HW 02a - Testing a legacy program and reporting on testing results.

1. **Assignment Description**: Sometimes you will be given a program that someone else has written, and you will be asked to fix, update, and enhance that program.   In this assignment you will start with an existing implementation of the classify triangle program that will be given to you.   You will also be given a starter test program that tests the classify triangle program, but those tests are not complete.

2. **Author**: Shashank Janagam Chandra

3. **Summary**: The assignment involved implementing a Python script for triangle classification and creating corresponding test cases using the unittest module. The classify triangle function was designed to categorize triangles as equilateral, isosceles, scalene, or right based on their side lengths. The accompanying test cases covered various scenarios, including valid triangle types and cases with invalid inputs.

**This assignment provided valuable hands-on experience with test-driven development principles, specifically using the unittest module in Python. The process emphasized the importance of clear and descriptive test case names for quick identification of failures. Crafting a variety of test cases, including invalid inputs and edge cases, contributed to building a robust test suite. The assignment reinforced the significance of automated testing in ensuring the correctness of functions. Overall, it enhanced understanding and proficiency in creating effective tests to validate code functionality.**

**4. Honor Pledge: I affirm that the work provided here is our own. Any external resources used have been appropriately cited.**

***Deliverables:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Results | Actual Results | Pass/Fail |
| 1 | 3, 4, 5 | Right | Invalid Input | Fail |
| 2 | 5, 3, 4 | Right | Invalid Input | Fail |
| 3 | 1, 1, 1 | Equilateral | Invalid Input | Fail |

1. Results of running initial test sets
2. Uploaded improved code with file named Triangle.py.
3. Uploaded the test set file named TestTriangle.py.

A screenshot of a computer

Description automatically generated

1. Improved implementation of classify triangle.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Input | Expected Results | Actual Results | Pass or fail |
| 1 | 1, 1, 1 | Equilateral | Equilateral | Pass |
| 2 | 3, 4, 5 | Right | Right | Pass |
| 3 | 3, 3, 4 | Isosceles | Isosceles | Pass |
| 4 | -3, 4, 5 | Invalid Input | Invalid Input | Pass |

1. Assignment summary results,

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Run 1 | Test Run 2 | Test Run 3 |
| Tests Planned | 3  (Equilateral,  Right-Angled) | 5  (Equilateral, Isosceles, Scalene, Right-Angled, Invalid Input) | 5  (Equilateral, Isosceles, Scalene, Right-Angled, Invalid Input) |
| Tests Executed | All planned tests | All planned tests | All planned tests |
| Tests Passed | None passed | 2 passed and 3 failed | All passed without failure |
| Defects Found | AssertionError: 'InvalidInput' != 'Right' | AssertionError: 'InvalidInput' != 'Right' | None found |
| Defects Fixed | Condition to be right angle triangle fixed | Conditions to check if its triangle or not fixed. | None |

1. [GitHub Repo](https://github.com/SHASHANKJC/TestTriangle)