

Hands-on Activity 2.1	
Control Structures	
Course Code: CPE007	Program: Computer Engineering
Course Title: Programming Logic and Design	Date Performed: 08/6/2025
Section: CPE11S1	Date Submitted: 08/6/2025
Name(s): Cenar, Marqui Joshua	Instructor: Engr. Jimlord M. Quejado
6. Output	
<pre> 1. Start Input accountNumber While accountNumber is not -1 Input beginningBalance Input totalCharges Input totalCredits Input creditLimit Set newBalance = beginningBalance + totalCharges - totalCredits If newBalance > creditLimit then Output "Account: ", accountNumber Output "Credit limit: ", creditLimit Output "Balance: ", newBalance Output "Credit Limit Exceeded." Else Output "Account: ", accountNumber Output "Credit limit: ", creditLimit Output "Balance: ", newBalance Output "Within credit limit." End if End while Input accountNumber Output "Program ends." Stop </pre>	

```
main.cpp
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main() {
6     int accountNumber;
7     float beginningBalance;
8     float totalCharges;
9     float totalCredits;
10    float creditLimit;
11    float newBalance;
12
13    cout << fixed << setprecision(2);
14
15    cout << "Enter account number (-1 to stop):";
16    cin >> accountNumber;
17
18    while (accountNumber != -1) {
19        cout << "Enter beginning balance: ";
20        cin >> beginningBalance;
21
22        cout << "Enter total charges: ";
23        cin >> totalCharges;
24
25        cout << "Enter total credits: ";
26        cin >> totalCredits;
27
28        cout << "Enter credit limit: ";
29        cin >> creditLimit;
30    }
```

Output

```
Account: 100
Credit limit: 100000.00
Balance: 6752945.00
Credit Limit Exceeded.

Enter account number (-1 to stop): 200
Enter beginning balance: 78364873
Enter total charges: 400
Enter total credits: 500
Enter credit limit: 5500
Account: 200
Credit limit: 5500.00
Balance: 78364768.00
Credit Limit Exceeded.

Enter account number (-1 to stop): 330
Enter beginning balance: 79875
Enter total charges: 777
Enter total credits: 444
Enter credit limit: 90000
Account: 330
Credit limit: 90000.00
Balance: 80208.00
Within credit limit.

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

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

2.

Start

Set totalMiles = 0

Set totalGallons = 0

Output "Enter the gallons used (-1 to end): "

Input gallonsUsed

While gallonsUsed is not equal to -1

Output "Enter the miles driven: "

Input milesDriven

Set milesPerGallon = milesDriven / gallonsUsed

Output "The miles/gallon for this tank was ", milesPerGallon

totalMiles = totalMiles + milesDriven

totalGallons = totalGallons + gallonsUsed

Output "Enter the gallons used (-1 to end): "

Input gallonsUsed

End while

If totalGallons > 0 then

Set overallAverage = totalMiles / totalGallons

Output "The overall average miles/gallon was ", overallAverage

End if

Stop

```
main.cpp
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main() {
6     float gallonsUsed;
7     float milesDriven;
8     float milesPerGallon;
9     float totalMiles = 0;
10    float totalGallons = 0;
11
12    cout << fixed << setprecision(6);
13
14    cout << "Enter the gallons used (-1 to end):";
15    cin >> gallonsUsed;
16
17    while (gallonsUsed != -1) {
18        cout << "Enter the miles driven: ";
19        cin >> milesDriven;
20
21        milesPerGallon = milesDriven / gallonsUsed;
22        cout << "The miles / gallon for this tank was " <<
            milesPerGallon << endl;
23
24        totalMiles += milesDriven;
25        totalGallons += gallonsUsed;
26
27        cout << "\nEnter the gallons used (-1 to end): ";
28        cin >> gallonsUsed;
29    }
```

Output

```
Enter the gallons used (-1 to end):15
Enter the miles driven: 300
The miles / gallon for this tank was 20.000000

Enter the gallons used (-1 to end): 10
Enter the miles driven: 30
The miles / gallon for this tank was 3.000000

Enter the gallons used (-1 to end): -1

The overall average miles/gallon was13.200000















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

3.
Start
Input Weight
If weight > 1000 then
 Output "Parcel exceeds maximum weight limit."
Else
 If weight <= 100 then
 cost ← 2.00
 Else if weight <= 300 then
 cost ← 5.00
 Else
 cost ← 5.00
 extraWeight ← weight - 300
 extraUnits ← extraWeight / 100




 If extraWeight mod 100 ≠ 0 then
 extraUnits ← extraUnits + 1
 End if
 cost ← cost + (extraUnits * 2.00)

 End if

 Output "The cost of sending the parcel is: P", cost
End if
Stop



main.cpp



Run

Output

Clear

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main() {
6     int weight;
7     float cost;
8
9     cout << fixed << setprecision(2);
10
11     cout << "Enter weight of parcel in grams:";
12     cin >> weight;
13
14     if (weight > 100) {
15         cout << "Parcel exceeds maximum weight limit." << endl;
16     } else {
17         if (weight <= 100) {
18             cost = 2.00;
19         }
20         else if (weight <= 300) {
21             cost = 5.00;
22         }
23         else {
24             cost = 5.00;
25             int extremeweight = weight - 300;
26             int extraunits = extremeweight / 100;
27
28             if (extremeweight % 100 != 0) {
29                 extraunits += 1;
30             }
31         }
32     }
33 }
```

Enter weight of parcel in grams:100
The cost of sending the parcel is: P2.00

=== Code Execution Successful ===

Output

Enter weight of parcel in grams:300
The cost of sending the parcel is: P5.00

=== Code Execution Successful ===

Output

Enter weight of parcel in grams:1000
The cost of sending the parcel is: P7.00

=== Code Execution Successful ===

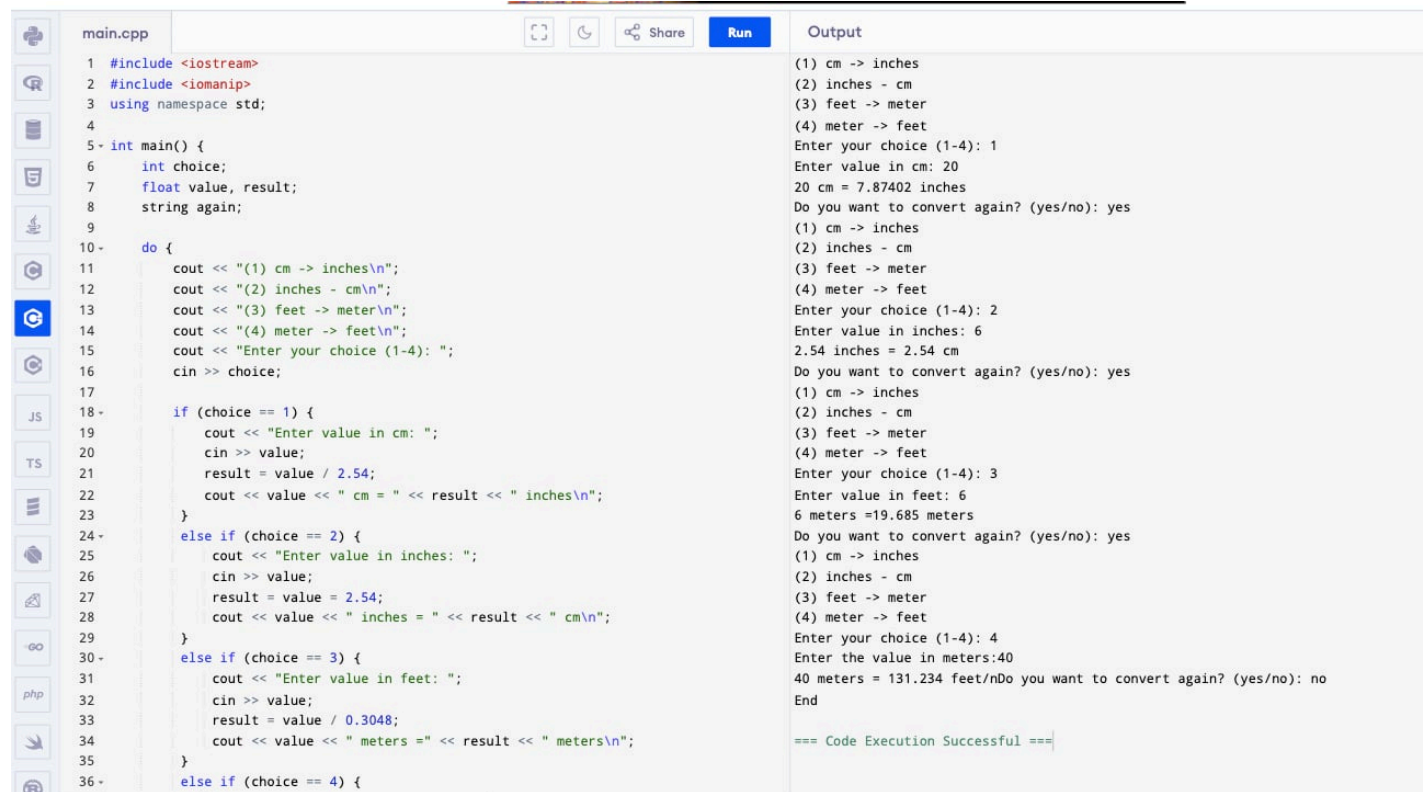
Output

```
Enter weight of parcel in grams:20202020
Parcel exceeds maximum weight limit.
```

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

```
4.
Start
do
output "(1) cm -> inches"
output "(2) inches -> cm"
output "(3) feet -> meter"
output "(4) meter -> feet"
output "Enter your choice (1-4): "
input choice
if choice == 1 then
output "Enter value in cm: "
input value
result = value / 2.54
output value, " cm = ", result, " inches"
else if choice == 2 then
output "Enter value in inches: "
input value
result = value * 2.54
output value, " inches = ", result, " cm"
else if choice == 3 then
output "Enter value in feet: "
input value
result = value * 0.3048
output value, " feet = ", result, " meters"
else if choice == 4 then
output "Enter value in meters: "
input value
result = value / 0.3048 output value, " meters = ", result, " feet"
else
output "Invalid choice. Please select 1-4."
end if
output "Do you want to convert again? (yes/no): "
input again
```

while again == "yes" or again == "Yes"
output "End"
stop



The screenshot shows a C++ code editor with a file named `main.cpp`. The code implements a unit conversion program. It includes `<iostream>` and `<iomanip>`, uses the `std` namespace, and defines a `main` function. Inside `main`, it declares `int choice`, `float value`, `float result`, and `string again`. A `do` loop starts with a menu of four options: (1) cm to inches, (2) inches to cm, (3) feet to meter, and (4) meter to feet. The user enters choice 1, then 20 cm, resulting in 7.87402 inches. The user enters 'yes' to convert again. Then choice 2 is entered, 2.54 inches are entered, resulting in 2.54 cm. The user enters 'yes' again. Then choice 3 is entered, 6 feet are entered, resulting in 19.685 meters. The user enters 'yes' again. Then choice 4 is entered, 40 meters are entered, resulting in 131.234 feet. The user enters 'no' to stop. The program ends with the message "Code Execution Successful".

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main() {
6     int choice;
7     float value, result;
8     string again;
9
10    do {
11        cout << "(1) cm -> inches\n";
12        cout << "(2) inches - cm\n";
13        cout << "(3) feet -> meter\n";
14        cout << "(4) meter -> feet\n";
15        cout << "Enter your choice (1-4): ";
16        cin >> choice;
17
18        if (choice == 1) {
19            cout << "Enter value in cm: ";
20            cin >> value;
21            result = value / 2.54;
22            cout << value << " cm = " << result << " inches\n";
23        }
24        else if (choice == 2) {
25            cout << "Enter value in inches: ";
26            cin >> value;
27            result = value * 2.54;
28            cout << value << " inches = " << result << " cm\n";
29        }
30        else if (choice == 3) {
31            cout << "Enter value in feet: ";
32            cin >> value;
33            result = value * 0.3048;
34            cout << value << " feet = " << result << " meters\n";
35        }
36        else if (choice == 4) {
37            cout << "Enter value in meters: ";
38            cin >> value;
39            result = value * 3.28084;
40            cout << value << " meters = " << result << " feet\n";
41        }
42        cout << "Do you want to convert again? (yes/no): ";
43        cin >> again;
44    } while (again == "yes" || again == "Yes");
45    cout << "End\n";
46    return 0;
47 }
```

Output:

```
(1) cm -> inches
(2) inches - cm
(3) feet -> meter
(4) meter -> feet
Enter your choice (1-4): 1
Enter value in cm: 20
20 cm = 7.87402 inches
Do you want to convert again? (yes/no): yes
(1) cm -> inches
(2) inches - cm
(3) feet -> meter
(4) meter -> feet
Enter your choice (1-4): 2
Enter value in inches: 2.54
2.54 inches = 2.54 cm
Do you want to convert again? (yes/no): yes
(1) cm -> inches
(2) inches - cm
(3) feet -> meter
(4) meter -> feet
Enter your choice (1-4): 3
Enter value in feet: 6
6 meters = 19.685 meters
Do you want to convert again? (yes/no): yes
(1) cm -> inches
(2) inches - cm
(3) feet -> meter
(4) meter -> feet
Enter your choice (1-4): 4
Enter the value in meters: 40
40 meters = 131.234 feet
Do you want to convert again? (yes/no): no
End
=== Code Execution Successful ===
```

5.
Start
Repeat
Output "(1) Area of Circle"
Output "(2) Area of Rectangle"
Output "(3) Area of Triangle"
Output "(4) Area of Square"
Output "Enter your choice (1-4): "
Input choice

If choice = 1 then
Output "Enter radius: "
Input radius
 $area = 3.1416 * radius * radius$
Output "Area of Circle = ", area
Else if choice = 2 then
Output "Enter length: "
Input length
Output "Enter width: "
Input width
 $area = length * width$
Output "Area of Rectangle = ", area

Else if choice = 3 then
 Output "Enter base: "
 Input base
 Output "Enter height: "
 Input height
 $area = 0.5 * base * height$
 Output "Area of Triangle = ", area
 Else if choice = 4 then
 Output "Enter side: "
 Input side
 $area = side * side$
 Output "Area of Square = ", area
 Else
 Output "Invalid choice. Please select 1-4."
 End if

Output "Do you want to compute again? (yes/no): "
 Input again
 Until again \neq "yes" and again \neq "Yes"

Output "End"
 Stop

The screenshot shows a C++ IDE with a file named `main.cpp`. The code implements a program to calculate the area of different shapes based on user input. The output window shows the program's execution, including prompts for shape choice, dimensions, and the calculated area, ending with a successful execution message.

```

1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5- int main() {
6    int choice;
7    float radius, lenght, width, base, height, side, area;
8    string again;
9
10-   do {
11       cout << "(1) Area of Circle\n";
12       cout << "(2) Area of Rectangle\n";
13       cout << "(3) Area of Triangle\n";
14       cout << "(4) Area of Square\n";
15       cout << "Enter your choice (1-4): ";
16       cin >> choice;
17
18-       if (choice == 1) {
19           cout << "Enter radius: ";
20           cin >> radius;
21           area = 3.1416 * radius * radius;
22           cout << "Area of Circle = " << area << endl;
23       }
24-       else if (choice == 2) {
25           cout << "Enter length: ";
26           cin >> lenght;
27           cout << "Enter width: ";
28           cin >> width;
29           area = lenght * width;
30           cout << "Area of Rectangle = " << area << endl;
31       }
32-       else if (choice == 3) {
33           cout << "Enter base: ";
34           cin >> base;
35           cout << "Enter height: ";
36           cin >> height;
37           area = 0.5 * base * height;
38           cout << "Area of Triangle = " << area << endl;
39       }
40-       else if (choice == 4) {
41           cout << "Enter side: ";
42           cin >> side;
43           area = side * side;
44           cout << "Area of Square = " << area << endl;
45       }
  
```

Output:

```

(1) Area of Circle
(2) Area of Rectangle
(3) Area of Triangle
(4) Area of Square
Enter your choice (1-4): 1
Enter radius: 10
Area of Circle = 314.16

Do you want to compute again? (yes/no): yes
(1) Area of Circle
(2) Area of Rectangle
(3) Area of Triangle
(4) Area of Square
Enter your choice (1-4): 2
Enter length: 20
Enter width: 30
Area of Rectangle = 600

Do you want to compute again? (yes/no): yes
(1) Area of Circle
(2) Area of Rectangle
(3) Area of Triangle
(4) Area of Square
Enter your choice (1-4): 3
Enter base: 66
Enter height: 90
Area of Triangle = 2970

Do you want to compute again? (yes/no): yes
(1) Area of Circle
(2) Area of Rectangle
(3) Area of Triangle
(4) Area of Square
Enter your choice (1-4): 4
Enter side: 30
Area of Square = 900

Do you want to compute again? (yes/no): no
End





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


7. Supplementary Activity

8. Conclusion

I learned while typing the pseudo codes how to arrange and rearrange certain steps to make a program. While it was tough making this, I still learned something in the process

9. Assessment Rubric

Rubric for SO 7 (8)							
Criteria	Ratings						Pts
 SO 7 PI 1 ILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Educational interests and pursuits exist and flourish outside classroom requirements, knowledge and/or experiences are pursued independently and applies knowledge learned into practice	5 pts Good Educational interests and pursuits exist and flourish outside classroom requirements, knowledge and/or experiences are pursued independently	4 pts Satisfactory Look beyond classroom requirements, showing interest in pursuing knowledge independently	3 pts Unsatisfactory Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently	2 pts Poor Relies on classroom instruction only	1 pts Very Poor No initiative or interest in acquiring new knowledge	6 pts
 SO 7 PI 2 ILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Completes an assigned task independently and practices continuous improvement	5 pts Good Completes an assigned task without supervision or guidance	4 pts Satisfactory Requires minimal guidance to complete an assigned task	3 pts Unsatisfactory Requires detailed or step-by-step instructions to complete a task	2 pts Poor Shows little interest to complete a task independently	1 pts Very Poor No interest to complete a task independently	6 pts
 SO 7 PI 3 ILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions	5 pts Good Evaluate information from a variety of sources; formulates a clear and precise perspective.	4 pts Satisfactory Analyze information from a variety of sources; formulates a clear and precise perspective.	3 pts Unsatisfactory Apply the gathered information to formulate the problem	2 pts Poor Gather and summarized the information from a variety of sources but failed to formulate the problem	1 pts Very Poor Gather information from a variety of sources	6 pts
 SO 7 PI 4 ILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.	5 pts Good Ideas are creative and adapt the new knowledge to solve a problem or address an issue	4 pts Satisfactory Ideas are creative in solving a problem, or address an issue	3 pts Unsatisfactory Shows some creative ways to solve the problem	2 pts Poor Shows initiative and attempt to develop creative ideas to solve the problem	1 pts Very Poor Ideas are copied or restated from the sources consulted	6 pts
Total Points: 24							