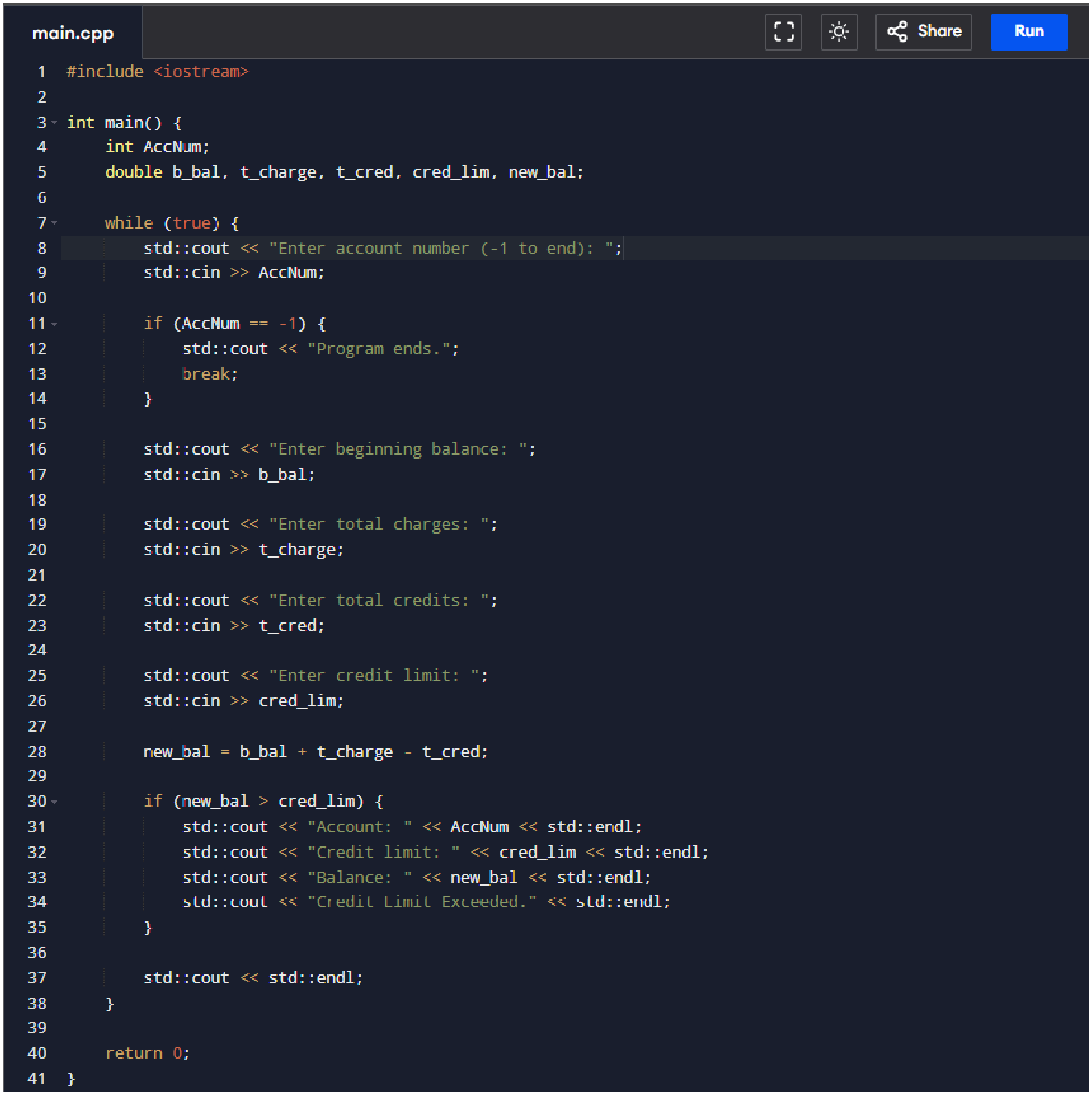
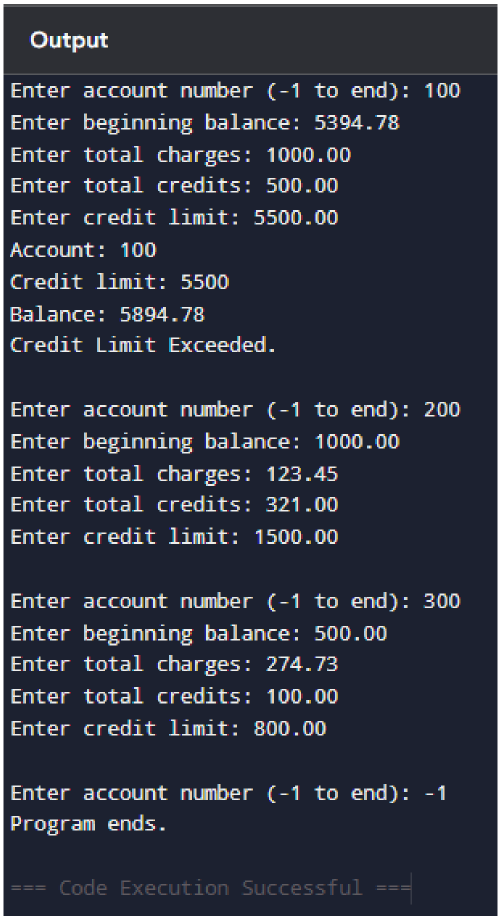
Activity No. 3.1	
Control Structures (pt. 2)	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed: August 16, 2025**
Section: CPE11S1	**Date Submitted: August 18, 2025**
Name(s): Lopez, Andrei Dion C.	**Instructor: Sir Jimlord**
6. Output	

Exercise 1:



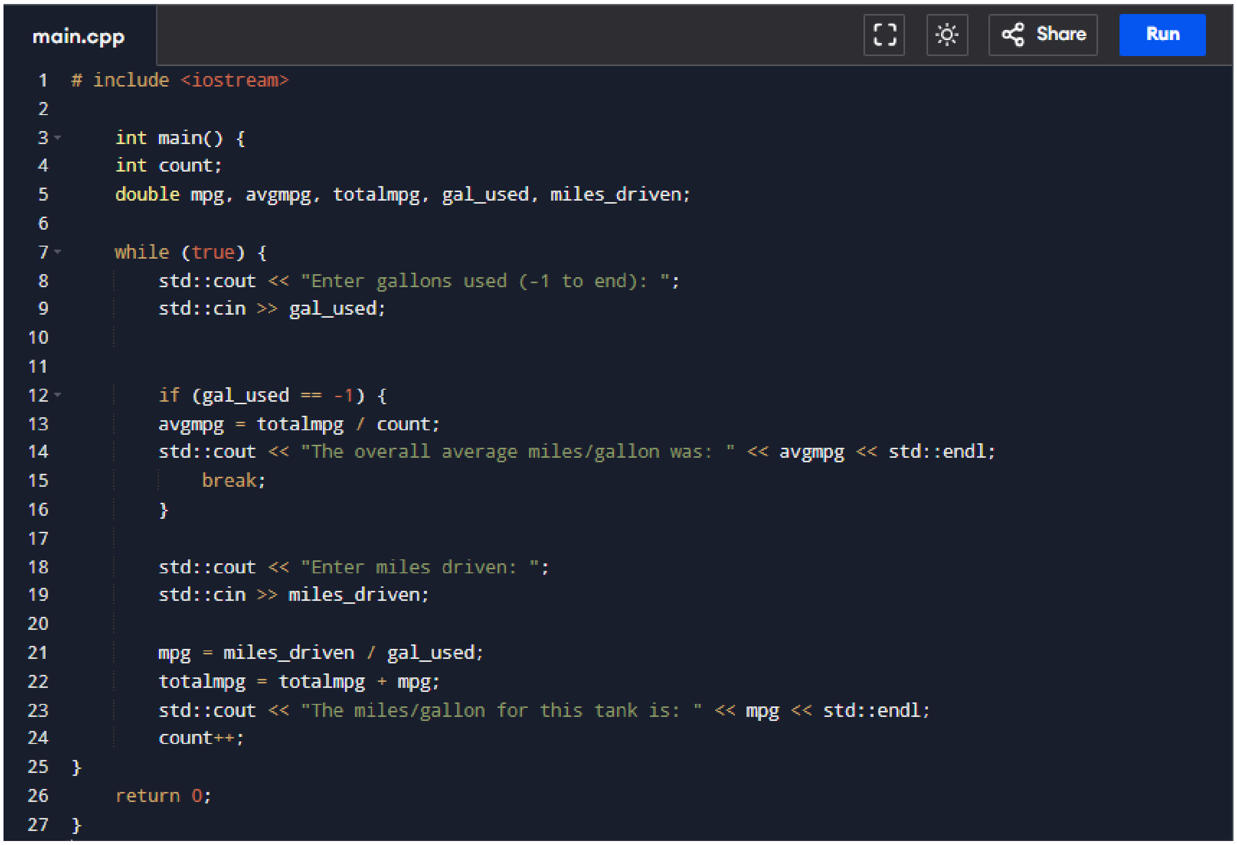
```cpp
#include <iostream>

int main() {
    int AccNum;
    double b_bal, t_charge, t_cred, cred_lim, new_bal;

    while (true) {
        std::cout << "Enter account number (-1 to end): ";
        std::cin >> AccNum;

        if (AccNum == -1) {
            std::cout << "Program ends.";
            break;
        }

        std::cout << "Enter beginning balance: ";
        std::cin >> b_bal;

        std::cout << "Enter total charges: ";
        std::cin >> t_charge;

        std::cout << "Enter total credits: ";
        std::cin >> t_cred;

        std::cout << "Enter credit limit: ";
        std::cin >> cred_lim;

        new_bal = b_bal + t_charge - t_cred;

        if (new_bal > cred_lim) {
            std::cout << "Account: " << AccNum << std::endl;
            std::cout << "Credit limit: " << cred_lim << std::endl;
            std::cout << "Balance: " << new_bal << std::endl;
            std::cout << "Credit Limit Exceeded." << std::endl;
        }

        std::cout << std::endl;
    }

    return 0;
}
```

Output:



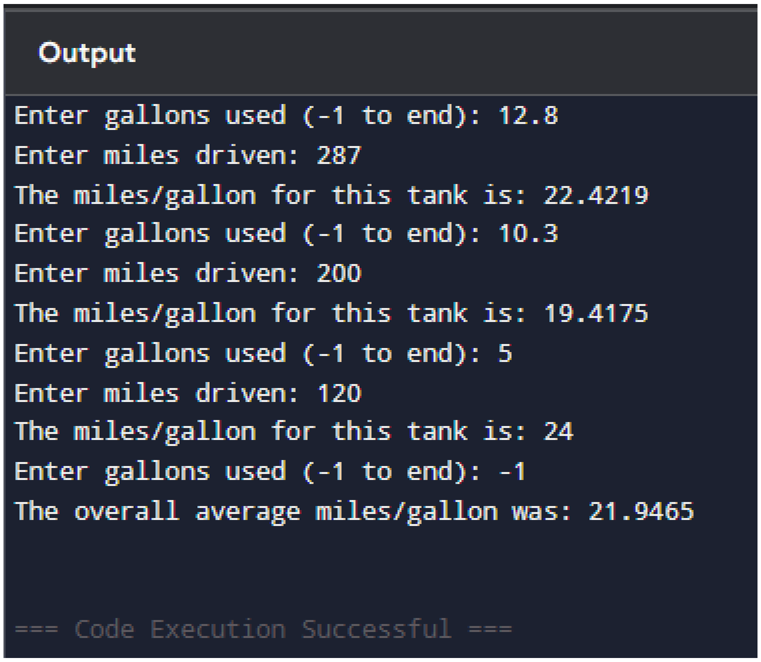
```
Enter account number (-1 to end): 100
Enter beginning balance: 5394.78
Enter total charges: 1000.00
Enter total credits: 500.00
Enter credit limit: 5500.00
Account: 100
Credit limit: 5500
Balance: 5894.78
Credit Limit Exceeded.

Enter account number (-1 to end): 200
Enter beginning balance: 1000.00
Enter total charges: 123.45
Enter total credits: 321.00
Enter credit limit: 1500.00

Enter account number (-1 to end): 300
Enter beginning balance: 500.00
Enter total charges: 274.73
Enter total credits: 100.00
Enter credit limit: 800.00

Enter account number (-1 to end): -1
Program ends.

=== Code Execution Successful ===
```

Exercise 2:



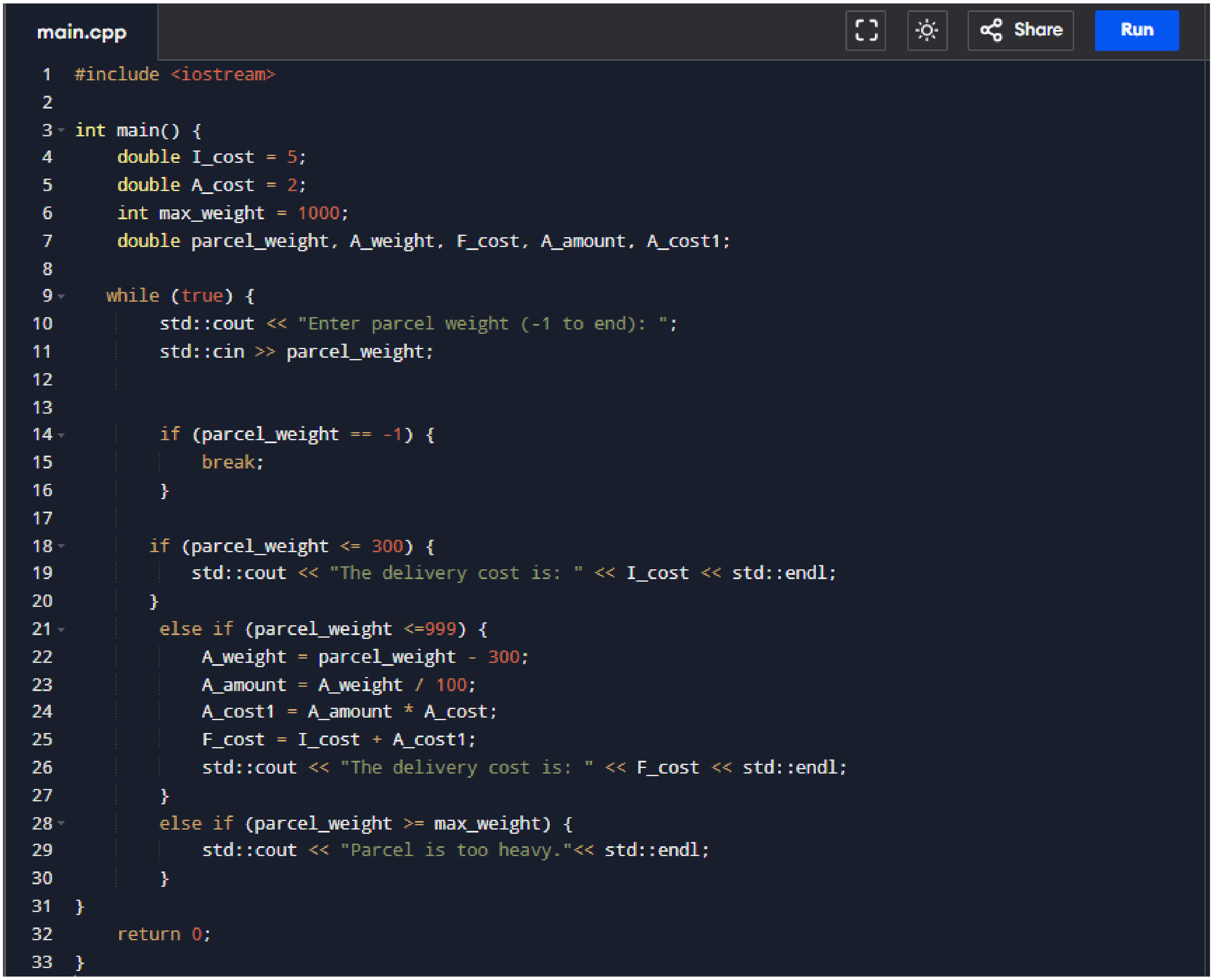
```cpp
# include <iostream>

    int main() {
    int count;
    double mpg, avgmpg, totalmpg, gal_used, miles_driven;

    while (true) {
        std::cout << "Enter gallons used (-1 to end): ";
        std::cin >> gal_used;


        if (gal_used == -1) {
        avgmpg = totalmpg / count;
        std::cout << "The overall average miles/gallon was: " << avgmpg << std::endl;
            break;
        }

        std::cout << "Enter miles driven: ";
        std::cin >> miles_driven;

        mpg = miles_driven / gal_used;
        totalmpg = totalmpg + mpg;
        std::cout << "The miles/gallon for this tank is: " << mpg << std::endl;
        count++;
}
    return 0;
}
```

Output:



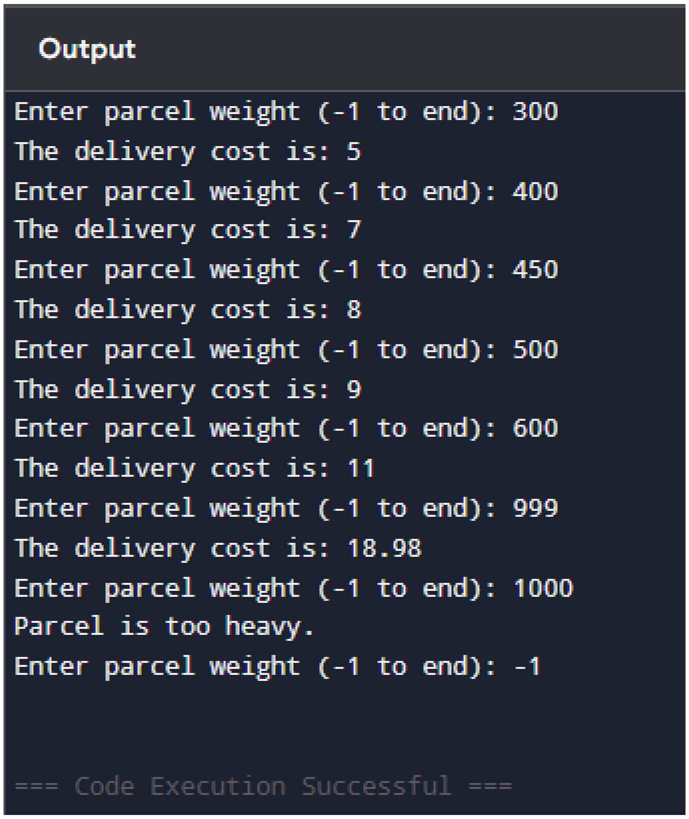
```
Enter gallons used (-1 to end): 12.8
Enter miles driven: 287
The miles/gallon for this tank is: 22.4219
Enter gallons used (-1 to end): 10.3
Enter miles driven: 200
The miles/gallon for this tank is: 19.4175
Enter gallons used (-1 to end): 5
Enter miles driven: 120
The miles/gallon for this tank is: 24
Enter gallons used (-1 to end): -1
The overall average miles/gallon was: 21.9465


=== Code Execution Successful ===
```

Exercise 3:



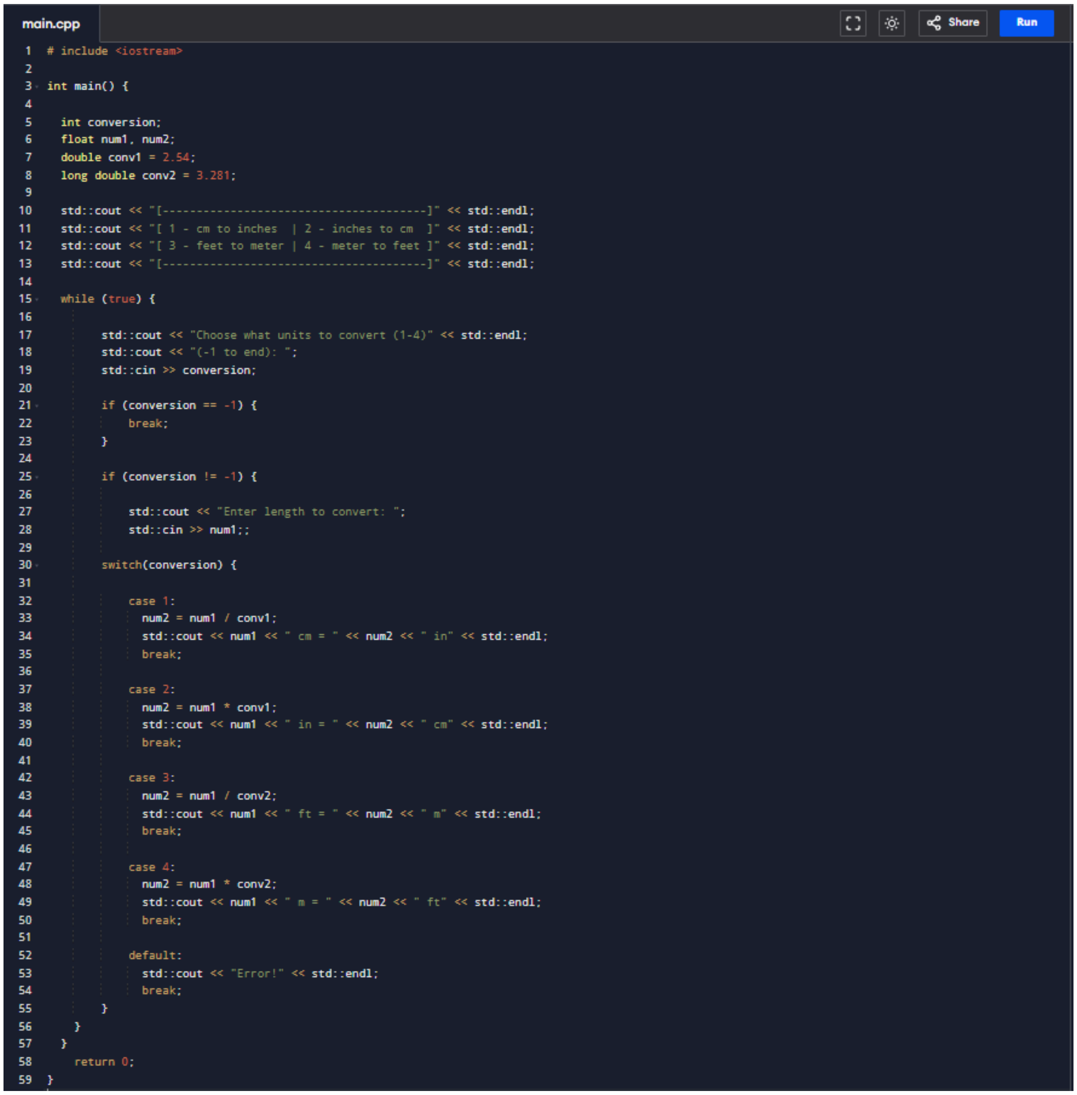
```cpp
#include <iostream>

int main() {
    double I_cost = 5;
    double A_cost = 2;
    int max_weight = 1000;
    double parcel_weight, A_weight, F_cost, A_amount, A_cost1;

   while (true) {
        std::cout << "Enter parcel weight (-1 to end): ";
        std::cin >> parcel_weight;


        if (parcel_weight == -1) {
            break;
        }

       if (parcel_weight <= 300) {
           std::cout << "The delivery cost is: " << I_cost << std::endl;
       }
        else if (parcel_weight <=999) {
            A_weight = parcel_weight - 300;
            A_amount = A_weight / 100;
            A_cost1 = A_amount * A_cost;
            F_cost = I_cost + A_cost1;
            std::cout << "The delivery cost is: " << F_cost << std::endl;
        }
        else if (parcel_weight >= max_weight) {
            std::cout << "Parcel is too heavy."<< std::endl;
        }
}
    return 0;
}
```

Output:



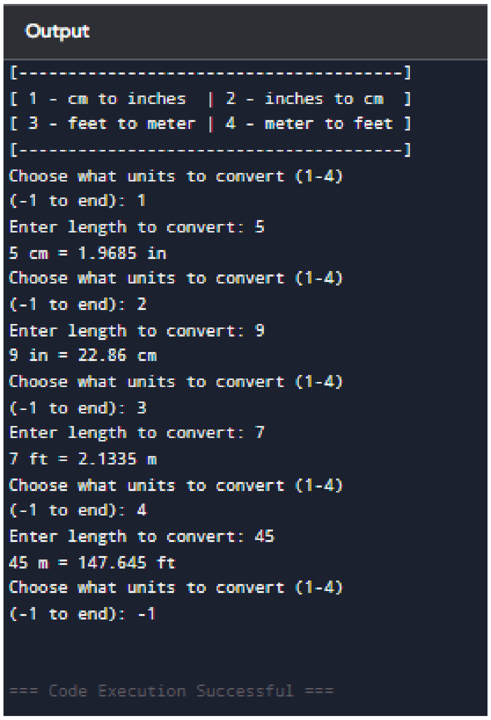
```
Enter parcel weight (-1 to end): 300
The delivery cost is: 5
Enter parcel weight (-1 to end): 400
The delivery cost is: 7
Enter parcel weight (-1 to end): 450
The delivery cost is: 8
Enter parcel weight (-1 to end): 500
The delivery cost is: 9
Enter parcel weight (-1 to end): 600
The delivery cost is: 11
Enter parcel weight (-1 to end): 999
The delivery cost is: 18.98
Enter parcel weight (-1 to end): 1000
Parcel is too heavy.
Enter parcel weight (-1 to end): -1


=== Code Execution Successful ===
```

Exercise 4.



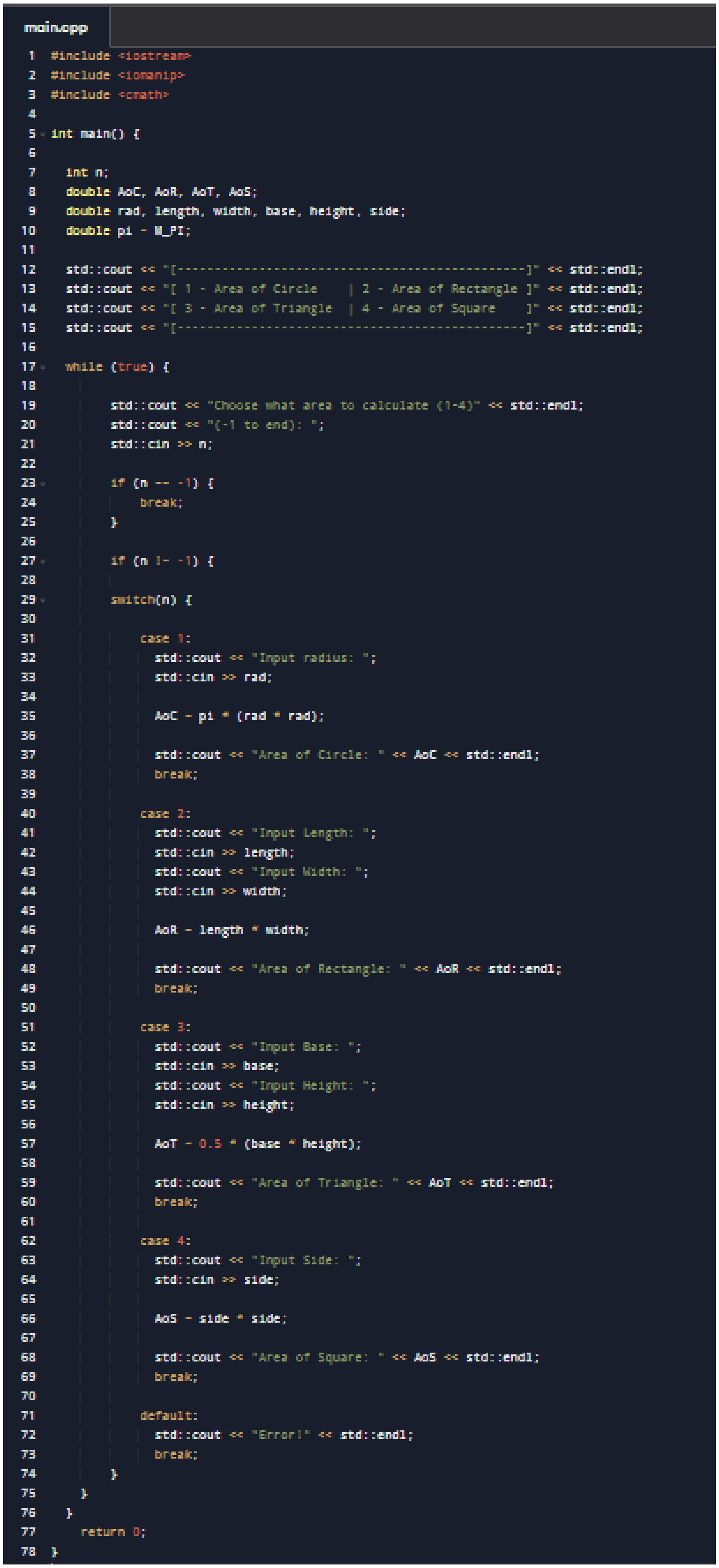
```cpp
# include <iostream>

int main() {

  int conversion;
  float num1, num2;
  double conv1 = 2.54;
  long double conv2 = 3.281;

  std::cout << "[---------------------------------------]" << std::endl;
  std::cout << "[ 1 - cm to inches  | 2 - inches to cm  ]" << std::endl;
  std::cout << "[ 3 - feet to meter | 4 - meter to feet ]" << std::endl;
  std::cout << "[---------------------------------------]" << std::endl;

  while (true) {

        std::cout << "Choose what units to convert (1-4)" << std::endl;
        std::cout << "(-1 to end): ";
        std::cin >> conversion;

        if (conversion == -1) {
            break;
        }

        if (conversion != -1) {

            std::cout << "Enter length to convert: ";
            std::cin >> num1;;

        switch(conversion) {

            case 1:
              num2 = num1 / conv1;
              std::cout << num1 << " cm = " << num2 << " in" << std::endl;
              break;

            case 2:
              num2 = num1 * conv1;
              std::cout << num1 << " in = " << num2 << " cm" << std::endl;
              break;

            case 3:
              num2 = num1 / conv2;
              std::cout << num1 << " ft = " << num2 << " m" << std::endl;
              break;

            case 4:
              num2 = num1 * conv2;
              std::cout << num1 << " m = " << num2 << " ft" << std::endl;
              break;

            default:
              std::cout << "Error!" << std::endl;
              break;
        }
    }
  }
    return 0;
}
```

Output:



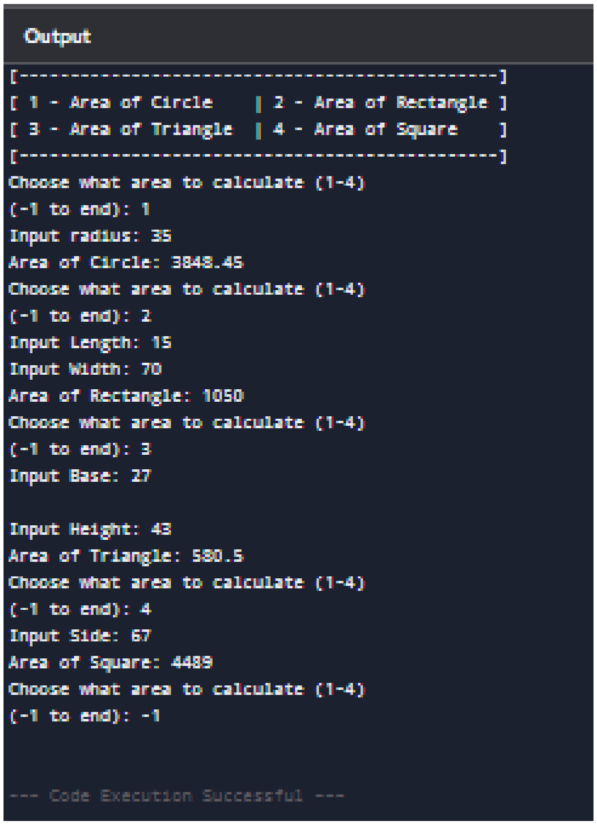
```
[---------------------------------------]
[ 1 - cm to inches  | 2 - inches to cm  ]
[ 3 - feet to meter | 4 - meter to feet ]
[---------------------------------------]
Choose what units to convert (1-4)
(-1 to end): 1
Enter length to convert: 5
5 cm = 1.9685 in
Choose what units to convert (1-4)
(-1 to end): 2
Enter length to convert: 9
9 in = 22.86 cm
Choose what units to convert (1-4)
(-1 to end): 3
Enter length to convert: 7
7 ft = 2.1335 m
Choose what units to convert (1-4)
(-1 to end): 4
Enter length to convert: 45
45 m = 147.645 ft
Choose what units to convert (1-4)
(-1 to end): -1


=== Code Execution Successful ===
```

Exercise 5.



```cpp
#include <iostream>
#include <iomanip>
#include <cmath>

int main() {

  int n;
  double AoC, AoR, AoT, AoS;
  double rad, length, width, base, height, side;
  double pi = M_PI;

  std::cout << "[-----------------------------------------------]" << std::endl;
  std::cout << "[ 1 - Area of Circle    | 2 - Area of Rectangle ]" << std::endl;
  std::cout << "[ 3 - Area of Triangle  | 4 - Area of Square    ]" << std::endl;
  std::cout << "[-----------------------------------------------]" << std::endl;

  while (true) {

        std::cout << "Choose what area to calculate (1-4)" << std::endl;
        std::cout << "(-1 to end): ";
        std::cin >> n;

        if (n == -1) {
            break;
        }

        if (n != -1) {

        switch(n) {

            case 1:
              std::cout << "Input radius: ";
              std::cin >> rad;

              AoC = pi * (rad * rad);

              std::cout << "Area of Circle: " << AoC << std::endl;
              break;

            case 2:
              std::cout << "Input Length: ";
              std::cin >> length;
              std::cout << "Input Width: ";
              std::cin >> width;

              AoR = length * width;

              std::cout << "Area of Rectangle: " << AoR << std::endl;
              break;

            case 3:
              std::cout << "Input Base: ";
              std::cin >> base;
              std::cout << "Input Height: ";
              std::cin >> height;

              AoT = 0.5 * (base * height);

              std::cout << "Area of Triangle: " << AoT << std::endl;
              break;

            case 4:
              std::cout << "Input Side: ";
              std::cin >> side;

              AoS = side * side;

              std::cout << "Area of Square: " << AoS << std::endl;
              break;

            default:
              std::cout << "Error!" << std::endl;
              break;
        }
    }
  }
    return 0;
}
```

Output



```
[-----------------------------------------------]
[ 1 - Area of Circle    | 2 - Area of Rectangle ]
[ 3 - Area of Triangle  | 4 - Area of Square    ]
[-----------------------------------------------]
Choose what area to calculate (1-4)
(-1 to end): 1
Input radius: 35
Area of Circle: 3848.45
Choose what area to calculate (1-4)
(-1 to end): 2
Input Length: 15
Input Width: 70
Area of Rectangle: 1050
Choose what area to calculate (1-4)
(-1 to end): 3
Input Base: 27

Input Height: 43
Area of Triangle: 580.5
Choose what area to calculate (1-4)
(-1 to end): 4
Input Side: 67
Area of Square: 4489
Choose what area to calculate (1-4)
(-1 to end): -1


--- Code Execution Successful ---
```

7. Supplementary Activity

8. Conclusion

I learnt how to create programs using switch structures, how to use conditions with the while structure, if/else structures, and how to use sentinel and counter controlled repetition

9. Assessment Rubric