**Assignment 00B – Triangle**

**Deliverable 2**: Upload a text file or screen shot to show the input and output of running the program and demonstrating that your program has been adequately tested.

**Code : Triangle.py**

from math import isclose, isfinite

def classify\_triangle(a, b, c):

# Validate numeric, positive, finite inputs

for x in (a, b, c):

if not isinstance(x, (int, float)) or not isfinite(x) or x <= 0:

return "Not a triangle"

# Sort so c is the largest; helps the right-triangle check

a, b, c = sorted(map(float, (a, b, c)))

eps = 1e-9

# Triangle inequality (strict)

if a + b <= c + eps:

return "Not a triangle"

# Equilateral check

if isclose(a, b, abs\_tol=eps) and isclose(b, c, abs\_tol=eps):

return "Equilateral"

# Right-triangle check

if isclose(a\*a + b\*b, c\*c, rel\_tol=1e-9, abs\_tol=1e-9):

return "Right"

# Isosceles (with tolerance)

if (isclose(a, b, abs\_tol=eps) or

isclose(b, c, abs\_tol=eps) or

isclose(a, c, abs\_tol=eps)):

return "Isosceles"

return "Scalene"

if \_\_name\_\_ == "\_\_main\_\_":

try:

a = float(input("Enter side a: "))

b = float(input("Enter side b: "))

c = float(input("Enter side c: "))

print(f"Triangle type: {classify\_triangle(a, b, c)}")

except ValueError:

print("Not a triangle")

A black screen with white text

AI-generated content may be incorrect.

**Code: Test\_Triangle.py**

from math import sqrt

from triangle import classify\_triangle

def test\_equilateral():

assert classify\_triangle(2, 2, 2) == "Equilateral"

def test\_isosceles():

assert classify\_triangle(5, 5, 8) == "Isosceles"

def test\_scalene():

assert classify\_triangle(4, 5, 6) == "Scalene"

def test\_right\_scalene\_triplet\_and\_permutations():

assert classify\_triangle(3, 4, 5) == "Right"

assert classify\_triangle(4, 3, 5) == "Right"

assert classify\_triangle(5, 3, 4) == "Right"

def test\_right\_isosceles():

s = sqrt(2)

assert classify\_triangle(1, 1, s) == "Right"

def test\_not\_a\_triangle\_degenerate\_and\_invalid():

assert classify\_triangle(1, 2, 3) == "Not a triangle" # degenerate

assert classify\_triangle(0, 1, 1) == "Not a triangle" # non-positive

A black background with white text

AI-generated content may be incorrect.

**Deliverable 3**: Describe your experience with this assignment, specifically:

* What challenges did you encounter with this assignment, if any?
* What did you think about the requirements specification for this assignment?
  + The requirements were very easy and clear to follow.
* What challenges did you encounter with the tools?
  + At first, I couldn’t enter values in VS Code because it was in a read-only panel but then running it in the terminal fixed it.
* 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 was done once I had the test cases for equilateral, isosceles, scalene, right triangles, and non triangles cases and all of them passed without issues.

**Deliverable 4**: Submit the URL of the GitHub repo containing your complete solution.   The files in the repo should match what you have uploaded to Canvas.

<https://github.com/Aayush-Gandhi/SSW-567/tree/main/HW00b>