Jason Meyerberg

HW5

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.

2. **Author**: Jason Meyerberg

3. **Summary**:

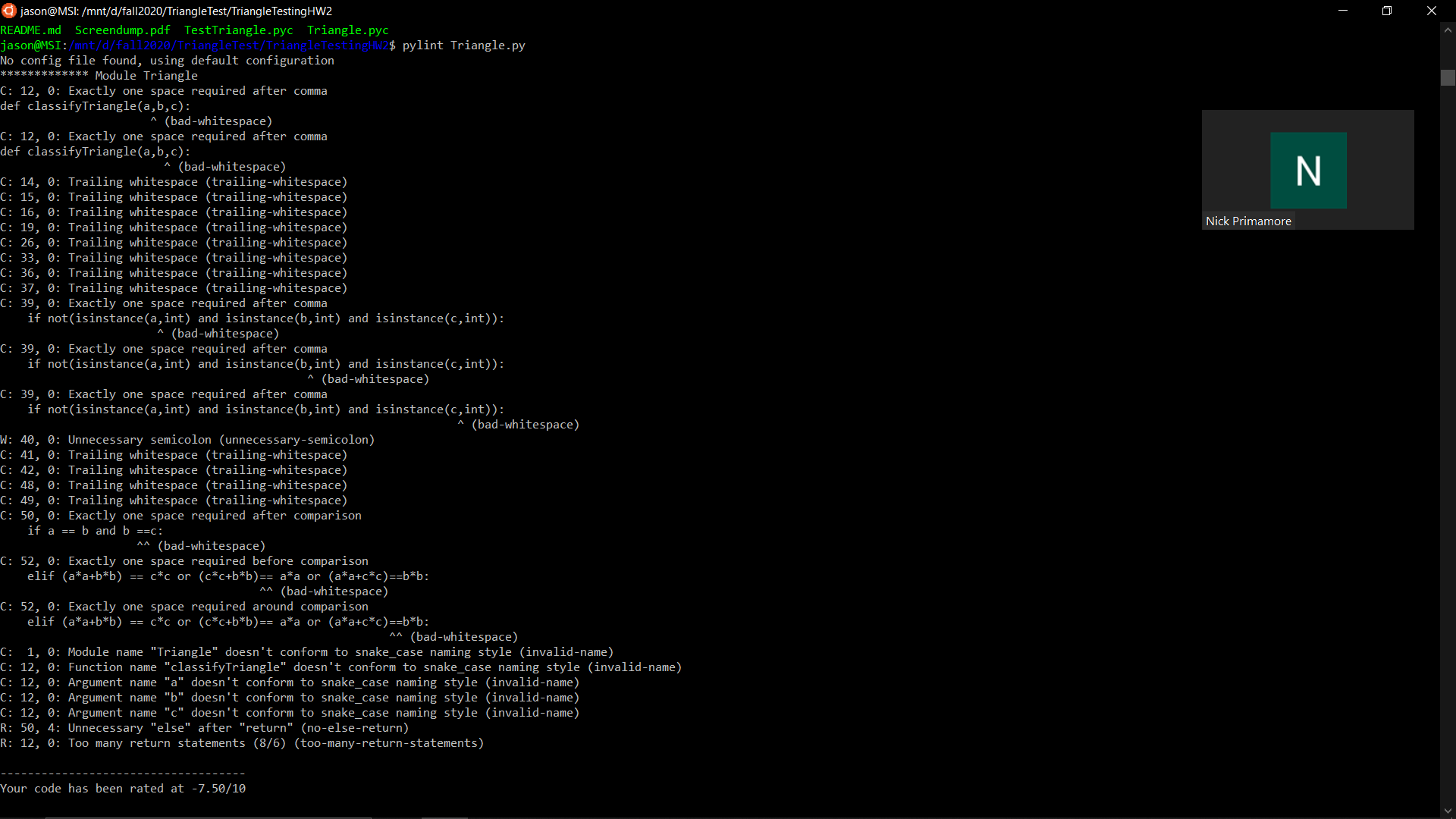
This assignment involved fixing the style and checking if the coverage was over 80%. I had a lot of style errors that took a while to fix which was a little tedious. Eventually I was able to get a 10 and very close to a 10 on both files. My testing coverage was already at 99% so my test cases did not need to change, which was a relief. A lot of the style mistakes were the same so once I knew how to fix it, it was just the process of doing it repeatedly.

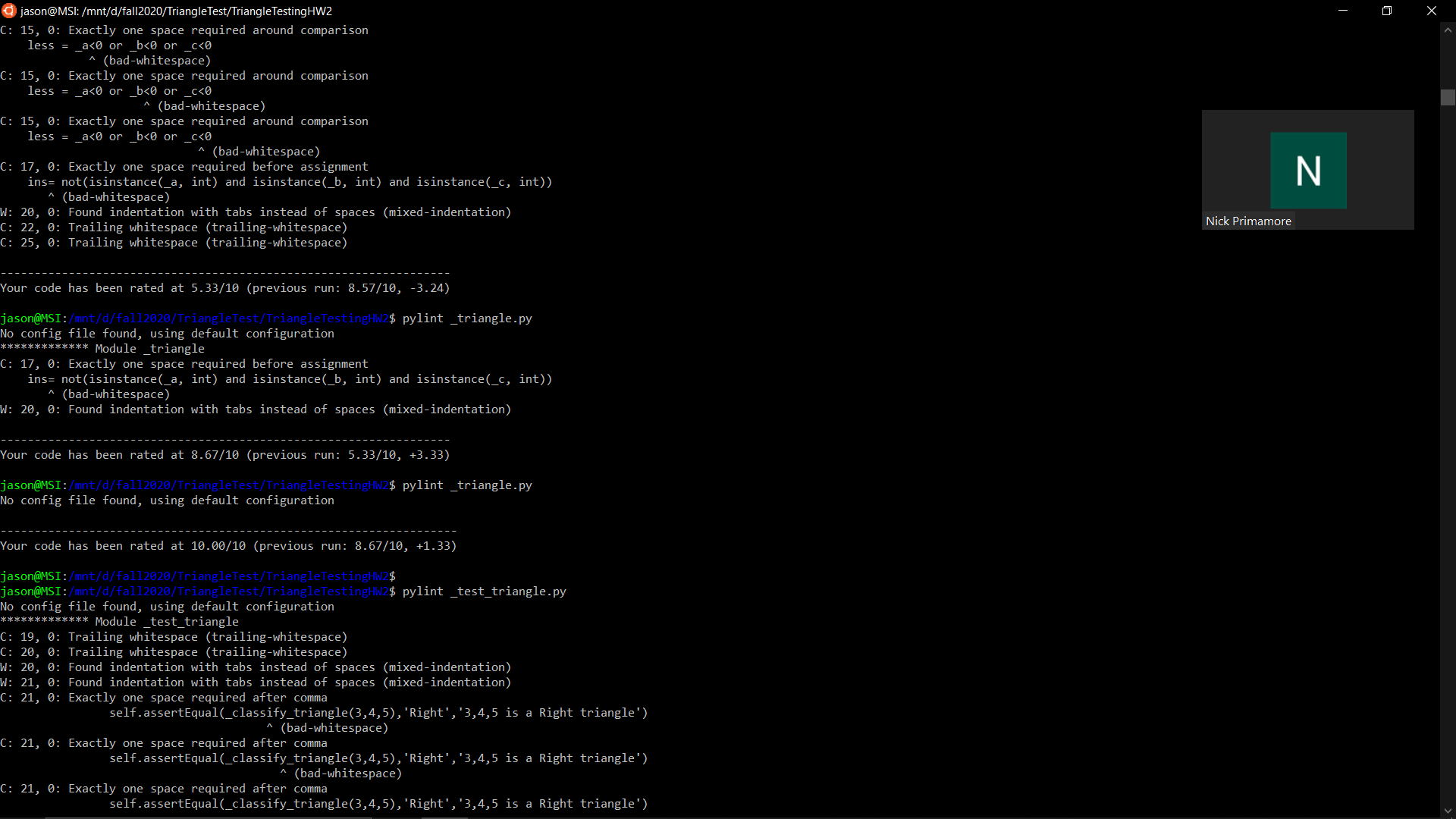
Reflection:

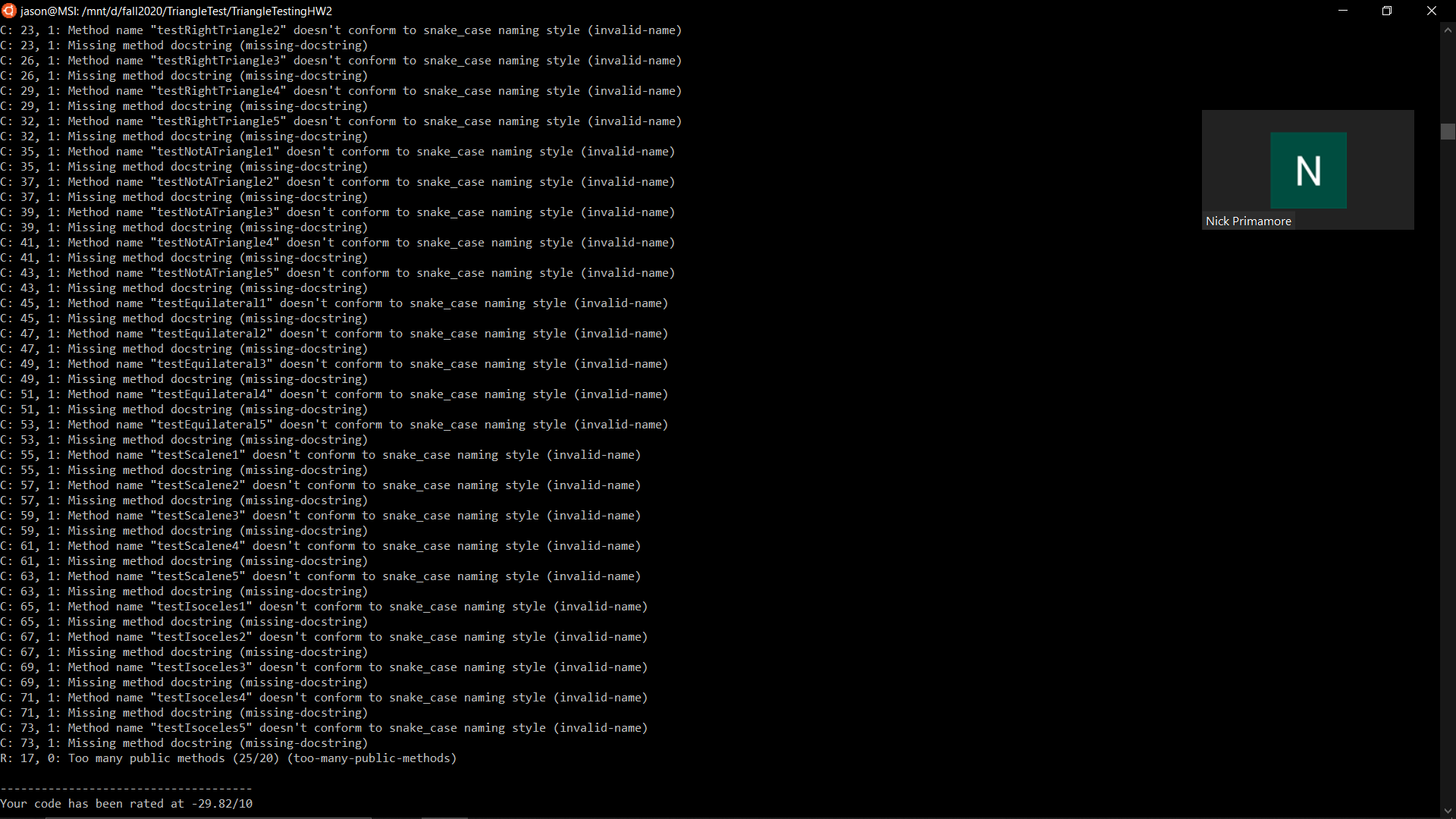
Although this may help people with reading the code and style points. I did not find it necessary for my personal coding. I do not normally code like this so to go back and fix every mistake was a lot of work. Especially if the code gets long. The functionality is all the same and it just created a lot of extra work. The coverage is good to make sure you are testing properly.

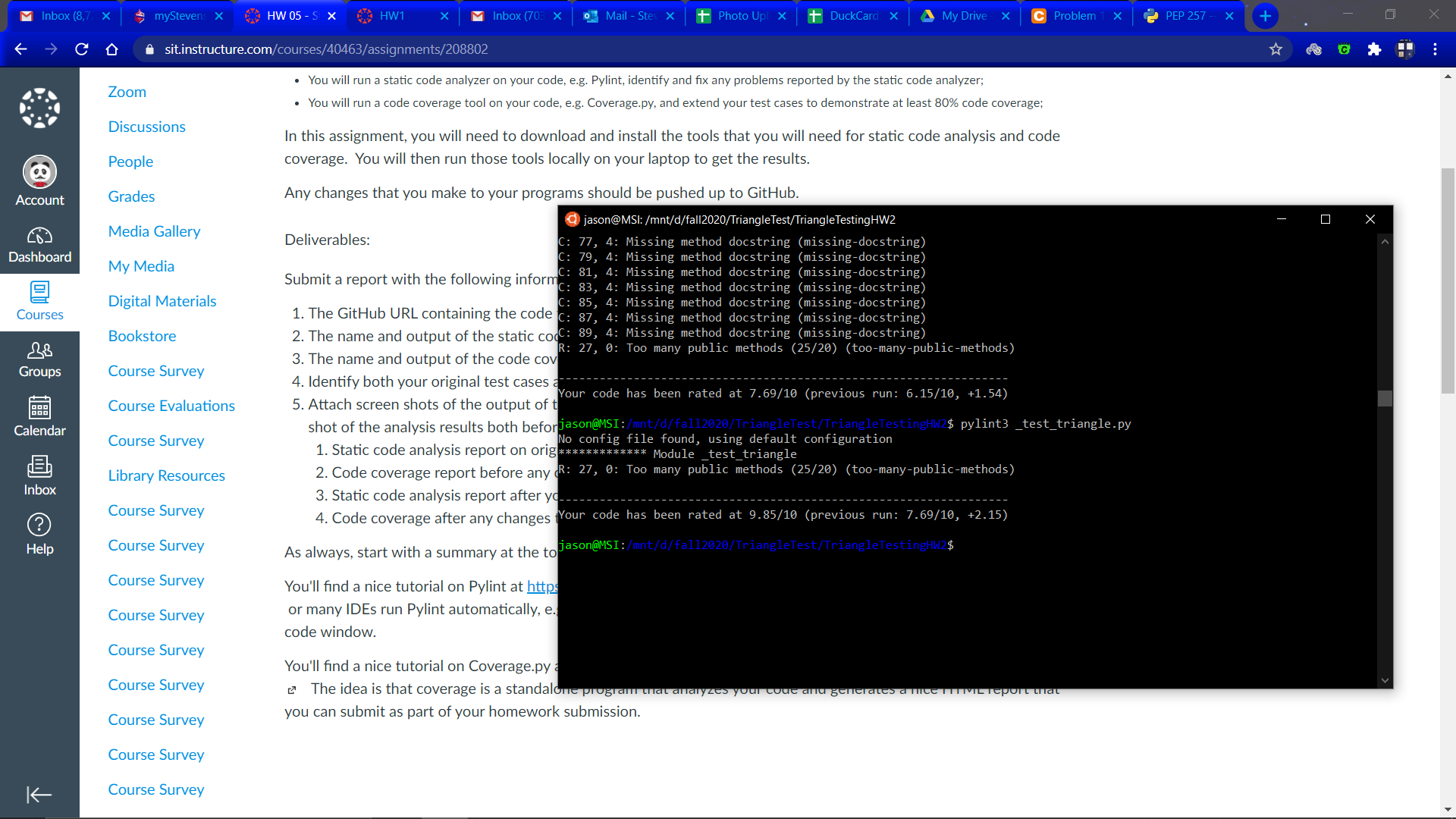
**Detailed Results:**

Pylint Static Code Analyzer

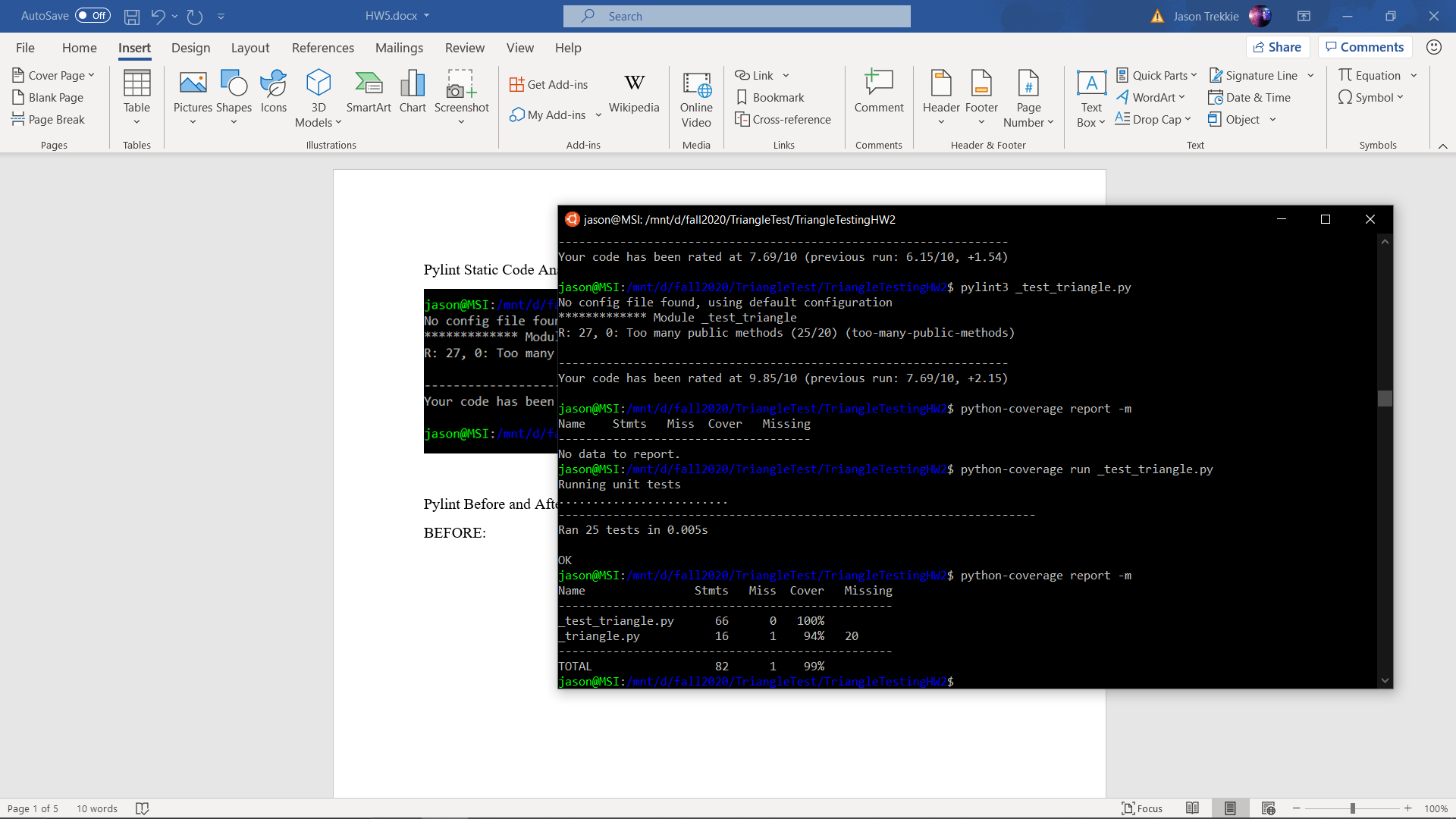








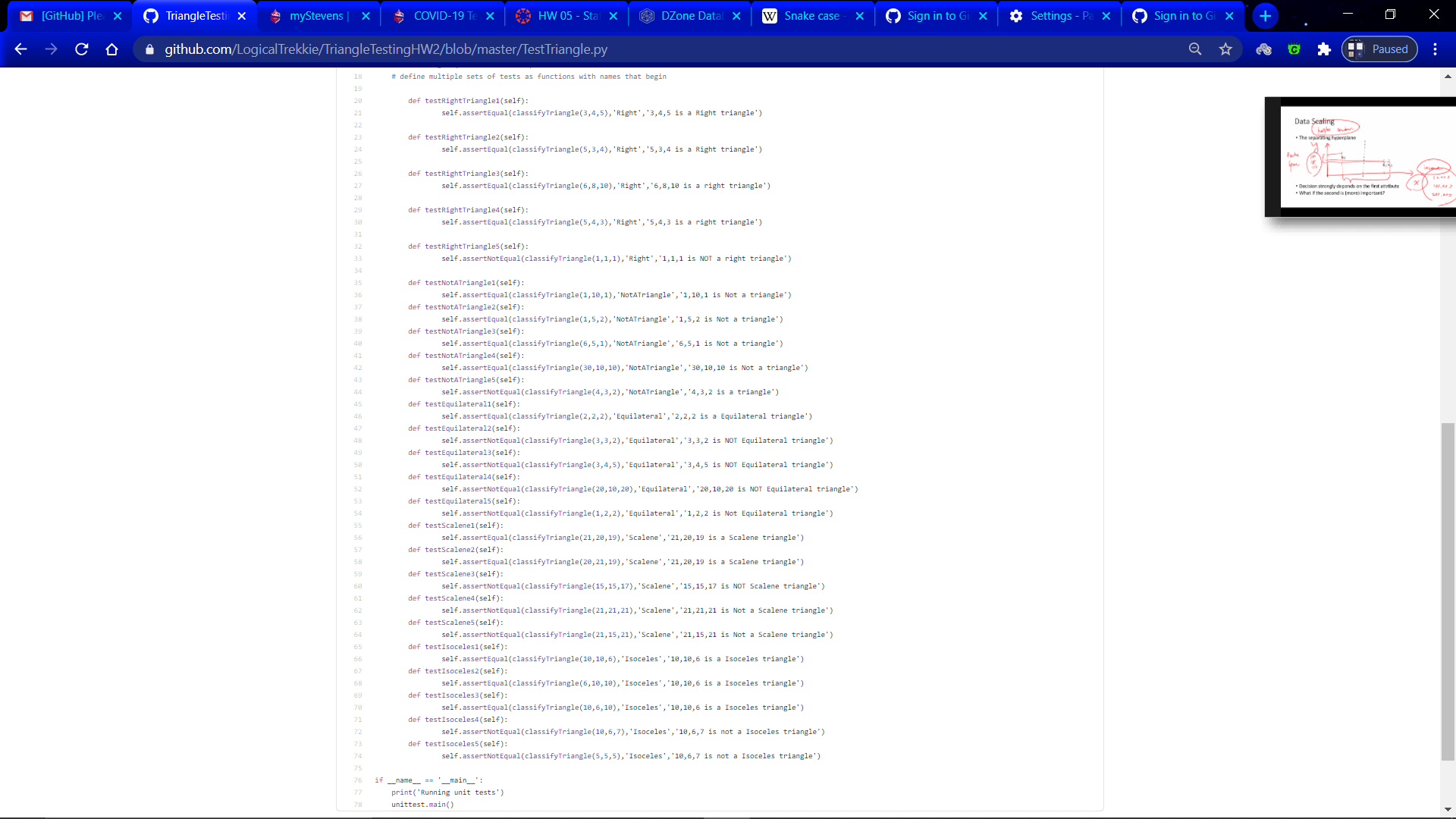
Coverage.py

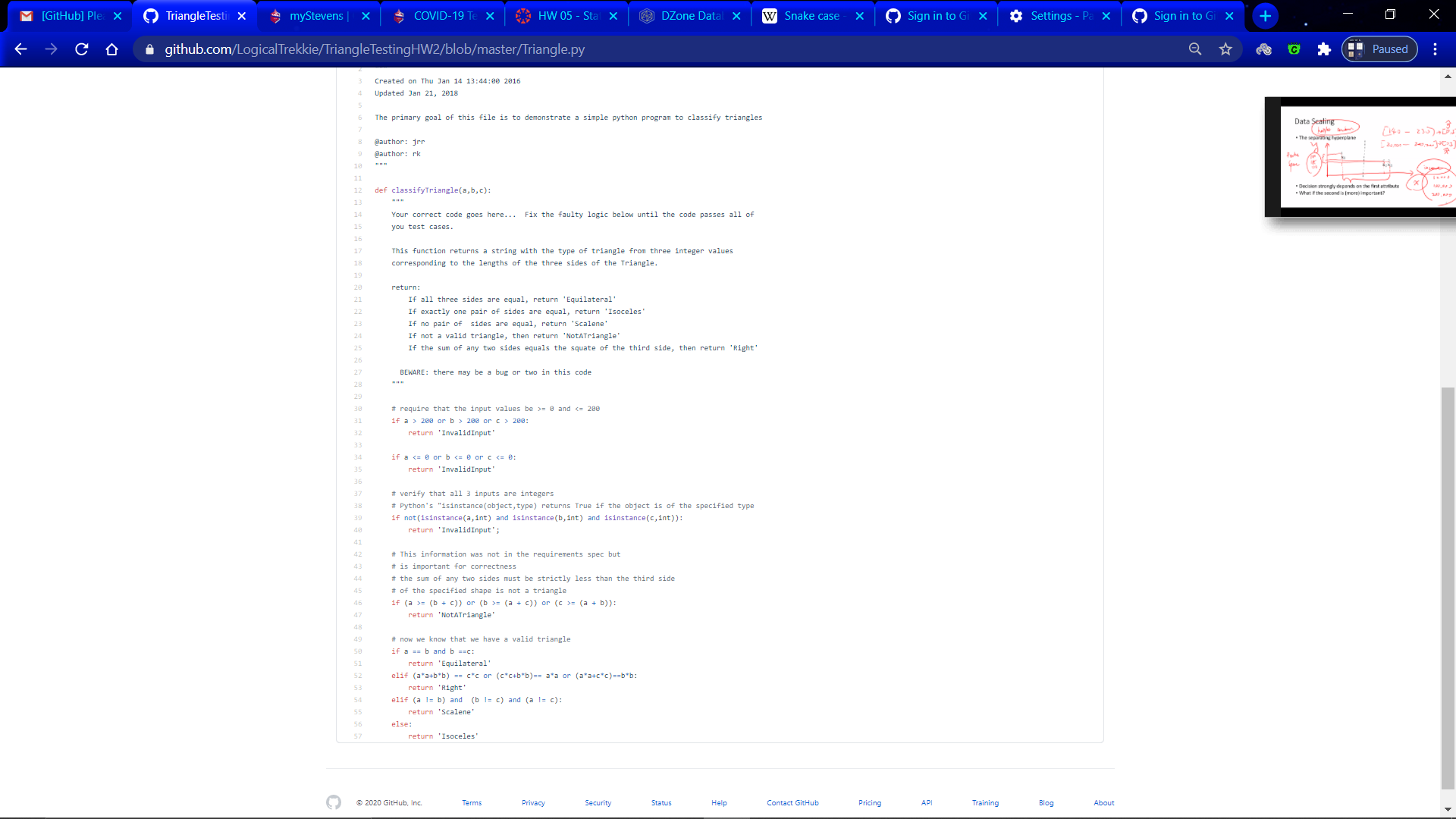


My coverage was at 99% so I did not change anything to fix this. I changed a lot for the Pylint which is detailed in the following screenshots.

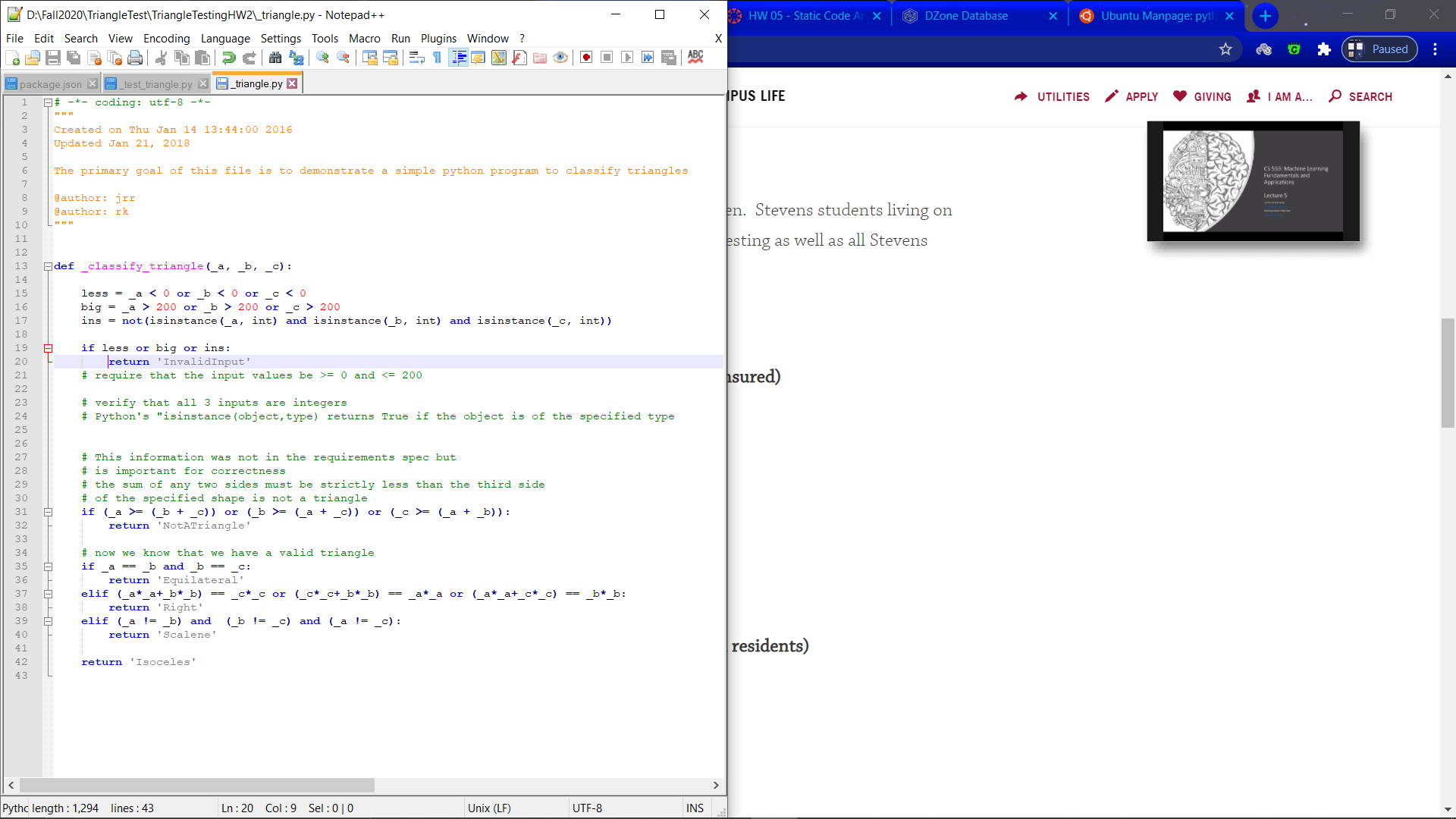
Pylint Before and After

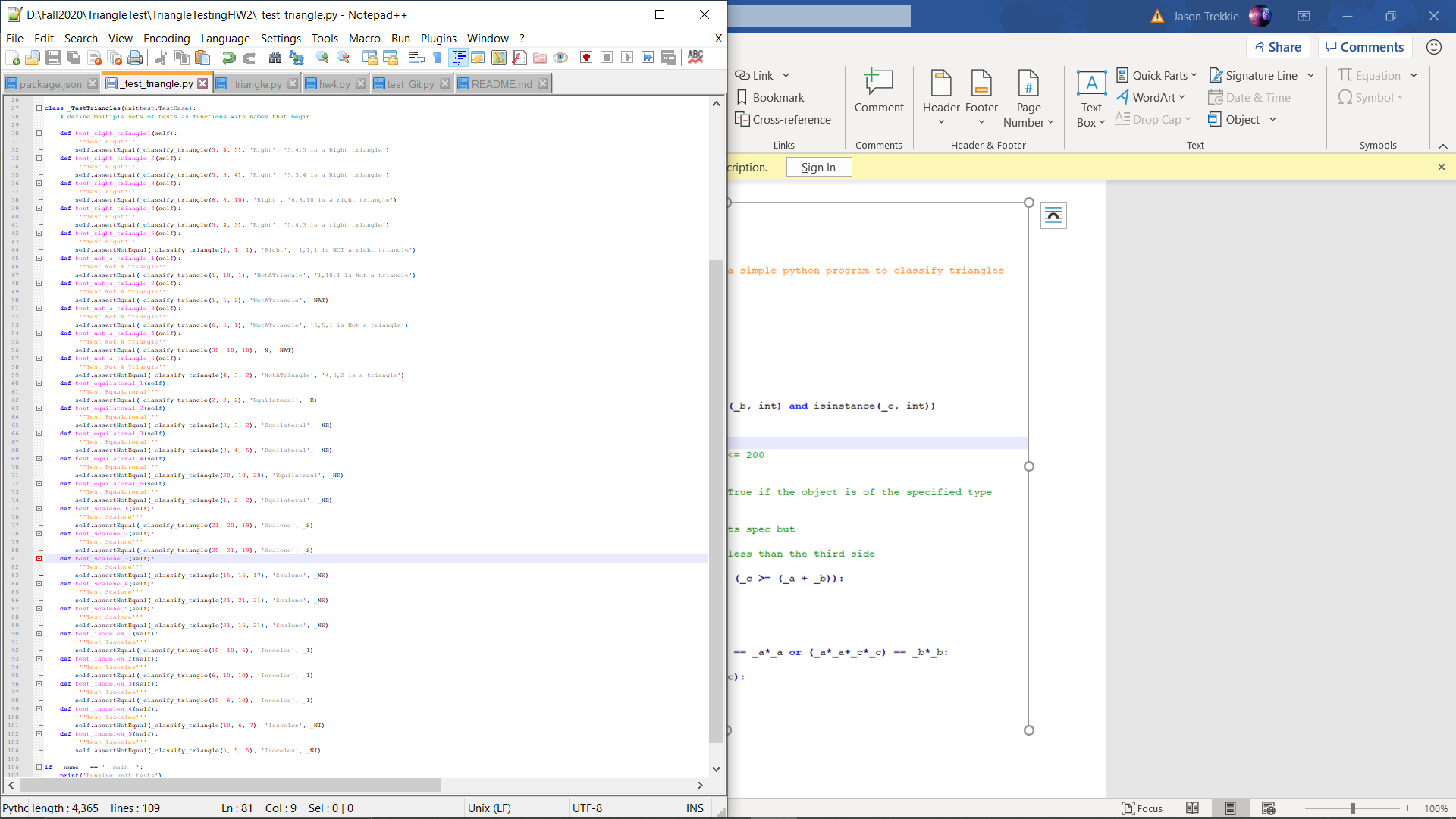
BEFORE:





AFTER:



I pledge my honor that I have abided by the stevens honor system.