Christopher McKenzie

**Description:**

The primary goal of this file is to demonstrate a simple python program to classify triangles.

**Author:**

Christopher McKenzie

**Results:**

I fixed the classifyTriangle function and all of the predefined tests, along with my additional tests, passed.

**Reflection:**

This was the first time I had been given buggy code to fix and at least for this assignment, it came easily to me. I already had many different ideas of how to break the code from the previous assignments, so I did not run into any problems; everything worked. Also, it was my first time using bash to run Python code, so I was glad to finally be given the opportunity to do so.

**Honor Pledge:**

I pledge my honor that I have abided by the Stevens Honor System.

**Detailed Results/Strategy:**

My strategy was to find at least 2 different inputs for each possible result, including every possible triangle, incorrect input, and NotATriangle. I assumed that the correct number of parameters was always entered, and that there could not be an Isosceles Right Triangle or a Scalene Triangle. When I thought I fixed all defects, I ran the tests. When I discovered most of my tests failed with the similar incorrect result, I added print statements at the end of each return statement to see which conditions they were triggering. Once I determined where they were being directed, I found what was causing errors in that condition. Furthermore, I carefully read through the code to see if there were any other existing defects. After I fixed those defects, I tested again and only found 2 errors. At that point, it was self-explanatory where the errors were coming from. I determined I had enough tests when all defects were fixed, each of my tests passed, and I could not detect any more bugs. My test inputs included 2 examples of every type of triangle, and 2 different examples of how to initiate each error condition specified. My tests and their results confirmed that I had fixed the buggy classifyTriangle program, and that it could accurately classify a triangle.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Input** | **Expected Results** | **Actual Result** | **Pass or Fail** |
| 1RTA | classifyTriangle(3,4,5) | Right | InvalidInput | Fail |
| 1RTB | classifyTriangle(5,3,4) | Right | InvalidInput | Fail |
| 1ETA | classifyTriangle(1,1,1) | Equilateral | InvalidInput | Fail |
| 1ETB | classifyTriangle(50,50,50) | Equilateral | InvalidInput | Fail |
| 1ITA | classifyTriangle(2,4,4) | Isosceles | InvalidInput | Fail |
| 1ITB | classifyTriangle(8,8,3) | Isosceles | InvalidInput | Fail |
| 1STA | classifyTriangle(9,13,14) | Scalene | InvalidInput | Fail |
| 1STB | classifyTriangle(18,28,39) | Scalene | InvalidInput | Fail |
| 1IIA | classifyTriangle(0,0,0) | InvalidInput | InvalidInput | Pass |
| 1IIB | classifyTriangle(-1,1,1) | InvalidInput | InvalidInput | Pass |
| 1IIC | classifyTriangle(201,201,201) | InvalidInput | InvalidInput | Pass |
| 1IID | classifyTriangle(2,2,'two') | InvalidInput | TypeError | Fail |
| 1NATA | classifyTriangle(1,1,3) | NotATriangle | InvalidInput | Fail |
| 1NATB | classifyTriangle(2,7,9) | NotATriangle | InvalidInput | Fail |
| 2RTA | classifyTriangle(3,4,5) | Right | Right | Pass |
| 2RTB | classifyTriangle(5,3,4) | Right | Scalene | Fail |
| 2ETA | classifyTriangle(1,1,1) | Equilateral | Equilateral | Pass |
| 2ETB | classifyTriangle(50,50,50) | Equilateral | Equilateral | Pass |
| 2ITA | classifyTriangle(2,4,4) | Isosceles | Isoceles | Fail |
| 2ITB | classifyTriangle(8,8,3) | Isosceles | Isoceles | Fail |
| 2STA | classifyTriangle(9,13,14) | Scalene | Scalene | Pass |
| 2STB | classifyTriangle(18,28,39) | Scalene | Scalene | Pass |
| 2IIA | classifyTriangle(0,0,0) | InvalidInput | InvalidInput | Pass |
| 2IIB | classifyTriangle(-1,1,1) | InvalidInput | InvalidInput | Pass |
| 2IIC | classifyTriangle(201,201,201) | InvalidInput | InvalidInput | Pass |
| 2IID | classifyTriangle(2,2,'two') | InvalidInput | InvalidInput | Pass |
| 2NATA | classifyTriangle(1,1,3) | NotATriangle | NotATriangle | Pass |
| 2NATB | classifyTriangle(2,7,9) | NotATriangle | NotATriangle | Pass |
| 3RTA | classifyTriangle(3,4,5) | Right | Right | Pass |
| 3RTB | classifyTriangle(5,3,4) | Right | Scalene | Pass |
| 3ETA | classifyTriangle(1,1,1) | Equilateral | Equilateral | Pass |
| 3ETB | classifyTriangle(50,50,50) | Equilateral | Equilateral | Pass |
| 3ITA | classifyTriangle(2,4,4) | Isosceles | Isosceles | Pass |
| 3ITB | classifyTriangle(8,8,3) | Isosceles | Isosceles | Pass |
| 3STA | classifyTriangle(9,13,14) | Scalene | Scalene | Pass |
| 3STB | classifyTriangle(18,28,39) | Scalene | Scalene | Pass |
| 3IIA | classifyTriangle(0,0,0) | InvalidInput | InvalidInput | Pass |
| 3IIB | classifyTriangle(-1,1,1) | InvalidInput | InvalidInput | Pass |
| 3IIC | classifyTriangle(201,201,201) | InvalidInput | InvalidInput | Pass |
| 3IID | classifyTriangle(2,2,'two') | InvalidInput | InvalidInput | Pass |
| 3NATA | classifyTriangle(1,1,3) | NotATriangle | NotATriangle | Pass |
| 3NATB | classifyTriangle(2,7,9) | NotATriangle | NotATriangle | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Run 1** | **Test Run 2** | **Test Run 3** |
| **Tests Planned** | 14 | 14 | 14 |
| **Tests Executed** | 14 | 14 | 14 |
| **Tests Passed** | 3 | 11 | 14 |
| **Defects Found** | 6 | 2 | 0 |
| **Defects Fixed** | 6 | 2 | 0 |

**Repo:**

https://github.com/camckenzie/Triangle