

| Hands on Activity 1.2   |   |
|---|---|
| Basic C++ Programming   |   |
| Course Code: CPE010   | Program: Computer Engineering   |
| Course Title: Data Structures and Algorithms  | Date Performed: 09/09/24  |
| Section: CPE21S4  | Date Submitted: 09/10/24  |
| Name(s): Rodriguez, Kyle Franz P.   | Instructor: Maria Rizette Sayo  |
| 6. Output   |   |
| ILO A   |   |
| Section   | Code  |
| Header File<br>Declaration Section  | <pre> C/C++ #include &lt;iostream&gt; using namespace std; </pre>   |
| Create a function that will take the sum of 2 numbers   | <pre> C/C++ int sum(int num1,int num2) {     return (num1 + num2); }; </pre>  |
| Create a function that will output the greater number   | <pre> C/C++ int more(int num1,int num2) {     if (num1 &gt; num2) {         return num1;     }     else {         return num2;     } }; </pre>  |
| Create a function that will take 2 Boolean values and display the result of all logical operations then return true if it is a success. | <pre> C/C++ bool logic(bool num1,bool num2) {     cout &lt;&lt; num1 &lt;&lt; " and " &lt;&lt; num2 &lt;&lt; ": " &lt;&lt; (num1 &amp;&amp; num2) &lt;&lt; endl;     cout &lt;&lt; num1 &lt;&lt; " or " &lt;&lt; num2 &lt;&lt; ": " &lt;&lt; (num1    num2) &lt;&lt; endl;     cout &lt;&lt; "Not " &lt;&lt; num1 &lt;&lt; ": " &lt;&lt; (!num1) &lt;&lt; endl;     cout &lt;&lt; "Not " &lt;&lt; num2 &lt;&lt; ": " &lt;&lt; (!num2) &lt;&lt; endl;     return true; }; </pre> |

## Main code // Driver

```
C/C++
int main() {
    cout << "-----" << endl;
    cout << "ILO A:" << endl << endl;

    //Calling sum function
    cout << "1) Sum function:" << endl;
    int n1=4, n2=6;
    int sums = sum(4,6);
    cout << n1 << " + " << n2 << " = " << sums << endl <<endl;

    //Calling the "Print the greatest number function"
    cout << "2) Printing the greatest number:" << endl;
    cout << "Between " << n1 << " and " << n2 <<" , ";
    cout << "the bigger number is ";
    cout << more(n1,n2) << endl << endl;

    //Calling Boolean function
    cout << "3) Logical operation of 2 boolean values:" << endl;
    bool True = true, False = false;
    logic(True, False);
    return 0;
}
```

## Output

```
-----
ILO A:

1) Sum function:
4 + 6 = 10

2) Printing the greatest number:
Between 4 and 6, the bigger number is 6

3) Logical operation of 2 boolean values:
1 and 0: 0
1 or 0: 1
Not 1: 0
Not 0: 1

-----
Process exited after 0.0424 seconds with return value 0
Press any key to continue . . .
```

## ILO B

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

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();
};

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);
}

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

The shape is a valid triangle.

The code given for the ILO B has an output of “The shape is valid triangle”. I believe this is because of the functions within the class. I noticed that the input set for the triangle has a sum of 180, which is the total sum of all the angles in a triangle. The code is made in a way where the input set in the main function is sent to check whether it is valid as a triangle by looking at the sum of the set.

## 7. Supplementary Activity

| Section                               | Code   |
|---------------------------------------|--|
| Header                                | <pre>C/C++ #include &lt;iostream&gt; #include &lt;cmath&gt; using namespace std;</pre>                               |
| Create a C++ code that swap 2 numbers | <pre>C/C++ void swap(int&amp; num1, int&amp; num2) {     int temp = num1;     num1 = num2;     num2 = temp; };</pre> |

|  |  |
|--|--|
|  |  |
| Create a C++ program that has a function to convert temperature in Kelvin to Fahrenheit.       | <pre> C/C++ double k_to_f(double K) {     return ((K - 273.15) * 9 / 5) + 32; }; </pre>  |
| Create a C++ program that has a function that will calculate the distance between two points.  | <pre> C/C++ double dist(double x1, double y1, double x2, double y2) {     double distance = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));     return distance; }; </pre>  |
| Create a C++ program to calculate the area and perimeter of a triangle and determine its type. | <pre> C/C++ void triangle(double a, double b, double c) {     double perimeter = a + b + c;     cout &lt;&lt; "Perimeter: " &lt;&lt; perimeter &lt;&lt; endl;     double semi = perimeter / 2;     double area = sqrt(semi*(semi - a)*(semi-b)*(semi-c));     cout &lt;&lt; "Area " &lt;&lt; area &lt;&lt; endl;     //type     double longest = max(max(a, b), c);     if (pow(longest, 2) &gt; pow(a, 2) + pow(b, 2) + pow(c, 2) - 2 * longest * max(min(a, b), min(b, c))) {         cout &lt;&lt; "It is an obtuse triangle" &lt;&lt; endl;     }     else if (pow(longest, 2) &lt; pow(a, 2) + pow(b, 2) + pow(c, 2) - 2 * longest * max(min(a, b), min(b, c))) {         cout &lt;&lt; "It is an acute triangle" &lt;&lt; endl;     }     else {         cout &lt;&lt;"This triangle is classified as 'others'" &lt;&lt; endl;     } }; </pre> |
| Main code // Driver  | <pre> C/C++ int main() {     //Swapping numbers     cout &lt;&lt; "1) Swapping values of 2 variables:" &lt;&lt; endl;     int n3 = 7, n4 = 3; </pre>   |

```

        cout << "Before swap: " << endl;
        cout << "n1: " << n3 << ", n2: " << n4 << endl;
        swap(n3,n4);
        cout << "After swap: " << endl;
        cout << "n1: " << n3 << ", n2: " << n4 << endl;
        cout << endl;

        //Temperature: Kelvin to Fahrenheit
        cout << "2) Convert Kelvin to Fahrenheit:" << endl;
        double K = 400;
        cout << "Kelvin: " << K << endl;
        cout << "Fahrenheit: " << Kfahrenheit(K);
        cout << endl << endl;

        //Distance of two points
        cout << "3) Distance of two points:" << endl;
        double x1=3, y1=4, x2=30, y2=40;
        cout << "First point: " << "(" << x1 << ", " << y1 << ")" <<
endl;
        cout << "Second point: " << "(" << x2 << ", " << y2 << ")" <<
endl;
        cout << "Distance: " << dist(x1,y1,x2,y2) << endl << endl;

        //Triangle area, perimeter, and type
        cout << "4) Triangle:" << endl;
        double s1 = 6, s2 = 3, s3 = 4;
        triangle(s1,s2,s3);
        return 0;
    }

```

Output

```

1) Swapping values of 2 variables:
Before swap:
n1: 7, n2: 3
After swap:
n1: 3, n2: 7

2) Convert Kelvin to Fahrenheit:
Kelvin: 400
Fahrenheit: 260.33

3) Distance of two points:
First point: (3, 4)
Second point: (30, 40)
Distance: 45

4) Triangle:
Perimeter: 13
Area 5.33268
It is an obtuse triangle

-----
Process exited after 0.08314 seconds with return value 0
Press any key to continue . . .

```

## 8. Conclusion

It was fun to recall the basic c++ codes with the use of functions, classes, and other variable manipulation. What I see here is the same with what we currently do in OOP, however it is really noticeable that c++ will always be different and harder, at least for me. In this activity, I managed to refresh my memory in how the syntax works and how each function behaves. In ILO A, I thought of it as a recall exercise of c++ syntax. The three functions I encoded only used basic arithmetic, basic inequality, and basic boolean conditions which use the and, or, and not logical operations. On the other hand, in the Supplementary Activity, it is a combination of those basic arithmetic, inequality, and boolean to create a more functional program. In the swapping function, I used int& for assigning the data types of the variables needed for it so that it will only address that value and not copy the variable. In the temperature conversion, this is just simply arithmetic with extra steps in which I used the formula for converting kelvin to celsius to fahrenheit. The same goes with the calculation for the distance. I assigned 4 values which represent the values for the x and y coordinate in a plane, and the distance was calculated with the formula I learned in physics. Lastly, for the triangle, instead of using the usual formula for the area with base and height, I used the one which utilizes the three sides. First the code will calculate the perimeter, then it will be divided by two for the semi-perimeter which will be used for the calculation for the area. Then to determine the type, I used the max() function from the library to determine the longest side, and followed the condition for determining if it's obtuse or acute. I also used the cmath libraries here so that the two certain functions I made will have a shorter and clearer code. Overall it was satisfying to code this.

## 9. Assessment Rubric