**Design:**

My code consists of all the test cases in the main method, with each case calling the triangleCheck method, which takes in three integers representing the three sides of the triangle. This method has a void return since it instead prints the result. The triangleCheck method is put in the TriangleChecker class for encapsulation and to improve organization.

Within the triangleCheck method, I first check to see if the three integer inputs are valid. These checks will print error statements and end if any value is less than or equal to zero, or if any two sides are less than or equal to the other. Next, we can validate if the inputs correspond to an equilateral, isosceles, or scalene triangles, printing the type of triangle as a result.

**Implementation:**

#include <iostream>

using namespace std;

// Triangle class encapsulates the triangle logic

class TriangleChecker {

public:

/\*

triangleCheck reads three integer values, representing the integer lengths of the sides of a triangle.

If the inputs are valid, the program prints a message that states whether the triangle is

- equilateral (three side equal),

- isosceles (two side equal), or

- scalene (no side equal).

\*/

void triangleCheck(int side1, int side2, int side3) {

// Check that all inputs are positive integers

if (side1 <= 0 || side2 <= 0 || side3 <= 0) {

cout << "Invalid triangle. Sides must be greater than 0." << endl;

}

// Check that the triangle inequality holds

else if (side1 + side2 <= side3 || side1 + side3 <= side2 || side2 + side3 <= side1) {

cout << "Invalid triangle. The sum of any two sides must be greater than the third." << endl;

}

// Find type of triangle

else if (side1 == side2 && side2 == side3) {

cout << "Equilateral triangle!" << endl;

}

else if (side1 == side2 || side2 == side3 || side1 == side3) {

cout << "Isosceles triangle!" << endl;

}

else {

cout << "Scalene triangle!" << endl;

}

}

};

/\*

Main method to run and test triangleCheck method

\*/

int main() {

cout << "Determine the type of triangle based on side lengths!" << endl;

// Create an object of TriangleChecker

TriangleChecker checker;

// Invalid test cases

cout << "\nsides = (0,0,0)" << endl;

checker.triangleCheck(0,0,0);

cout << "\nsides = (4,-3,2)" << endl;

checker.triangleCheck(4,-3,2);

cout << "\nsides = (1,2,3)" << endl;

checker.triangleCheck(1,2,3);

// Equilateral triangle test cases

cout << "\nsides = (5,5,5)" << endl;

checker.triangleCheck(5,5,5);

cout << "\nsides = (100,100,100)" << endl;

checker.triangleCheck(100,100,100);

// Isosceles triangle test cases

cout << "\nsides = (3,3,5)" << endl;

checker.triangleCheck(3,3,5);

cout << "\nsides = (100,150,100)" << endl;

checker.triangleCheck(100,150,100);

// Scalene triangle test cases

cout << "\nsides = (4,3,2)" << endl;

checker.triangleCheck(4,3,2);

cout << "\nsides = (40,60,50)" << endl;

checker.triangleCheck(40,60,50);

return 0;

}

**Correctness:**

*% g++ triangleTest.cpp -o triangleTest && ./triangleTest*

Determine the type of triangle based on side lengths!

sides = (0,0,0)

Invalid triangle. Sides must be greater than 0.

sides = (4,-3,2)

Invalid triangle. Sides must be greater than 0.

sides = (1,2,3)

Invalid triangle. The sum of any two sides must be greater than the third.

sides = (5,5,5)

Equilateral triangle!

sides = (100,100,100)

Equilateral triangle!

sides = (3,3,5)

Isosceles triangle!

sides = (100,150,100)

Isosceles triangle!

sides = (4,3,2)

Scalene triangle!

sides = (40,60,50)

Scalene triangle!