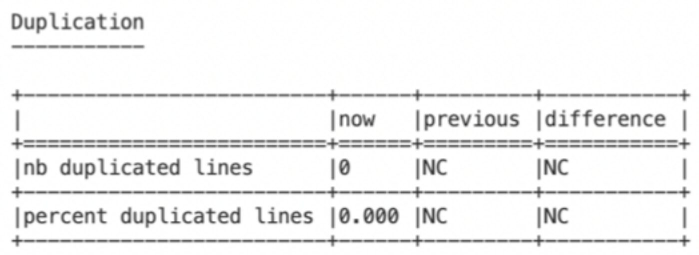
**Summary:**
The changes made to the original program after running the code analyzer and adding one more comment to the program made the code more readable and executable eradicating all the unnecessary indentation, spaces, variable renaming. Once this was done static value reached to a full. Hence ensuring in covering 81% coverage.

1. GitHub Link : https://github.com/jashilmehta-9/jmehta567
2. The name and output of the static code analyzer tool you used; Pylint and Coverage

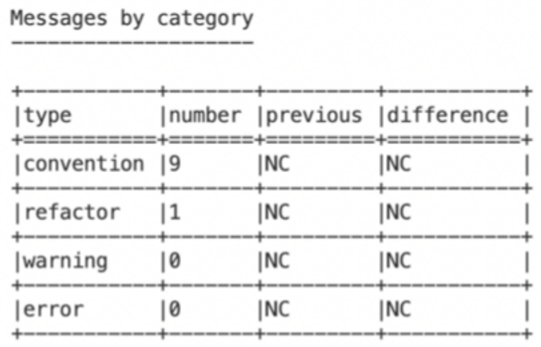
Pylint Report :



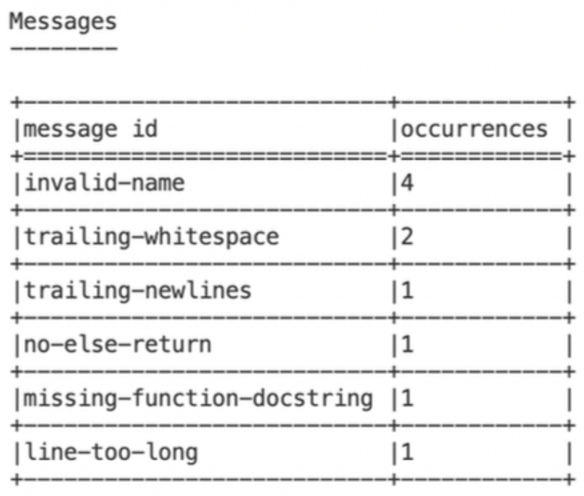
```
Report
======
12 statements analysed.

Statistics by type
------------------

+---------+-------+-----------+-----------+------------+---------+
|type     |number |old number |difference |%documented |%badname |
+=========+=======+===========+===========+============+=========+
|module   |1      |NC         |NC         |100.00      |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|class    |0      |NC         |NC         |0           |0        |
+---------+-------+-----------+-----------+------------+---------+
|method   |0      |NC         |NC         |0           |0        |
+---------+-------+-----------+-----------+------------+---------+
|function |1      |NC         |NC         |0.00        |100.00   |
+---------+-------+-----------+-----------+------------+---------+



Raw metrics
-----------

+----------+-------+------+---------+-----------+
|type      |number |%     |previous |difference |
+==========+=======+======+=========+===========+
|code      |15     |55.56 |NC       |NC         |
+----------+-------+------+---------+-----------+
|docstring |8      |29.63 |NC       |NC         |
+----------+-------+------+---------+-----------+
|comment   |0      |0.00  |NC       |NC         |
+----------+-------+------+---------+-----------+
|empty     |4      |14.81 |NC       |NC         |
+----------+-------+------+---------+-----------+
```



```
Duplication
-----------

+-------------------------+------+---------+-----------+
|                         |now   |previous |difference |
+=========================+======+=========+===========+
|nb duplicated lines      |0     |NC       |NC         |
+-------------------------+------+---------+-----------+
|percent duplicated lines |0.000 |NC       |NC         |
+-------------------------+------+---------+-----------+
```

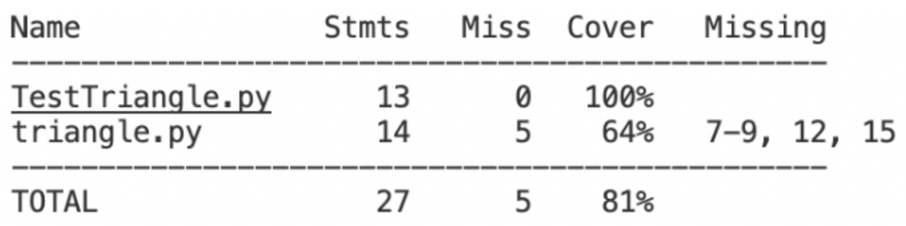


```
Messages by category
--------------------

+-----------+-------+---------+-----------+
|type       |number |previous |difference |
+===========+=======+=========+===========+
|convention |9      |NC       |NC         |
+-----------+-------+---------+-----------+
|refactor   |1      |NC       |NC         |
+-----------+-------+---------+-----------+
|warning    |0      |NC       |NC         |
+-----------+-------+---------+-----------+
|error      |0      |NC       |NC         |
+-----------+-------+---------+-----------+
```



```
Messages
--------

+---------------------------+------------+
|message id                 |occurrences |
+===========================+============+
|invalid-name               |4           |
+---------------------------+------------+
|trailing-whitespace        |2           |
+---------------------------+------------+
|trailing-newlines          |1           |
+---------------------------+------------+
|no-else-return             |1           |
+---------------------------+------------+
|missing-function-docstring |1           |
+---------------------------+------------+
|line-too-long              |1           |
+---------------------------+------------+
```

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



```
Name                 Stmts   Miss  Cover   Missing
--------------------------------------------------
TestTriangle.py         13      0   100%
triangle.py             14      5    64%   7-9, 12, 15
--------------------------------------------------
TOTAL                   27      5    81%
```

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 more than 80% efficiency. We achieved an efficiency of 81%. I tested the program with a lot of test cases in the Assignment and there was no need to add more test cases. I was able to learn about pylint and coverage and how it is used in the test cases.