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

One of the main challenges I faced was making sure the program handled all kinds of inputs correctly. At first, I didn't think about negative numbers, zero, or non-numeric inputs, which caused some issues. I had to update my code to make sure only valid numbers were accepted for the sides of the triangle.
I also found it tricky to add intentional bugs. I wanted to make sure the bugs weren't too obvious but still helped me test if my unit tests were working correctly without breaking the program completely.

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

The requirements were clear in terms of what the program should do classify triangles as equilateral, isosceles, scalene, or right. However, they didn't really mention how to handle bad input like negative numbers or non-numeric values, which I had to figure out on my own. Overall, the instructions were straightforward and gave me room to think through some parts on my own.

3. What challenges did you encounter with the tools?

A challenge I had with the tools was getting used to Python's `unittest` framework. Since I'm not very familiar with it, I had to spend some time learning how to set up test cases and check if my results were correct.
I also had to be careful while using GitHub for version control. I've used Git before, but managing both the code and the test files, and making sure everything was tracked properly, took a bit of extra effort.

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

I knew I had enough test cases when I tested for all possible triangle types equilateral, isosceles, scalene, and right triangles along with cases where the sides don't form a valid triangle. I also added tests for invalid inputs like non-numeric values, negative numbers, and zero-length sides, just to make sure the program could handle them without crashing.
I also introduced a few bugs on purpose to see if my tests would catch them. When all my tests passed, I felt confident that I had enough coverage and was done with testing.