Zack Edwards
SSW 567 Software testing, Quality Assurance and Maintenance
HW1 Triangle Classification
2/9/2021
Dr Raz Saremi

Deliverable 3

- One challenge I ran into while completing this assignment was how to verify right triangles. The problem is that in $a^2+b^2=c^2$, c must be the largest of the three numbers. The way I solved this was by inserting the numbers into a list and sorting the list in reverse order, making the first element the largest.
- I liked the requirements specification for this assignment as it gave me a clear idea of what the class needed to do. There were however two things which I had to make decisions as I did not feel they were properly addressed in the requirements.
    - One bit of wording which I found confusing, at the end where it says 'specifies whether the triangle is scalene, isosceles, or equilateral, and whether it is a right triangle as well' which seemed to mean report whether it is both scalene, isosceles or equilateral AND also report whether it is right or not on top of that. However after considering there are no equilateral and right triangles and almost all right triangles are scalene, I opted to simply report right triangles as a 4th option. Isosceles right triangles are rare and require more mathematical function to compute than are standard in python, and I did not want to use extra libraries, so I opted not to report on them.
    - Furthermore the requirements specification did not make stipulations about what kind of data types were not allowed. In order to avoid errors, I assumed based on the problem that negative numbers and words would not be suitable entries and so I filtered them out, returning 'invalid' when such things happened.
- I did not encounter problems with the toolset. I have worked comfortably with python before and unit tests which enabled me to use the tools involved in this lesson without trouble.
- The criteria I used to determine whether I had created enough tests was to create a set for each outcome including scalene, isosceles, equilateral, right, and invalid giving me 5 test sets. I focused on creating several varied tests for each using both assertEqual and assertNotEqual and trying things in each of the 3 parameters to ensure that everything worked as I intended.