Activity No. 2

REVIEW OF C++ PROGRAMMING

Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed:** September 9, 2024
Section: CPE21S4	**Date Submitted:** September 10, 2024
Name(s): Bona, Andrei Nycole So	**Instructor:** Prof. Maria Rizette Sayo

6. Output

Sections	**Answer**

Header File Declaration Section	#include <iostream>
	using namespace std;

Global Declaration Section	N/A

Class Declaration and
Method Definition Section
	class Triangle
	{
	    private:
	        double totalAngle, angleA, angleB, angleC;
	
	    public:
	        Triangle(double A, double B, double C);
	        void setAngles(double A, double B, double C);
	        const bool validateTriangle();
	};

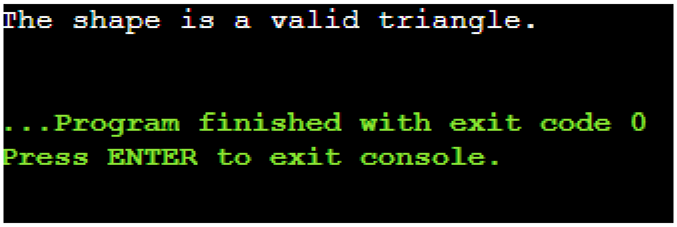
Main Function	int main()
	{
	    Triangle set1(40, 30, 110);
	    if(set1.validateTriangle())
	    {
	        cout << "The shape is a valid triangle.\n";
	    }
	        else
	        {
	            cout << "The shape is NOT a valid triangle.\n";
	        }
	    return 0;
	}

Method Definition	Triangle::Triangle(double A, double B, double C)
	{
	    angleA = A;
	    angleB = B;
	    angleC = C;
	    totalAngle = A+B+C;
	}

```
void Triangle::setAngles(double A, double B, double C)
{
    angleA = A;
    angleB = B;
    angleC = C;
    totalAngle = A+B+C;
}

const bool Triangle::validateTriangle()
{
    return (totalAngle <= 180);
}
```

Table 1-1. C++ Structure Code for Answer



```
The shape is a valid triangle.


...Program finished with exit code 0
Press ENTER to exit console.
```

The program correctly determines if a given set of angles is a valid triangle. The validateTriangle function determines if the sum of the angles is equal to 180 degrees, a fundamental characteristic of triangles. But in theory, I notice that the condition on the function is **less than or equal to 180**, a triangle **must be equal** to 180° because the sum of all angles of a triangle cannot exceed or less than 180°. Nonetheless, the program returns "The shape is a valid triangle" or "The shape is NOT a valid triangle." accordingly.

Table 1-2. ILO B output observations and comments.

7. Supplementary Activity

1. Create a C++ program to swap the two numbers in different variables.

```
#include <iostream>
using namespace std;

int num1, num2, null;

int main ()
{
    cout << "Enter your first Number:";
    cin >> num1;
    cout << "Enter your second Number:";
    cin >> num2;
            cout << "Before the swap:" <<endl;
            cout << num1<<endl;
            cout << num2<<endl;

            null = num1;
            num1 = num2;
            num2 = null;
            cout << "After the swap:" <<endl;
            cout << num1<<endl;
            cout << num2<<endl;

    return 0;
}
```

2. Create a C++ program that has a function to convert temperature in Kelvin to Fahrenheit.

```
#include <iostream>
using namespace std;

double tempK, tempF;

double convertTemp(double tempK)
{
    return (9.0/5.0)*(tempK - 273.15) + 32.0;
}

int main ()
{
    cout << "Enter the temperature: ";
    cin >> tempK;

    tempF = convertTemp (tempK);
    cout << "The temperature from Kelvin to Fahrenheit: " <<tempF;
    return 0;
}
```

3. Create a C++ program that has a function that will calculate the distance between two points.

```
#include <iostream>
#include <cmath>
using namespace std;

double calculate(double x1, double x2, double y1, double y2)
{
    double a = pow(x2 - x1, 2);
    double b = pow(y2 - y1, 2);

    double distance = sqrt(a + b);

    return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));
}
int main ()
{
    double x1, x2, y1, y2, answer;
    cout << "Enter first point for x1: ";
    cin >>x1;
    cout << "Enter second point for x2: ";
    cin >>x2;
    cout << "Enter first point for y1: ";
    cin >>y1;
    cout << "Enter second point for y2: ";
    cin >>y2;
    cout <<endl;


    answer = calculate(x1, x2, y1, y2);
    cout << "The distance between two points: "<<answer;

    return 0;

}
```

4. Modify the code given in ILO B and add the following functions:
a. A function to compute for the area of a triangle
b. A function to compute for the perimeter of a triangle
c. A function that determines whether the triangle is acute-angled, obtuse-angled or ‘others.’

```
#include <iostream>
#include <cmath>
using namespace std;

class Triangle
{
    private:
        double totalAngle, angleA, angleB, angleC;
    public:
        Triangle(double A, double B, double C);

        double area();
        double perimeter();
        string type();

        void setAngles(double A, double B, double C);
        const bool validateTriangle();

};

Triangle::Triangle(double A, double B, double C)
{
    angleA = A;
    angleB = B;
    angleC = C;
    totalAngle = A+B+C;
}

void Triangle::setAngles(double A, double B, double C)
{
    angleA = A;
    angleB = B;
    angleC = C;
    totalAngle = A+B+C;
}

const bool Triangle::validateTriangle()
{
    return (totalAngle == 180);
}

double Triangle::area()
{
    double s = (angleA + angleB + angleC) / 2;
    double area = sqrt(s * (s - angleA) * (s - angleB) * (s - angleC));//Heron's Formula
    return area;
}

double Triangle::perimeter()
{
    return totalAngle;
}

string Triangle::type()
{
    if (angleA > 90 || angleB > 90 || angleC > 90)
        return "obtuse-angled";
    else if (angleA == 90 || angleB == 90 || angleC == 90)
        return "right-angled";
    else
        return "acute-angled";
}
int main()
{

    Triangle set1(40, 30, 110);
    if(set1.validateTriangle())
    {
        cout << "The shape is a valid triangle.\n";
        cout << "The area of the triangle is: "<<set1.area()<<"\n";
        cout << "The perimeter of the triangle is: "<<set1.perimeter()<<"\n";
        cout << "The triangle is " << set1.type() << "\n";
    }
        else
        {
            cout << "The shape is NOT a valid triangle.\n";
        }


    return 0;
}
```

8. Conclusion

In conclusion, this laboratory covers many problems, including developing proper class definitions and instances. We also utilized mathematical methods, such as Heron's formula, to calculate the triangle area with the given three different sides and conditional statements to identify the triangle type, changing the two numbers in separate variables, calculating the distance between two points, and converting temperature – Kelvin to Fahrenheit. To further clarify the proper use of methods, I use additional resources. Overall, this activity demonstrated our ability to develop efficient and effective C++ programs to tackle a wide range of issues while also providing a thorough grasp of the language and its applications.

9. Assessment Rubric