Assignment: Classic Triangle Testing Problem (Myer's Triangle)

Introduction

The Classic Triangle Testing Problem is a well-known example in software testing, focusing on developing a test suite to verify the correctness of a triangle classification program. The program takes three integer inputs and classifies the triangle as **equilateral**, **isosceles**, or **scalene** based on the side lengths. Additionally, it must also account for invalid triangle inputs, where the sides do not satisfy the triangle inequality.

Test Case Design

To ensure adequate coverage, we consider **valid triangle types** and **invalid input conditions**. A triangle is valid if the sum of the lengths of any two sides is greater than the third side.

Test Case 1: Equilateral Triangle

- **Input:** 6, 6, 6
- **Expected Output:** Equilateral
- **Explanation:** All sides are equal, satisfying the triangle inequality (6 + 6 > 6).

Test Case 2: Isosceles Triangle

- **Input:** 5, 5, 3

- **Expected Output:** Isosceles
- **Explanation:** Two sides are equal. Triangle inequality holds (5 + 5 > 3, etc.).

Test Case 3: Scalene Triangle

- **Input:** 4, 5, 6
- **Expected Output:** Scalene
- **Explanation:** All sides are different and satisfy triangle conditions.

Test Case 4: Invalid Triangle (Triangle Inequality Violation)

- **Input:** 1, 2, 3
- **Expected Output:** Not a Triangle
- **Explanation:** 1 + 2 is not greater than 3; violates triangle rule.

Test Case 5: Zero-Length Side

- **Input:** 0, 5, 5
- **Expected Output:** Not a Triangle
- **Explanation:** Zero side length is invalid.

Test Case 6: Negative Side Value

- **Input:** -3, 4, 5
- **Expected Output:** Not a Triangle
- **Explanation:** Negative length side is invalid input.

Justification

These test cases follow **equivalence partitioning** and **boundary value analysis**, which are core techniques in software testing (Myers, 2011). They ensure both normal and abnormal inputs are tested, covering structural and functional aspects of the problem (Pressman & Maxim, 2020).

Conclusion

The Classic Triangle Testing Problem provides a solid foundation for understanding the importance of thorough test case design. By using six carefully selected test cases, this assignment demonstrates how to test all possible triangle classifications—**equilateral**, **isosceles**, and **scalene**—as well as invalid inputs such as negative or zero-length sides and violations of the triangle inequality. These cases ensure that the program is robust, handles edge conditions, and produces accurate results. Applying software testing principles like **equivalence partitioning** and **boundary value analysis** leads to more reliable and maintainable code, which is essential in real-world software development.

References

Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). Wiley. https://www.wiley.com/en-us/The+Art+of+Software+Testing%2C+3rd+Edition-p-9781119202486

Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education. https://www.amazon.com/Software-Engineering-Practitioners-Roger-Pressman/dp/0078022126

Wordcount: 391