**Deliverable-3**

* What challenges did you encounter with this assignment, if any?  
  + No challenges in the assignment except it is not clear the level of explanation that is expected for a perfect grade
* What did you think about the requirements specification for this assignment?  
  + It wasn’t clear how the triangle is to be classified in case it comes under more than one category

My output displays – “Right, scalene” if the triangle is both

* + It wasn’t clear how we should deal with float type numbers. Even if we round them, there can be errors e.g.

classify\_triangle(6.4,**6.404**,6.4) returns “Equilateral” if the values are rounded to two decimal places

classify\_triangle(6.4,**6.406**,6.4) returns “Isosceles”

* What challenges did you encounter with the tools?  
  + No challenges with the tools
* Describe the criteria you used to determine that you had sufficient test cases, i.e. how did you know you were done?
  + Checked if any side was equal to or less than zero
  + Checked if the sides form a valid triangle – Sum of any two sides should be greater than the third
  + Checked if all the inputs can be converted to a float number. E.g. “2” is a valid input; “two” is not a valid input
  + Changed the order of parameters
  + Tested at boundary conditions e.g. side value zero, just below zero or just above zero
  + Tested for each possible output e.g.

Right, scalene

Right, isosceles

Equilateral

Scalene

Isosceles

* + Included positive and negative tests i.e. assertEqual() and assertNotEqual()
  + Included float, integer and string inputs
  + Since a side of a triangle cannot be zero or negative, our equivalence class remains positive integers or float type numbers above the value of zero. Tested for each type of input for each type of possible output
  + Stopped testing when I had test cases for the types of exceptions that can be raised, test cases for each type of output – with changed parameter order, string, float and integer input