Brittany DiFede
Pledge: I pledge my honor that I have abided by the Stevens Honor System -Brittany DiFede

HW01 Assignment Deliverable 3

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

   The challenges that I faced were in regards to ensuring that I had my code cover all the requirements, make sure that my test cases were sufficient, and learning how to use unittest. I did not find the code itself challenging but more ensuring that I covered all that should be required challenging. I made sure to add a check for if the three sides formed a triangle after originally not thinking to check for that. Additionally, I have never worked with unittest in Visual Studio Code before, therefore I had to do some research and figure out how to use it as well as utilize its functions. With the testing, it was also a bit challenging to completely ensure that I was covering everything that needed to be tested. I made sure to look through my code, as well as the requirements in order to ensure that I was testing everything that I was able to think of that should be tested when looking at this assignment.

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

   I think that the requirements specification for this assignment provided a good start and overview. They introduced the types of triangles to consider, as well as what classifies each of these types. However, they did not cover everything that one may want to consider when approaching this assignment. When looking at this assignment with just the given requirements alone, one may fail to realize that it must be checked if the given three sides even form a triangle. Just because a shape has three sides doesn't necessarily make it a triangle. In order to address this, I made sure in my code to check if the three sides formed a triangle by using the Triangle Inequality Theorem, which is when the sum of two sides of the triangle is always greater than the third side. Additionally, if the three sides did not satisfy this Theorem/ my check, I had the code return that the triangle was not a triangle. Therefore, I think that in the requirements it could also be addressed what should be done when the sides do not form a triangle. I decided to just state that it is not a triangle, but an error could possibly be thrown as well.

3) What challenges did you encounter with the tools?

   I did not encounter any major challenges with these tools. I used Visual Studio Code in order to code in python for this assignment and within Visual Studio Code utilized unittest in order to create my test cases. I have never used unittest in Visual Studio Code prior to this, therefore the challenges that I faced were in regards to figuring out how to use unittest in Visual Studio code, as well as figuring out all of the uses of the assert methods.

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

   In order to determine that I had sufficient test cases, I first went through my code in order to determine what exactly I wanted to check. I knew that I wanted to test cases with both assertEqual and assertNotEqual. The reason I wanted to do this was to make sure that it was recognized when something was accurate, as well as not accurate. Additionally, in order to determine that I had sufficient test cases, I also wanted to ensure that I had test cases for all of the requirements specifications. In regards to the tests itself, one criteria that I wanted to test was that the input was even a triangle. In order to do this, I created test cases that had both triangles that were actual triangles, as well as tests that showed if the sides did not form a triangle. I also wrote test cases for each type of triangle equilateral, isosceles and scalene. Additionally, I wanted to test right triangle scenarios so did this by testing both isosceles and scalene right triangles since these types of triangles can be right triangles. In these test cases, I also tested when the sides were not in the correct order since the code should pick up which side is the hypothesis. Since equilateral triangles can't be right triangles, I made sure to cover test cases that proved that the triangles that were equilateral were not right triangles. Once I ensured that all requirements both the given and determined ones were tested, as well as having valid and invlaid cases tested, I was able to conclude that I had sufficient test cases.