

Activity No. 1	
Basic C++ Programming	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed:09/09/24
Section:CPE21S4	Date Submitted:09/09/24
Name(s):Sanchez, Christan Ray R.	Instructor: Ma. Rizzete Sayo

6. Output

Header File Declaration Section

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

Global Declaration Section

```
// Global variable to keep track of the number of rectangle objects created
int count = 0;
```

Class Declaration and Method Definition Section

```
class rectangle {
private:
    double recLength, recWidth; // Private member variables for length and width
public:
    rectangle(double L, double W); // Constructor
    void setLength(double L); // Setter for length
    void setWidth(double W); // Setter for width
    double getPerimeter(); // Method to calculate the perimeter
};
```

Main Function

```
int main() {
    rectangle shape1(5, 7); // Create a rectangle object with length 5 and width 7
    std::cout << "The perimeter of the rectangle is " <<
    shape1.getPerimeter() << ".\n"; // Output the perimeter
    std::cout << count << " number of objects created."; // Output the number of objects created
    return 0;
}
```

Method Definition

```

// Constructor implementation
rectangle::rectangle(double L, double W) {
    recLength = L;
    recWidth = W;
    count++; // Increment the count of objects created
}

// Setter for length
void rectangle::setLength(double L) {
    recLength = L;
}

// Setter for width
void rectangle::setWidth(double W) {
    recWidth = W;
}

// Method to calculate the perimeter
double rectangle::getPerimeter() {
    return (2 * recLength) + (2 * recWidth); // Perimeter formula
}

```

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

```

#include<iostream>
using namespace std;

int count = 0;

class rectangle{
private:
    double recLength, recWidth;
public:
    rectangle(double L, double W);
    void setLength(double L);
    void setWidth(double W);
    double getPerimeter();
};

rectangle::rectangle(double L, double W) {
    recLength = L;
    recWidth = W;
    count++;
}

void rectangle::setLength(double L) {
    recLength = L;
}

void rectangle::setWidth(double W) {
    recWidth = W;
}

double rectangle::getPerimeter() {
    return (2*recLength) + (2*recWidth);
}

int main(){
    rectangle shape1(5, 7);
    std::cout << "The perimeter of the rectangle is " <<
    shape1.getPerimeter() << ".\n";
    std::cout << count << " number of objects created.";
    return 0;
}

```

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

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

// Global variable to keep track of the number of rectangle objects created
int count = 0;

// Class definition for a rectangle
class rectangle {
private:
    double recLength, recWidth; // Private member variables for length and width
public:
    rectangle(double L, double W); // Constructor
    void setLength(double L); // Setter for length
    void setWidth(double W); // Setter for width
    double getPerimeter(); // Method to calculate the perimeter
};

// Constructor implementation
rectangle::rectangle(double L, double W) {
    recLength = L;
    recWidth = W;
    count++; // Increment the count of objects created
}

// Setter for length
void rectangle::setLength(double L) {
    recLength = L;
}

// Setter for width
void rectangle::setWidth(double W) {
    recWidth = W;
}

// Method to calculate the perimeter
double rectangle::getPerimeter() {
    return (2 * recLength) + (2 * recWidth); // Perimeter formula
}

int main() {
    rectangle shape1(5, 7); // Create a rectangle object with length 5 and width 7
    std::cout << "The perimeter of the rectangle is " <<
    shape1.getPerimeter() << ".\n"; // Output the perimeter
    std::cout << count << " number of objects created."; // Output the number of objects created
    return 0;
}
```

Run

The perimeter of the rectangle is 24.
 1 number of objects created. The perimeter of the rectangle is 24.
 1 number of objects created. The perimeter of the rectangle is 24.
 1 number of objects created. The perimeter of the rectangle is 24.
 1 number of objects created.

7. Supplementary Activity

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

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

int main()
{
    //Create a C++ program to swap the two numbers in different variables.
    int var1,var2,var3;

    cout<<"Enter the value of first variable: ";
    cin>>var1;
    cout<<"Enter the value of second variable: ";
    cin>>var2;

    var3=var1;
    var1=var2;
    var2=var3
    ;
    cout<<"The value of a is: "<<var1<<endl;
    cout<<"The value of b is: "<<var2<<endl;

    return 0;
}
```

Run

Enter the value of first variable: 3
 Enter the value of second variable: 5
 The value of a is: 5
 The value of b is: 3

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

```

1  #include<iostream>
2  using namespace std;
3
4
5  int main()
6  {
7      /*Create a C++ program that has a function to convert temperature in Kelvin to
      Fahrenheit.*/
8      float kelvin, fahrenheit;
9      cout<<"Enter temperature in Kelvin: ";
10     cin>>kelvin;
11     fahrenheit = (kelvin - 273.15) * 9/5 + 32;
12     cout<<"Fahrenheit: "<<fahrenheit;
13
14     return 0;
15 }
16
17

```

Run

Enter temperature in Kelvin: 5
Fahrenheit: -450.67

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

```

1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5
6  int main()
7  {
8      /* Create a C++ program that has a function that will calculate the distance between
      two point.*/
9
10     int x1,x2,y1,y2;
11     float distance;
12
13     cout<<"Enter the value of x1: ";
14     cin>>x1;
15     cout<<"Enter the value of x2: ";
16     cin>>x2;
17     cout<<"Enter the value of y1: ";
18     cin>>y1;
19     cout<<"Enter the value of y2: ";
20     cin>>y2;
21
22     distance = sqrt(pow(x2-x1,2)+pow(y2-y1,2));
23     cout<<"The distance between two points is: "<<distance;
24
25     return 0;
26 }
27

```

Run

Enter the value of x1: 5
Enter the value of x2: 4
Enter the value of y1: 6
Enter the value of y2: 7
The distance between two points is: 1.41421

4. Modify the code given in ILO B and add the following functions:

a. A function to compute for the area of a triangle

```

void Area_triangle(double a, double b, double c){
    double side = (a+b+c)/2;
    double area = sqrt(side*(side-a)*(side-b)*(side-c));
    cout << "The area of the triangle is: " << area << endl;
}

```

b. A function to compute for the perimeter of a triangle

```

void Perimeter_triangle(double a, double b, double c){
    double perimeter = a+b+c;
    cout << "The perimeter of the triangle is: " << perimeter << endl;
}

```

c. A function that determines whether the triangle is acute-angled, obtuse-angled or 'others.'

```

void Type_triangle(double a, double b, double c){
    if(a == 90 || b == 90 || c == 90) {
        cout << "The triangle is a right triangle." << endl;
    } else if (a*a + b*b > c*c && b*b + c*c > a*a && a*a + c*c > b*b) {
        cout << "The triangle is an acute-angled triangle." << endl;
    } else {
        cout << "The triangle is an obtuse-angled triangle." << endl;
    }
}

```

OUTPUT:

```

void Area_triangle(double a, double b, double c){
    double side = (a+b+c)/2;
    double area = sqrt(side*(side-a)*(side-b)*(side-c));
    cout << "The area of the triangle is: " << area << endl;
}
void Perimeter_triangle(double a, double b, double c){
    double perimeter = a+b+c;
    cout << "The perimeter of the triangle is: " << perimeter << endl;
}
void Type_triangle(double a, double b, double c){
    if(a == 90 || b == 90 || c == 90) {
        cout << "The triangle is a right triangle." << endl;
    } else if (a*a + b*b > c*c && b*b + c*c > a*a && a*a + c*c > b*b) {
        cout << "The triangle is an acute-angled triangle." << endl;
    } else {
        cout << "The triangle is an obtuse-angled triangle." << endl;
    }
}

int main(){
    //driver code
    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";
    }

    double side1, side2, side3;
    cout << "Enter the length of side 1: ";
    cin >> side1;
    cout << "Enter the length of side 2: ";
    cin >> side2;
    cout << "Enter the length of side 3: ";
    cin >> side3;

    Area_triangle(side1, side2, side3);
    Perimeter_triangle(side1, side2, side3);
}

```

The shape is a valid triangle.
 Enter the length of side 1: 5
 Enter the length of side 2: 6
 Enter the length of side 3: 8
 The area of the triangle is: 14.9812
 The perimeter of the triangle is: 19
 The triangle is an obtuse-angled triangle.

8. Conclusion

Provide the following: Summary of lessons learned Analysis of the procedure Analysis of the supplementary activity
 Concluding statement / Feedback: How well did you think you did in this activity? What are your areas for improvement?

What I learned to today is the different part of structure of coding in c++ like for example the Header File Declaration Section, Global Declaration Section, Class Declaration and Method Definition Section, Main Function, Method Definition.

these sections are necessary in order for a code to function/process. i learned through trial and error as I review back the things I learned in C++. I think this activity is a good way to refresh my knowledge, I need to practice more or experiment more

9. Assessment Rubric