Activity No. <1>	
<REVIEW OF C++ PROGRAMMING >	
Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed: Sept 9, 2024**
Section: CPE21S4	**Date Submitted: Sept 10, 2024**
Name(s): Kenn Jie Valleser	**Instructor: Ma'am Rizette Sayo**
6. Output	

Table 1-1 Structure Code of Answer

Sections | **Answer**

Header File Declaration Section

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

Global Declaration Section

Class Declaration and Method Definition Section

```
class Utility {
public:
    void displaySum(int a, int b)
    bool isGreater(int a, int b)
    bool logicalOperations(bool x, bool y) {
```

Main Function

```
int main() {
    Utility util;

    //Sum
    int a = 12, b = 10;
    cout << "Sum:" << endl;
    util.displaySum(a, b);

    // is Greater
    bool result = util.isGreater(a, b);
    cout << a << " is ";
    if (result) {
        cout << "greater than ";
    } else {
        cout << "not greater than ";
    }
    cout << b << endl;

    // Test logicalOperations
    bool x = true, y = false;
    bool success = util.logicalOperations(x, y);

    cout << "Logical operations executed successfully: ";
    if (success) {
        cout << "true";
    } else {
        cout << "false";
    }
    cout << endl;

    return 0;
}
```

Method Definition

```
void displaySum(int a, int b) {
        int sum = a + b;
```

	` cout << "Sum: " << sum << endl;`
	` }`
	
	` // Method to return whether a is greater than b`
	` bool isGreater(int a, int b) {`
	` return a > b;`
	` }`
	
	` // Method to display results of logical operations and return true if successful`
	` bool logicalOperations(bool x, bool y) {`
	` cout << "x AND y: " << (x && y) << endl;`
	` cout << "x OR y: " << (x || y) << endl;`
	` cout << "x XOR y: " << (x != y) << endl;`
	` cout << "NOT x: " << (!x) << endl;`
	` cout << "NOT y: " << (!y) << endl;`
	
	` // Logical operations are considered successful if executed`
	` return true;`
	` }`
	`};`

Table 1-2 Output ILO B

The shape is a valid triangle.

//The output displayed “The shape is a valid triangle”. because the sides is the sum of 180 which accepted in response to the output

7. Supplementary Activity

1)

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

void swapNumbers(int &a, int &b) {
   int temp = a;
   a = b;
   b = temp;
}

int main() {
   // Swap two numbers
   int num1 = 10, num2 = 20;
   cout << "Before swapping: " << endl;
   cout << "num1 = " << num1 << ", num2 = " << num2 << endl;
   swapNumbers(num1, num2);
   cout << "After swapping: " << endl;
   cout << "num1 = " << num1 << ", num2 = " << num2 << endl;
return 0;
}
```

2)
```
#include <iostream>
using namespace std;

double kelvinToFahrenheit(double kelvin) {
    return (kelvin * 9/5) - 459.67;
   }
int main() {

// Convert Kelvin to Fahrenheit
    double kelvin = 273.15;
    double fahrenheit = kelvinToFahrenheit(kelvin);
    cout << "Kelvin to Fahrenheit: " << kelvin << "K = " << fahrenheit << "F" << endl;
return 0;
}
```

3)
```
#include <iostream>
#include <cmath>
using namespace std;

double distanceBetweenPoints(double x1, double y1, double x2, double y2) {
    return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2)); }
int main() {
 // Calculate distance between two points
    double x1 = 1, y1 = 3, x2 = 5, y2 = 10;
    double distance = distanceBetweenPoints(x1, y1, x2, y2);
    cout << "Distance between points (" << x1 << ", " << y1 << ") and (" << x2 << ", " << y2 << "): " << distance << endl;

    return 0;
}
```

4)
```
#include <iostream>
#include <cmath>

class Triangle {
private:
    double totalAngle, angleA, angleB, angleC;
    double sideA, sideB, sideC;

public:

    Triangle(double A, double B, double C);


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


    double computeArea(double a, double b, double c) const;


    double computePerimeter() const;


    std::string triangleType() const;
};


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

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

// Validate if the angles make a valid triangle
bool Triangle::validateTriangle() const {
    return (totalAngle == 180);
}


double Triangle::computeArea(double a, double b, double c) const {
    double s = (a + b + c) / 2; // semi-perimeter
    return sqrt(s * (s - a) * (s - b) * (s - c));
}


double Triangle::computePerimeter() const {
    return sideA + sideB + sideC;
}


std::string Triangle::triangleType() const {
    if (angleA < 90 && angleB < 90 && angleC < 90) {
        return "Acute-angled";
    } else if (angleA > 90 || angleB > 90 || angleC > 90) {
        return "Obtuse-angled";
    } else {
        return "Right-angled";
    }
}
```

```
int main() {
    double A = 120, B = 30, C = 30; // Example angles with one obtuse angle
    double sideA = 7, sideB = 5, sideC = 5; // Example sides

    Triangle set1(A, B, C);


    if (set1.validateTriangle()) {
        std::cout << "The shape is a valid triangle.\n";


        std::cout << "The area of the triangle is: " << set1.computeArea(sideA, sideB, sideC) << std::endl;
        std::cout << "The perimeter of the triangle is: " << set1.computePerimeter() << std::endl;


        std::cout << "The triangle is: " << set1.triangleType() << std::endl;
    } else {
        std::cout << "The shape is NOT a valid triangle.\n";
    }

    return 0;
}
```

8. Conclusion

Summary of lessons learned
i learned that there is a lot of new syntax and keywords that i didn't know like encapsulation public: and private:
Analysis of the procedure
The syntax were complex for me since my last studied programming language is python
Analysis of the supplementary activity
most of the problems are easy for me since its in my familiarity and i did this a long time ago
Concluding statement / Feedback: How well did you think you did in this activity? What are your areas for improvement?
i did bad because most of the time i asked google for what keyword to replace my errors and my areas for improvement were i should be restudy c++ and practice more using it.

9. Assessment Rubric