Activity No. 3.1	
Control Structures [Part 2]	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed:** 08/12/25
Section: CPE11S1	**Date Submitted:** 08/18/25
Name(s): James Daniel M. Verano	**Instructor:** Engr. Jimlord M. Quejado
6. Output	

1. *Develop a C++ program that will determine if a department store customer has exceeded the credit limit on a charge account. For each customer, the following facts are available:*
   1. *Account number*
   2. *Balance at the beginning of the month*
   3. *Total of all items charged by this customer this month*
   4. *Total of all credits applied to this customer's account this month*
   5. *Allowed credit limit*

The program should input each of these facts, calculate the new balance (=beginning balance +charges -credits), and determine if the new balance exceeds the customer's credit limit. For those customers whose credit limit is exceeded, the program should display the customer's account number, credit limit, new balance and the message "Credit limit exceeded."

Sample 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: 5500.00

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.

[Source Code:]

hands-on Activitg 3.1.cpp | Hands-on Activity 3.1_P1.cpp

```cpp
1    #include <iostream>
2    #include <iomanip>
3
4    int main() {
5
6        int Accnum;
7        float beginBal, totCharg, totCred, credLim, totBal;
8        const int SENTINEL = -1;
9
10
11       while (Accnum != SENTINEL) {
12           std::cout << "Enter Account Number (-1 to end): "; std::cin>> Accnum;
13           if (Accnum == SENTINEL) {
14               break;
15           }
16           std::cout << "Enter beginning balance: "; std::cin>> beginBal;
17           std::cout << "Enter total charges: "; std::cin>> totCharg;
18           std::cout << "Enter total credits: "; std::cin>> totCred;
19           std::cout << "Enter credit limit: "; std::cin>> credLim; std::cout << "\n";
20
21           totBal = beginBal + totCred;
22
23           if(totBal > credLim) {
24               std::cout << "Account: " << Accnum; std::cout << "\n";
25               std::cout << "Credit limit: " << std::fixed << std::setprecision(2) << credLim; std::cout << "\n";
26               std::cout << "New Balance: " << totBal; std::cout << "\n";
27               std::cout << "[-- Credit Limit Exceeded. --]"; std::cout << "\n";
28               std::cout << "\n";
29           } else {
30               std::cout << "New Balance: " << totBal; std::cout << "\n";
31           }
32       }
33
34       return 0;
35   }
```

[Output/Result:]

```
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: 5500.00
New 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

New Balance: 1321.00
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

New Balance: 600.00
Enter Account Number (-1 to end): -1

--------------------------------
Process exited after 62.85 seconds with return value 0
Press any key to continue . . .
```

2. Because of the price of gasoline, drivers are concerned with the mileage obtained by their automobiles. One driver has kept track of several tankfuels of gasoline by recording miles driven and gallons used for each tankful. Develop a program that will input the miles driven and gallons used for each tankful. The program should calculate and display the miles per gallon obtained for each tankful. After processing all input information, the program should calculate and print the combined miles per gallon obtained for all tank fuels.

Sample output:

Enter the gallons used (-1 to end): 12.8

Enter the miles driven: 287

The miles / gallon for this tank was 22.421875

Enter the gallons used (-1 to end): 10.3

Enter the miles driven: 200

The miles / gallon for this tank was 19.417475

Enter the gallons used (-1 to end): 5

Enter the miles driven: 120

The miles / gallon for this tank was 24.000000

Enter the gallons used (-1 to end):

The overall average miles/gallon was 21.601423

[Source Code:]

hands-on Activitg 3.1.cpp

```cpp
1    #include <iostream>
2    #include <iomanip>
3
4    int main() {
5
6        float gallUsed,milesPerGall, overallAve;
7        float Miles, counter, sum;
8        const int SENTINEL= -1;
9
10       while (gallUsed != SENTINEL) {
11          counter++;
12          std::cout << "Enter the gallons used (-1 to end): "; std::cin>> gallUsed;
13          if (gallUsed == SENTINEL) {
14             counter--;
15             std::cout << "The overall average miles/gallon was: " << overallAve / counter;
16             break;
17          }
18          std::cout << "Enter the miles driven: "; std::cin>> Miles;
19          milesPerGall = Miles / gallUsed;
20          overallAve = sum += milesPerGall;
21          std::cout << "The miles / gallon for this tank was: " << std::fixed << std::setprecision(6) << milesPerGall << "\n";
22          std::cout << "\n";
23       }
24
25       return 0;
26    }
```

[Output/Result:]

```
Enter the gallons used (-1 to end): 12.8
Enter the miles driven: 287
The miles / gallon for this tank was: 22.421875

Enter the gallons used (-1 to end): 10.3
Enter the miles driven: 200
The miles / gallon for this tank was: 19.417475

Enter the gallons used (-1 to end): 5
Enter the miles driven: 120
The miles / gallon for this tank was: 24.000000

Enter the gallons used (-1 to end): -1
The overall average miles/gallon was: 21.946449
-------------------------------
Process exited after 18.19 seconds with return value 0
Press any key to continue . . .
```

3. Create a program that will calculate the cost of sending a small parcel. The post office charges P5.00 for the first 300g, and P2.00 for every 100g thereafter (rounded up), up to a maximum weight of 1000g.

[Source Code:]

hands-on Activitg 3.1.cpp | Hands-on Activity 3.1_P1.cpp | Hands-on Activity 3.1_P3.cpp

```cpp
1   #include <iostream>
2   #include <iomanip>
3
4   int main() {
5
6       float mainCharges = 5.00;
7       float addCharges = 2.00;
8       float totCharges;
9       int grams;
10      const int SENTINEL = -1;
11
12      while (grams != SENTINEL) {
13
14          std::cout << "Enter parcels weight in grams(g)(-1 to stop): "; std::cin>> grams; std::cout << "\n";
15
16      if (grams <= 300) { //5
17          std::cout << "Your parcel costs: P" << std::fixed << std::setprecision(2) << mainCharges; std::cout << "\n";
18          std::cout << "\n";
19      } else if (grams <= 400) { // 7
20          totCharges = mainCharges + addCharges;
21          std::cout << "Your parcel costs: P" << std::fixed << std::setprecision(2) << totCharges; std::cout << "\n";
22          std::cout << "\n";
23      } else if (grams <= 500) { // 9
24          totCharges = mainCharges + ( 2 * addCharges);
25          std::cout << "Your parcel costs: P" << std::fixed << std::setprecision(2) << totCharges; std::cout << "\n";
26          std::cout << "\n";
```

```
27        } else if (grams <= 600) { // 11
28            totCharges = mainCharges + ( 3 * addCharges);
29            std::cout << "Your parcel costs: P" << std::fixed << std::setprecision(2) << totCharges; std::cout << "\n";
30            std::cout << "\n";
31        } else if (grams <= 700) { // 13
32            totCharges = mainCharges + ( 4 * addCharges);
33            std::cout << "Your parcel costs: P" << std::fixed << std::setprecision(2) << totCharges; std::cout << "\n";
34            std::cout << "\n";
35        } else if (grams <= 800) { // 15
36            totCharges = mainCharges + ( 5 * addCharges);
37            std::cout << "Your parcel costs: P" << std::fixed << std::setprecision(2) << totCharges; std::cout << "\n";
38            std::cout << "\n";
39        } else if (grams <= 900) { // 17
40            totCharges = mainCharges + ( 6 * addCharges);
41            std::cout << "Your parcel costs: P" << std::fixed << std::setprecision(2) << totCharges; std::cout << "\n";
42            std::cout << "\n";
43        } else if (grams <= 1000) { // 19
44            totCharges = mainCharges + ( 7 * addCharges);
45            std::cout << "Your parcel costs: P" << std::fixed << std::setprecision(2) << totCharges; std::cout << "\n";
46            std::cout << "\n";
47        } else if (grams > 1000) {
48            std::cout << " [ Error! ] The weight limit is 1000g. "; std::cout << "\n";
49            std::cout << "\n";
50        } else {
51            std::cout << "[-- Invalid Input. --]"; std::cout << "\n";
52            std::cout << "\n";
53        }
54
55    }
56
57        return 0;
58 }
```

[Output/Result:]

```
C:\Users\lenovo\Downloads\Hands-on Activity 3.1_P3.exe

Enter parcels weight in grams(g)(-1 to stop): 300

Your parcel costs: P5.00

Enter parcels weight in grams(g)(-1 to stop): 400

Your parcel costs: P7.00

Enter parcels weight in grams(g)(-1 to stop): 500

Your parcel costs: P9.00

Enter parcels weight in grams(g)(-1 to stop): 600

Your parcel costs: P11.00

Enter parcels weight in grams(g)(-1 to stop): 700

Your parcel costs: P13.00

Enter parcels weight in grams(g)(-1 to stop): 800

Your parcel costs: P15.00

Enter parcels weight in grams(g)(-1 to stop): 1000

Your parcel costs: P19.00

Enter parcels weight in grams(g)(-1 to stop):
```

4. Write a program that displays a menu for simple conversion such as the following:
- (1) cm - inches
- (2) inches - cm
- (3) feet - meter
- (4) meter - feet

Once selected, user will be asked to enter a float and be converted. After the conversion the user would be ask to convert another until the user don't want anymore. Display your samples for all conversions

[Source Code:]

Hands-on Activity 3.1_P4.cpp

```cpp
1    #include <iostream>
2    #include <iomanip>
3
4    int main() {
5
6        int A;
7        float conversion, converted;
8        float inches = 0.393701;
9        float centimeter = 2.54;
10       float meter = 3.28084;
11       float feet = 0.3048;
12       const int SENTINEL = -1;
13
14       while (conversion != SENTINEL) {
15           std::cout << "Enter the desired number to convert: "; std::cin>> conversion; std::cout << "\n";
16           if (conversion == SENTINEL) {
17               break;
18           }
19           std::cout << "Press (1) to convert centimeter ==> inches \n" ;
20           std::cout << "Press (2) to convert inches ==> centimeter \n";
21           std::cout << "Press (3) to convert feet ==> meter \n";
22           std::cout << "Press (4) to convert meter ==> feet \n ==> ";
23           std::cin >> A; std::cout << "\n";
24
25           if (A == 1) {
26               converted = conversion * inches;
27               std::cout << "[ " << conversion << " ]" << " centimeter is equal to ==> " << std::fixed << std::setprecision(2) << converted << " inches";
28               std::cout << "\n \n";
29           } else if (A == 2) {
30               converted = conversion * centimeter;
31               std::cout << "[ " << conversion << " ]" << " inches is equal to ==> " << std::fixed << std::setprecision(2) << converted << " centimeter";
32               std::cout << "\n \n";
33           } else if (A == 3) {
34               converted = conversion * feet;
35               std::cout << "[ " << conversion << " ]" << " feet is equal to ==> " << std::fixed << std::setprecision(2) << converted << " meter";
36               std::cout << "\n \n";
37           } else if (A == 4) {
38               converted = conversion * meter;
39               std::cout << "[ " << conversion << " ]" << " meter is equal to ==> " << std::fixed << std::setprecision(2) << converted << " feet";
40               std::cout << "\n \n";
41           }
42       }
43       return 0;
44   }
```

[Output/Result:]

```
Enter the desired number to convert (-1 to stop): 5

Press (1) to convert centimeter ==> inches
Press (2) to convert inches ==> centimeter
Press (3) to convert feet ==> meter
Press (4) to convert meter ==> feet
 ==> 1

[ 5 ] centimeter is equal to ==> 1.97 inches

Enter the desired number to convert (-1 to stop): 4.60

Press (1) to convert centimeter ==> inches
Press (2) to convert inches ==> centimeter
Press (3) to convert feet ==> meter
Press (4) to convert meter ==> feet
 ==> 2

[ 4.60 ] inches is equal to ==> 11.68 centimeter

Enter the desired number to convert (-1 to stop): 7.40

Press (1) to convert centimeter ==> inches
Press (2) to convert inches ==> centimeter
Press (3) to convert feet ==> meter
Press (4) to convert meter ==> feet
 ==> 3

[ 7.40 ] feet is equal to ==> 2.26 meter
```

```
Enter the desired number to convert (-1 to stop): 6.20

Press (1) to convert centimeter ==> inches
Press (2) to convert inches ==> centimeter
Press (3) to convert feet ==> meter
Press (4) to convert meter ==> feet
 ==> 4

[ 6.20 ] meter is equal to ==> 20.34 feet

Enter the desired number to convert (-1 to stop): -1


-------------------------------
Process exited after 47.26 seconds with return value 0
Press any key to continue . . .
```

5. Write a program that displays a menu for simple computation of formula such as the following:
- (1) Area of circle, will ask for radius from user
- (2) Area of rectangle, will ask for L and W
- (3) Area of triangle, will ask for B and H
- (4) Area of square - feet, will ask for S

Once selected, user will be asked to enter a float and be computed as selected. After the computation, the user would be ask to compute another until the user don't want anymore. Display your samples for all conversions
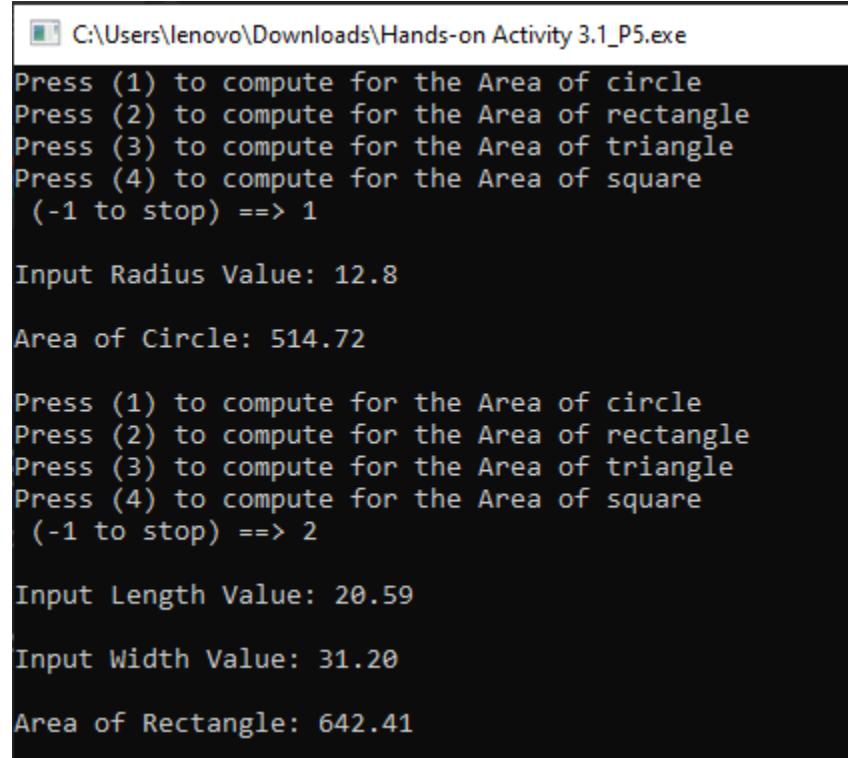
[Source Code:]

Hands-on Activity 3.1_P4.cpp Hands-on Activity 3.1_P5.cpp

```cpp
1    #include <iostream>
2    #include <iomanip>
3    #include <cmath>
4
5    int main() {
6
7        int A;
8        float areaOfC, areaOfRec, areaOfTri, areaOfSqua;
9        float radius, length, width, base, height, side;
10       float pi_value = M_PI;
11       const int SENTINEL = -1;
12
13       while (A != SENTINEL) {
14           std::cout << "Press (1) to compute for the Area of circle \n";
15           std::cout << "Press (2) to compute for the Area of rectangle \n";
16           std::cout << "Press (3) to compute for the Area of triangle \n";
17           std::cout << "Press (4) to compute for the Area of square \n (-1 to stop) ==> ";
18           std::cin >> A; std::cout << "\n";
19           if (A == SENTINEL) {
20               break;
21           }
22
23           if (A == 1) {
24               std::cout << "Input Radius Value: "; std::cin>> radius; std::cout << "\n";
25               areaOfC = pi_value * ( radius * radius);
26               std::cout << "Area of Circle: " << std::fixed << std::setprecision(2) << areaOfC;
27               std::cout << "\n \n";
```

```cpp
28  } else if (A == 2) {
29      std::cout << "Input Length Value: "; std::cin>> length; std::cout << "\n";
30      std::cout << "Input Width Value: "; std::cin>>  width; std::cout << "\n";
31      areaOfRec = length * width;
32      std::cout << "Area of Rectangle: " << std::fixed << std::setprecision(2) << areaOfRec;
33      std::cout << "\n \n";
34  } else if (A == 3) {
35      std::cout << "Input Base Value: "; std::cin>> base; std::cout << "\n";
36      std::cout << "Input height Value: "; std::cin>> height; std::cout << "\n";
37      areaOfTri = 0.5 * (base * height);
38      std::cout << "Area of Triangle: " << std::fixed << std::setprecision(2) << areaOfTri;
39      std::cout << "\n \n";
40  } else if (A == 4) {
41      std::cout << "Input Side Value: "; std::cin>> side; std::cout << "\n";
42      areaOfSqua = side * side;
43      std::cout << "Area of Square: " << std::fixed << std::setprecision(2) << areaOfSqua;
44      std::cout << "\n \n";
45  }
46  }
47  return 0;
48  }
```

[Output/Result:]

```
C:\Users\lenovo\Downloads\Hands-on Activity 3.1_P5.exe

Press (1) to compute for the Area of circle
Press (2) to compute for the Area of rectangle
Press (3) to compute for the Area of triangle
Press (4) to compute for the Area of square
 (-1 to stop) ==> 1

Input Radius Value: 12.8

Area of Circle: 514.72

Press (1) to compute for the Area of circle
Press (2) to compute for the Area of rectangle
Press (3) to compute for the Area of triangle
Press (4) to compute for the Area of square
 (-1 to stop) ==> 2

Input Length Value: 20.59

Input Width Value: 31.20

Area of Rectangle: 642.41
```

```
Press (1) to compute for the Area of circle
Press (2) to compute for the Area of rectangle
Press (3) to compute for the Area of triangle
Press (4) to compute for the Area of square
 (-1 to stop) ==> 3

Input Base Value: 45.60

Input height Value: 51.30

Area of Triangle: 1169.64

Press (1) to compute for the Area of circle
Press (2) to compute for the Area of rectangle
Press (3) to compute for the Area of triangle
Press (4) to compute for the Area of square
 (-1 to stop) ==> 4

Input Side Value: 52

Area of Square: 2704.00
```

```
Press (1) to compute for the Area of circle
Press (2) to compute for the Area of rectangle
Press (3) to compute for the Area of triangle
Press (4) to compute for the Area of square
 (-1 to stop) ==> -1


--------------------------------
Process exited after 185.3 seconds with return value 0
Press any key to continue . . . _
```

7. Supplementary Activity

additional activity #1

pseudo code

image of flow chart

additional activity #1-6
pseudo code

8. Conclusion

This activity has taught me to combine and to use different types of control structures such as, sequential, selection, and iteration. With the use of else if statements, I was able to successfully deliver the intended outcome of what the hands-on activity wished to represent in the pseudo code. This also enhanced my basic and advance knowledge on using different strategies on developing and creating loops and control structure to keep the sequence of the code going.

9. Assessment Rubric