

Activity No. 1	
REVIEW OF C++ PROGRAMMING	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: SEPTEMBER 11, 2024
Section: CPE21S1	Date Submitted:
Name(s): GASPAR, AARON ROWEN O.	Instructor: MA'AM MARIA RIZETTE SAYO

6. Output

main.cpp x +

main.cpp > Student

```
1  #include <iostream>
2  using namespace std;
3
4  void displaySum(int num1, int num2) {
5      int sum = num1 + num2;
6      cout << "The sum of " << num1 << " and " << num2 << " is " << sum << endl;
7  }
8
9  int main() {
10     int a = 5, b = 10;
11     displaySum(a, b);
12     return 0;
13 }
```

>\_ Console x Shell +

Run

The sum of 5 and 10 is 15

main.cpp x +

main.cpp > Student

```
1  #include <iostream>
2  using namespace std;
3
4  bool isAGreaterThanB(int a, int b) {
5      return a > b;
6  }
7
8  int main() {
9      int a = 10, b = 5;
10     if (isAGreaterThanB(a, b)) {
11         cout << "Variable A is greater than Variable B" << endl;
12     } else {
13         cout << "Variable A is not greater than Variable B" << endl;
14     }
15     return 0;
16 }
```

>\_ Console x Shell +

Run

Variable A is greater than Variable B

```
main.cpp x +
main.cpp > Student
1 #include <iostream>
2 using namespace std;
3
4 bool logicalOperations(bool val1, bool val2) {
5     cout << "Logical AND: " << (val1 && val2) << endl;
6     cout << "Logical OR: " << (val1 || val2) << endl;
7     cout << "Logical NOT (val1): " << (!val1) << endl;
8     cout << "Logical NOT (val2): " << (!val2) << endl;
9     cout << "Logical XOR: " << (val1 ^ val2) << endl;
10    return true;
11 }
12
13 int main() {
14     bool val1 = true, val2 = false;
15     if (logicalOperations(val1, val2)) {
16         cout << "Logical operations were successful" << endl;
17     }
18     return 0;
19 }
```

Run

Logical AND: 0  
Logical OR: 1  
Logical NOT (val1): 0  
Logical NOT (val2): 1  
Logical XOR: 1  
Logical operations were successful

## 7. Supplementary Activity

```
main.cpp x +
main.cpp > Student
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int a = 5, b = 10, temp;
6     cout << "Before swapping: a = " << a << ", b = " << b << endl;
7     temp = a;
8     a = b;
9     b = temp;
10    cout << "After swapping: a = " << a << ", b = " << b << endl;
11    return 0;
12 }
```

Run

Before swapping: a = 5, b = 10  
After swapping: a = 10, b = 5

```
main.cpp x +
main.cpp > Student
1 #include <iostream>
2 using namespace std;
3
4 float kelvinToFahrenheit(float kelvin) {
5     return (9.0 / 5) * (kelvin - 273.15) + 32;
6 }
7
8 int main() {
9     float kelvin;
10    cout << "Enter temperature in Kelvin: ";
11    cin >> kelvin;
12    cout << "Temperature in Fahrenheit: " << kelvinToFahrenheit(kelvin) << endl;
13    return 0;
14 }
```

Run

Enter temperature in Kelvin: 25  
Temperature in Fahrenheit: -414.67

```
main.cpp x +
main.cpp > Student
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 double distance(double x1, double y1, double x2, double y2) {
6     return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));
7 }
8
9 int main() {
10     double x1, y1, x2, y2;
11     cout << "Enter coordinates of the first point (x1 y1): ";
12     cin >> x1 >> y1;
13     cout << "Enter coordinates of the second point (x2 y2): ";
14     cin >> x2 >> y2;
15     cout << "Distance between the points: " << distance(x1, y1, x2, y2) << endl;
16     return 0;
17 }
```

Run

Enter coordinates of the first point (x1 y1): 34 56  
Enter coordinates of the second point (x2 y2): 7 8  
Distance between the points: 55.0727

```
main.cpp x +
main.cpp > Student
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4 class Triangle{
5 private:
6 double totalAngle, angleA, angleB, angleC;
7 public:
8 Triangle(double A, double B, double C);
9 void setAngles(double A, double B, double C);
10 const bool validateTriangle();
11 };
12
13 float areaOfTriangle(float a, float b, float c) {
14     float s = (a + b + c) / 2;
15     return sqrt(s * (s - a) * (s - b) * (s - c));
16 }
17
18 Triangle::Triangle(double A, double B, double C) {
19     angleA = A;
20     angleB = B;
21     angleC = C;
22     totalAngle = A+B+C;
23 }
24 void Triangle::setAngles(double A, double B, double C) {
25     angleA = A;
26     angleB = B;
27     angleC = C;
28     totalAngle = A+B+C;
29 }
30 const bool Triangle::validateTriangle() {
31     return (totalAngle <= 180);
32 }
33
34 int main() {
35     float a, b, c;
36     cout << "Enter the sides of the triangle: ";
37     cin >> a >> b >> c;
38
39     cout << "Area of the triangle: " << areaOfTriangle(a, b, c) << endl;
40     return 0;
41 }
```

Run

Enter the sides of the triangle: 12 13 14  
Area of the triangle: 72.3879

```
main.cpp x +
main.cpp > Student
6 double totalAngle, angleA, angleB, angleC;
7 public:
8 Triangle(double A, double B, double C);
9 void setAngles(double A, double B, double C);
10 const bool validateTriangle();
11 };
12
13 float areaOfTriangle(float a, float b, float c) {
14     float s = (a + b + c) / 2;
15     return sqrt(s * (s - a) * (s - b) * (s - c));
16 }
17
18 float perimeterOfTriangle(float a, float b, float c) {
19     return a + b + c;
20 }
21
22 Triangle::Triangle(double A, double B, double C) {
23     angleA = A;
24     angleB = B;
25     angleC = C;
26     totalAngle = A+B+C;
27 }
28 void Triangle::setAngles(double A, double B, double C) {
29     angleA = A;
30     angleB = B;
31     angleC = C;
32     totalAngle = A+B+C;
33 }
34 const bool Triangle::validateTriangle() {
35     return (totalAngle <= 180);
36 }
37
38 int main() {
39     float a, b, c;
40     cout << "Enter the sides of the triangle: ";
41     cin >> a >> b >> c;
42
43     cout << "Area of the triangle: " << areaOfTriangle(a, b, c) << endl;
44     cout << "Perimeter of the triangle: " << perimeterOfTriangle(a, b, c) << endl;
45     return 0;
46 }
```

Run

Enter the sides of the triangle: 13 14 15  
Area of the triangle: 84  
Perimeter of the triangle: 42

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4 class Triangle{
5 private:
6 double totalAngle, angleA, angleB, angleC;
7 public:
8 Triangle(double A, double B, double C);
9 void setAngles(double A, double B, double C);
10 const bool validateTriangle();
11 };
12
13 float areaOfTriangle(float a, float b, float c) {
14     float s = (a + b + c) / 2;
15     return sqrt(s * (s - a) * (s - b) * (s - c));
16 }
17
18 float perimeterOfTriangle(float a, float b, float c) {
19     return a + b + c;
20 }
21
22 string typeOfTriangle(float a, float b, float c) {
23     float a2 = a * a, b2 = b * b, c2 = c * c;
24     if (a2 + b2 > c2 && a2 + c2 > b2 && b2 + c2 > a2) {
25         return "Acute-angled";
26     } else if (a2 + b2 == c2 || a2 + c2 == b2 || b2 + c2 == a2) {
27         return "Right-angled";
28     } else {
29         return "Obtuse-angled";
30     }
31 }
32
33 Triangle::Triangle(double A, double B, double C) {
34     angleA = A;
35     angleB = B;
36     angleC = C;
37     totalAngle = A+B+C;
38 }
39
40 void Triangle::setAngles(double A, double B, double C) {
41     angleA = A;
42     angleB = B;
43     angleC = C;
44     totalAngle = A+B+C;
45 }
46
47 const bool Triangle::validateTriangle() {
48     return (totalAngle <= 180);
49 }
50
51 int main() {
52     float a, b, c;
53     cout << "Enter the sides of the triangle: ";
54     cin >> a >> b >> c;
55
56     cout << "Area of the triangle: " << areaOfTriangle(a, b, c) << endl;
57     cout << "Perimeter of the triangle: " << perimeterOfTriangle(a, b, c) << endl;
58     cout << "Type of the triangle: " << typeOfTriangle(a, b, c) << endl;
59     return 0;
60 }
```

Run

Enter the sides of the triangle: 13 14 15  
Area of the triangle: 84  
Perimeter of the triangle: 42  
Type of the triangle: Acute-angled

8. Conclusion

In conclusion, By mastering these concepts, you have gained valuable skills that are essential for solving a wide range of programming problems. Keep practicing and experimenting with these techniques to deepen your understanding and improve your coding proficiency.

9. Assessment Rubric