

- What challenges did you encounter with this assignment, if any?

My biggest challenge was having to redo some of my work after I misread the instructions and wrote my code in a class with functions for each case instead of a single function (this is pertaining to the triangle classification, not the testing). Other than that, I am still a bit uncertain if I am approaching this assignment correctly. However, I believe that is bound to happen at the beginning of any new class or undertaking.

- What did you think about the requirements specification for this assignment?

I think they were relatively clear. In the case that these were used in systems requirements, there definitely could've been more detail such as what values would be considered valid entries (for example, decimals versus integers). However, it makes sense to exclude such specifications for the purpose of this assignment. I have used pytest before. After using unittest for this assignment, I found it to be equally helpful and easy to use on smaller projects.

- What challenges did you encounter with the tools?

I did encounter the issue of my test cases not being found and ran. This was relatively easy to fix, as after asking ChatGPT and revisiting the link provided in canvas, I was simply missing the `unittest.TestCase` parameter in my testing class. Note this is the only instance for which I consulted ChatGPT and/or any other outside sources. I have coded in Python and with Visual Studio Code before. As such, I had no trouble accessing and using these tools.

- Describe the criteria you used to determine that you had sufficient test cases, i.e. how did you know you were done?

My criteria for determining if I had sufficient test cases were based on the branches present within my function. Because I have one branch for each triangle type, there are a total of four. However, I purposefully included a bug where the "is right triangle" clause is never reached (as a triangle will always be equilateral, isosceles, or scalene; the function will return before then). Furthermore, rounding has not been accounted for. As such, the "is right triangle" branch is relatively buggy. Now onto the test cases.

An equilateral triangle can never be a right triangle. This leaves us with five options given these four branches: a triangle that is equilateral, a triangle that is isosceles but not right, a triangle that is scalene but not right, a triangle that is isosceles and right, and a triangle that is scalene and right. As such, there are five test cases.

If this was one testing iteration, the next step would be to solve the “is right triangle” bug to further reveal possible bugs (such as in the rounding mechanism). However, as the rounding bug is within the “is right triangle” test cases (the rounding bug would take place at exponentiation), this is also covered within these test cases. Therefore, I am done and have written enough test cases.