Chris Corrado

HW05: Static Code Analysis

9-29-2018

**Summary**

For this assignment, I opted to continue to use the SSW567 Repository which I set up for HW04 with the Travis CI pipeline. The YML configuration is set up to execute unit tests on any changes to the repository, provided the tests are in the tests/ directory.

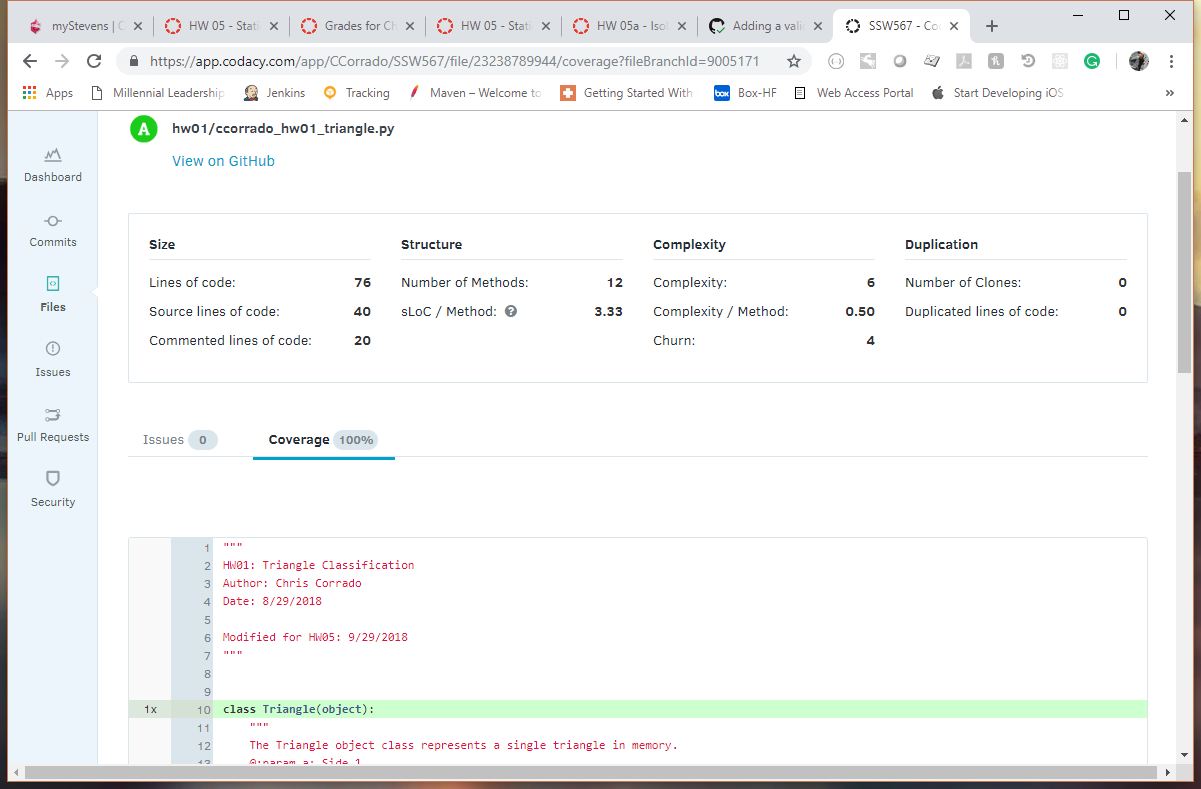
My HW01 Files live in this repository, and **Codecov** as well as **Codacy** both have scanned the Master branch which allowed me to create a pull request and trigger reports on the changes made for this assignment.

1. Github Link: <https://github.com/CCorrado/SSW567/blob/hw05/hw01/ccorrado_hw01_triangle.py>

* Github HW05 Pull Request: <https://github.com/CCorrado/SSW567/pull/3>

1. Codacy Test Coverage and Static Code Analysis:

* PRE changes: (There were no issues present) <https://app.codacy.com/app/CCorrado/SSW567/file/22891753046/coverage?bid=8916851&fileBranchId=8916851>
* POST changes: <https://app.codacy.com/app/CCorrado/SSW567/file/23238789944/coverage?fileBranchId=9005171>



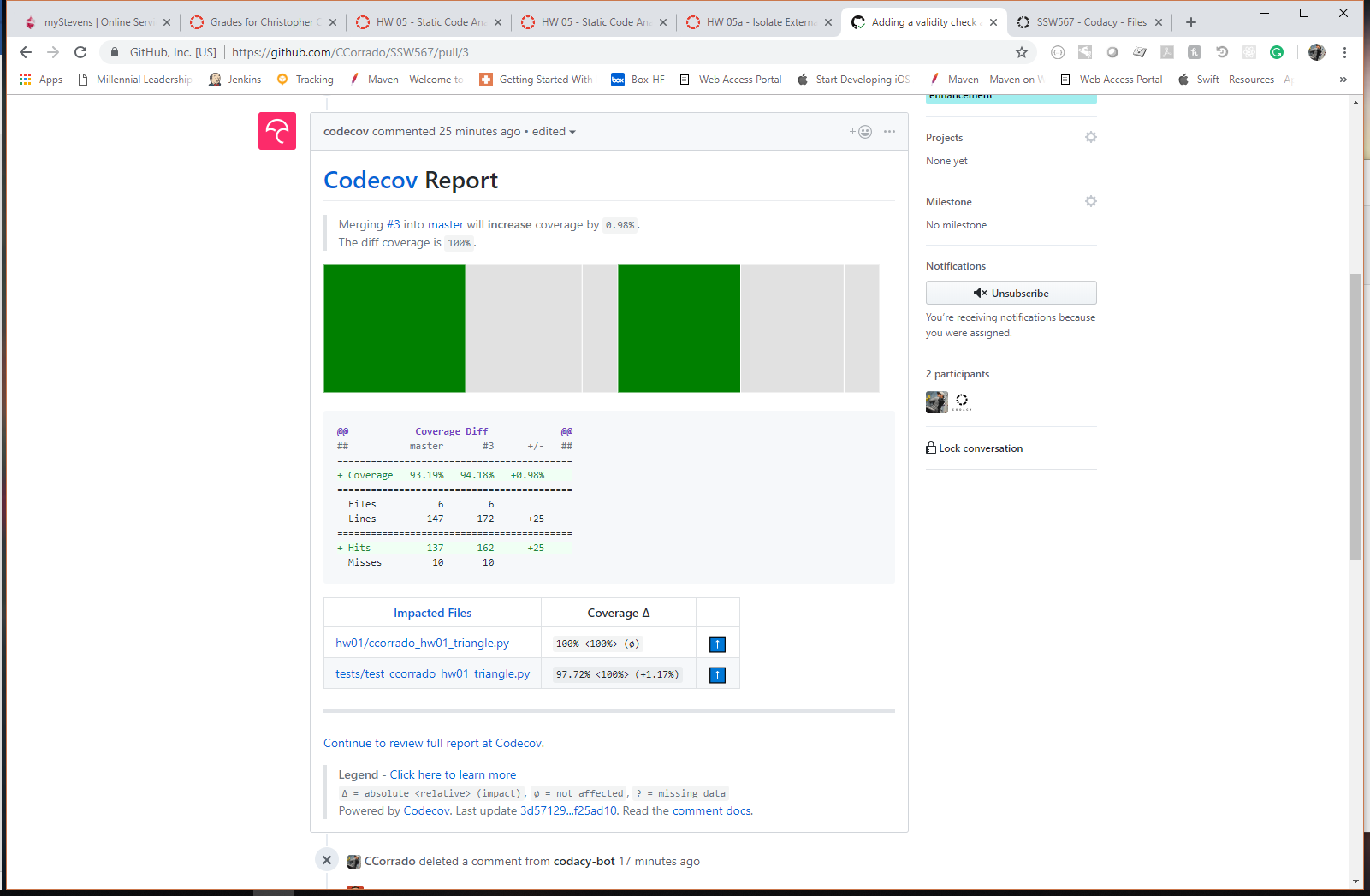
1. PRE-changes Coverage Report:

* <https://codecov.io/gh/CCorrado/SSW567/src/master/hw01/ccorrado_hw01_triangle.py>

1. POST changes Coverage Report

* <https://codecov.io/gh/CCorrado/SSW567/pull/3/diff?src=pr&el=tree#diff-aHcwMS9jY29ycmFkb19odzAxX3RyaWFuZ2xlLnB5>

1. Delta report between the two states.



1. Test Cases

* Before starting this assignment, there was 100% **statement** coverage on the Triangle file; however, the code was missing important validity checks. I added 4 new methods to assert validity of the triangle [(diff HERE)](https://github.com/CCorrado/SSW567/pull/3/commits/f25ad103dc9b9b7b011aef1e07dae42cbc423d47#diff-7714972ebdcc7a7e374afba8ead798e9) and 5 new test cases [(diff HERE)](https://github.com/CCorrado/SSW567/pull/3/commits/f25ad103dc9b9b7b011aef1e07dae42cbc423d47#diff-47b47c3862724b82acfab9e6e22e4642) in order to make up the missing conditions.
* Added Cases:

def test\_not\_a\_triangle(self):  
 triangle = classify\_triangle(0, 0, 0)  
 self.assertEqual(str(triangle), "Invalid")  
  
def test\_neg\_triangle(self):  
 triangle = classify\_triangle(-1, 1, 1)  
 self.assertEqual(str(triangle), "Invalid")  
  
def test\_neg\_triangle\_2(self):  
 triangle = classify\_triangle(1, -2, 1)  
 self.assertEqual(str(triangle), "Invalid")  
  
def test\_neg\_triangle\_3(self):  
 triangle = classify\_triangle(1, 1, -1)  
 self.assertEqual(str(triangle), "Invalid")  
  
def test\_str\_triangle(self):  
 triangle = classify\_triangle("not", "a", "triangle")  
 self.assertEqual(str(triangle), "Invalid")