Erik Buczek

SSW-567 SW Testing Qual. Assur. & Maintenance

03/05/19

HW 05 - Static Code Analysis

**Assignment Description:**

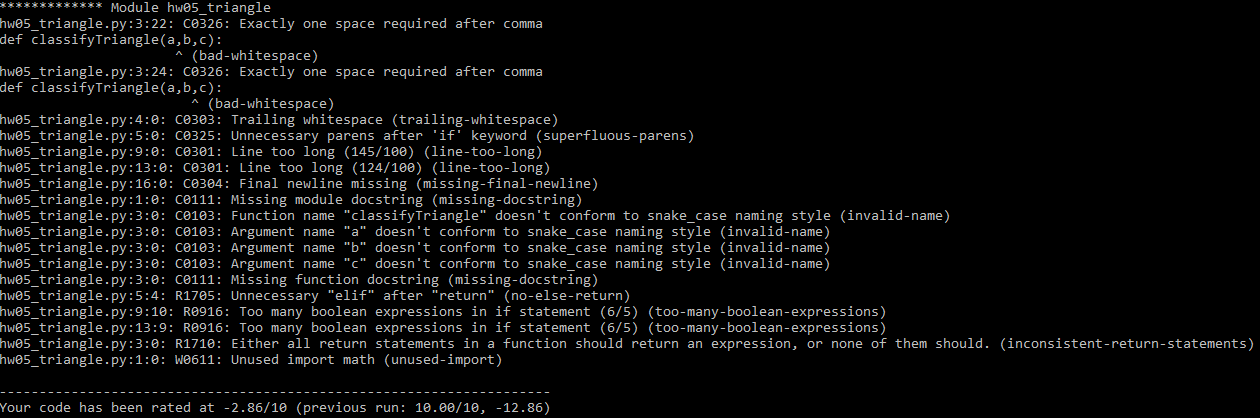
The objective of this assignment is to apply the techniques from the lecture to static testing of your Triangles program.

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.

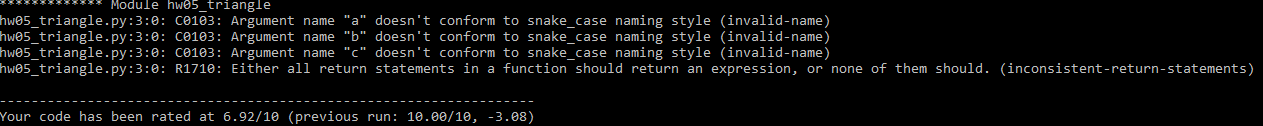
**Summary:**

**Part 1 - Static Code Analysis - Pylint**

**Before changes:**

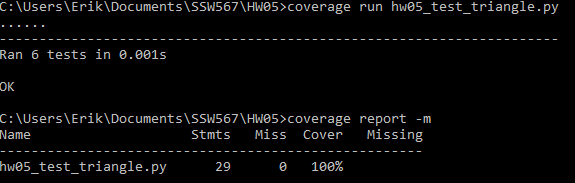
****

**After changes:**

****

**Part 2 - Code Coverage - Coverage module**

**Before changes (coverage = 100%):**

****

**Detailed Results:**

After running pylint on my code, I found that there were no errors. However, there were a few issues related to convention, refractors, and warnings. I eliminated all of the warnings and most of the refractors and conventions except one and 3 respectfully. The refractor issue that I left was related to “inconsistent return statements”. However, I am certain that all of my return statements are returning expressions. Also, pylint is calling out line 3 of my code, which does not make sense since there is no return statement there. The convention issues that I left was related to the naming conventions for my variables. Pylint wanted me to change the names of my variables to match snake\_case, which would be tedious, pointless, and cause more issues related to having very long lines of code. I am confident that I fixed all of the relevant issues identified by the static code analysis.

After running coverage, I found that my test cases already had a coverage of 100%. I only had statements within the test file, which had a coverage of 100%. Therefore, my overall coverage was 100%.

**Reflection:**

I enjoyed exploring the concepts posed by this assignment. Static code analysis and code coverage seem like incredibly useful tools. In my case, my triangle code did not contain any errors so Pylint did not find anything critical to report to me. However, I was interested to see all of the other issues that were in my code. I cleaned up the code and I think it looks better now. Coverage is also very useful, though it turned out that I already had full coverage of my code with my tests. I think that shows good planning on my part, to have included all of the necessary tests in the first place.

I pledge my honor that I have abided by the Stevens Honor System.