Kipsy Quevada

SSW-567-WS: SW Testing, Qual. Assur. & Maint

Fall 2018 | Professor Rich Kempinski

HW 01: Testing triangle classification

Deliverable 3

* What challenges did you encounter with this assignment, if any?
  + My biggest challenge was sticking to the requirements detail for detail. The fact that we were expected to include a few bugs in the program was very confusing. I then realized as I edited the code that in order to completely follow the requirements, bugs were going to be inevitable, and it was a challenge in and of itself to submit buggy code in order to fulfill what was asked for.
* What did you think about the requirements specification for this assignment?
  + As mentioned in the previous question, the requirements specification was very unclear. It is impossible to fulfill these requirements without sacrificing the functionality of the code. For instance, a scalene triangle may also be a right triangle, so even if the requirements state that either ‘Scalene’ or ‘Right’ must be returned, functions have no way of returning more than one variable. Thus, the requirements specification should be corrected to return either ‘Scalene AND Right’ for those situations, or simply ‘Scalene’ if the right triangle specifications are not fulfilled. Additionally, the requirements specification does not specify the range nor the data type for either a, b, or c, which means that in situations when a, b, or c does not equal a positive integer, the program should be returning ‘NotATriangle’ instead of the other four options.
* What challenges did you encounter with the tools?
  + My biggest challenge was proving that the program was buggy through the unittests. I have worked with python unittests before through SSW 810, so learning the tools was not an issue, but never have I used the tests to verify buggy code. This was the first time I ever created a test set specifically for invalid inputs and used self.assertNotEqual() instead of self.assertEqual().
* Describe the criteria you used to determine that you had sufficient test cases, i.e. how did you know you were done?
  + I ensured that the test cases covered all ranges, boundaries, and test conditions. I tested negative, zero, and positive numbers, and proved that the requirements do not account for all values and data types. Additionally, I included test cases that tested for Equilateral, Isosceles, Scalene, Scalene AND Right, and NotATriangle cases.