

Activity No. 3.1	
<b>Hands-on Activity 3.1: Control Structures (part 2)</b>	
<b>Course Code:</b> CPE 007	<b>Program:</b> Computer Engineer
<b>Course Title:</b> Programming Logic and Design	<b>Date Performed:</b> 11/08/2025
<b>Section:</b> CPE11S1	<b>Date Submitted:</b> 18/08/2025
<b>Name:</b> Canoy Hail B.	<b>Instructor:</b> Jimlord M. Quejado
<b>6. Output</b>	
<b>7. Supplementary Activity</b>	
<p><b>1. Code :</b></p> <pre>#include &lt;iostream&gt; using namespace std;  int main() {     int accountNumber;     double beginningBalance, charges, credits, creditLimit, newBalance;      while (true) {         cout &lt;&lt; "\nEnter account number (-1 to end): ";         cin &gt;&gt; accountNumber;          if (accountNumber == -1) {             cout &lt;&lt; "\nProgram ends." &lt;&lt; endl;             break;         }          cout &lt;&lt; "\nEnter beginning balance: ";         cin &gt;&gt; beginningBalance;          cout &lt;&lt; "\nEnter total charges: ";         cin &gt;&gt; charges;          cout &lt;&lt; "\nEnter total credits: ";         cin &gt;&gt; credits;          cout &lt;&lt; "\nEnter credit limit: ";         cin &gt;&gt; creditLimit;</pre>	

```

newBalance = beginningBalance + charges - credits;

if (newBalance > creditLimit) {
    cout << "\nAccount: " << accountNumber << endl;
    cout << "Credit limit: " << creditLimit << endl;
    cout << "Balance: " << newBalance << endl;
    cout << "Credit Limit Exceeded." << endl;
}

return 0;
}

```

```

main.cpp | Run | Output
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int accountNumber;
6     double beginningBalance, charges, credits, creditlimit, newBalance;
7
8     while (true) {
9         cout << "\nEnter account number (-1 to end): ";
10        cin >> accountNumber;
11
12        if (accountNumber == -1) {
13            cout << "\nProgram ends." << endl;
14            break;
15        }
16
17        cout << "\nEnter beginning balance: ";
18        cin >> beginningBalance;
19
20        cout << "\nEnter total charges: ";
21        cin >> charges;
22
23        cout << "\nEnter total credits: ";
24        cin >> credits;
25
26        cout << "\nEnter credit limit: ";
27        cin >> creditlimit;
28
29        newBalance = beginningBalance - charges - credits;
30
31        if (newBalance > creditlimit) {
32            cout << "\nAccount: " << accountNumber << endl;
33            cout << "Credit limit: " << creditlimit << endl;
34            cout << "Balance: " << newBalance << endl;
35            cout << "Credit Limit Exceeded." << endl;
36        }
37    }
38
39    return 0;
40 }
41

```

Enter account number (-1 to end): 24  
Enter beginning balance: 234  
Enter total charges: 34  
Enter total credits: 3  
Enter credit limit: 2  
Account: 24  
Credit limit: 2  
Balance: 265  
Credit limit Exceeded.

## 2. Code :

```

#include <iostream>
using namespace std;

int main() {
    double gallons, miles, mpg;
    double totalMiles = 0, totalGallons = 0;

    do {
        cout << "\nEnter the gallons used (-1 to end): ";
        cin >> gallons;

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

        if (gallons <= 0) {
            cout << "Invalid input. Gallons must be positive." << endl;
        }
    }
}

```

```

        continue;
    }

cout << "Enter the miles driven: ";
cin >> miles;

if (miles <= 0) {
    cout << "Invalid input. Miles must be positive." << endl;
    continue;
}

mpg = miles / gallons;
cout << "The miles / gallon for this tank was " << mpg << endl;

totalMiles += miles;
totalGallons += gallons;

} while (true);

if (totalGallons > 0) {
    double combinedMPG = totalMiles / totalGallons;
    cout << "\nThe overall average miles / gallon was " << combinedMPG << endl;
} else {
    cout << "\nNo valid data entered." << endl;
}

return 0;

```

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     double gallons, miles, mpg;
7     double totalMiles = 0, totalGallons = 0;
8
9     do {
10         cout << "Enter the gallons used (-1 to end): ";
11         cin >> gallons;
12         if (gallons == -1) {
13             break;
14         }
15         if (gallons <= 0) {
16             cout << "Invalid input. Gallons must be positive." << endl;
17             continue;
18         }
19         cout << "Enter the miles driven: ";
20         cin >> miles;
21         if (miles <= 0) {
22             cout << "Invalid input. Miles must be positive." << endl;
23             continue;
24         }
25         mpg = miles / gallons;
26         cout << "The miles / gallon for this tank was " << mpg << endl;
27         totalMiles += miles;
28         totalGallons += gallons;
29     } while (gallons != -1);
30
31     double combinedMPG = totalMiles / totalGallons;
32     cout << "\nThe overall average miles / gallon was " << combinedMPG << endl;
33     if (combinedMPG == 0) {
34         cout << "\nNo valid data entered." << endl;
35     }
36 }
37
38 return 0;

```

Output:

```

Enter the gallons used (-1 to end): 12
Enter the miles driven: 24
The miles / gallon for this tank was 2.00000
Enter the gallons used (-1 to end): 23
Enter the miles driven: 23
The miles / gallon for this tank was 1.00000
Enter the gallons used (-1 to end): -1
The overall average miles / gallon was 0.74286

*** Code Execution Successful ***

```

### 3. Code :

```

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

int main() {
    int weight;
    double cost;

```

```

do {
    cout << "Enter parcel weight in grams (or -1 to exit): ";
    cin >> weight;

    if (weight == -1) {
        cout << "\nProgram ended." << endl;
        break;
    }

    if (weight > 1000) {
        cout << "Parcel too heavy. Maximum allowed is 1000g.\n" << endl;
    }
    else if (weight <= 300) {
        cost = 5.00;
        cout << "Cost to send parcel: $" << fixed << setprecision(2) << cost << "\n" << endl;
    }
    else {
        int extraWeight = weight - 300;
        int extraUnits = (extraWeight + 99) / 100;
        cost = 5.00 + (extraUnits * 2.00);
        cout << "Cost to send parcel: $" << fixed << setprecision(2) << cost << "\n" << endl;
    }
}

} while (true);

return 0;
}

```

```

main.cpp | Run | Output
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main() {
6     int weight;
7     double cost;
8
9     do {
10         cout << "Enter parcel weight in grams (or -1 to exit): ";
11         cin >> weight;
12
13         if (weight == -1) {
14             cout << "\nProgram ended." << endl;
15             break;
16         }
17
18         if (weight > 1000) {
19             cout << "Parcel too heavy. Maximum allowed is 1000g.\n" << endl;
20         }
21         else if (weight <= 300) {
22             cost = 5.00;
23             cout << "Cost to send parcel: $" << fixed << setprecision(2) << cost << "\n" << endl;
24         }
25         else {
26             int extraWeight = weight - 300;
27             int extraUnits = (extraWeight + 99) / 100;
28             cost = 5.00 + (extraUnits * 2.00);
29             cout << "Cost to send parcel: $" << fixed << setprecision(2) << cost << "\n" << endl;
30         }
31
32     } while (true);
33
34     return 0;
35 }
36

```

#### 4. Code :

```

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

int main() {
    int choice;
    float value, result;

```

```
char again;

do {
    cout << "\nUnit Conversion Menu:\n";
    cout << "(1) cm to inches\n";
    cout << "(2) inches to cm\n";
    cout << "(3) feet to meter\n";
    cout << "(4) meter to feet\n";
    cout << "Enter your choice (1-4): ";
    cin >> choice;

    cout << "Enter value to convert: ";
    cin >> value;

    if (choice == 1) {
        result = value / 2.54;
        cout << fixed << setprecision(2);
        cout << value << " cm = " << result << " inches" << endl;
    } else if (choice == 2) {
        result = value * 2.54;
        cout << fixed << setprecision(2);
        cout << value << " inches = " << result << " cm" << endl;
    } else if (choice == 3) {
        result = value * 0.3048;
        cout << fixed << setprecision(2);
        cout << value << " feet = " << result << " meters" << endl;
    } else if (choice == 4) {
        result = value / 0.3048;
        cout << fixed << setprecision(2);
        cout << value << " meters = " << result << " feet" << endl;
    } else {
        cout << "Invalid choice." << endl;
    }

    cout << "\nDo you want to convert another? (y/n): ";
    cin >> again;

} while (again == 'Y' || again == 'N');

cout << "\nProgram ended." << endl;
return 0;
}
```

```

main.cpp | Run | Output
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main() {
6     int choice;
7     float value, result;
8     char again;
9
10    do {
11        cout << "\nUnit Conversion Menu:\n";
12        cout << "(1) cm to inches";
13        cout << "(2) inches to cm";
14        cout << "(3) feet to meter";
15        cout << "(4) meter to feet\n";
16        cout << "Enter your choice (1-4): ";
17        cin >> choice;
18
19        cout << "Enter value to convert: ";
20        cin >> value;
21
22        if (choice == 1) {
23            result = value / 2.54;
24            cout << fixed << setprecision(2);
25            cout << value << " cm " << result << " inches" << endl;
26        } else if (choice == 2) {
27            result = value * 2.54;
28            cout << fixed << setprecision(2);
29            cout << value << " inches " << result << " cm" << endl;
30        } else if (choice == 3) {
31            result = value * 0.3048;
32            cout << fixed << setprecision(2);
33            cout << value << " feet " << result << " meters" << endl;
34        } else if (choice == 4) {
35            result = value / 0.3048;
36            cout << fixed << setprecision(2);
37            cout << value << " meters " << result << " feet" << endl;
38        } else {
39            cout << "Invalid choice." << endl;
40        }
41
42        cout << "\nDo you want to convert another? (y/n): ";
43        cin >> again;
44
45    } while (again == 'Y' || again == 'N');
46
47    cout << "\nProgram ended." << endl;
48    return 0;
49 }
50

```

## 5. Code :

```

#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

int main() {
    int choice;
    float radius, length, width, base, height, side, area;
    char again;

    do {
        cout << "\nArea Computation Menu:\n";
        cout << "(1) Area of Circle\n";
        cout << "(2) Area of Rectangle\n";
        cout << "(3) Area of Triangle\n";
        cout << "(4) Area of Square (in feet)\n";
        cout << "Enter your choice (1-4): ";
        cin >> choice;

        cout << fixed << setprecision(2);

        if (choice == 1) {
            cout << "Enter radius: ";
            cin >> radius;
            area = 3.1416 * radius * radius;
            cout << "Area of Circle = " << area << endl;
        } else if (choice == 2) {
            cout << "Enter length: ";
            cin >> length;
            cout << "Enter width: ";
            cin >> width;
        }
    }
}

```

```

area = length * width;
cout << "Area of Rectangle = " << area << endl;
} else if (choice == 3) {
    cout << "Enter base: ";
    cin >> base;
    cout << "Enter height: ";
    cin >> height;
    area = 0.5 * base * height;
    cout << "Area of Triangle = " << area << endl;
} else if (choice == 4) {
    cout << "Enter side (in feet): ";
    cin >> side;
    area = side * side;
    cout << "Area of Square = " << area << " square feet" << endl;
} else {
    cout << "Invalid choice." << endl;
}

cout << "\nDo you want to compute another? (y/n): ";
cin >> again;

} while (again == 'y' || again == 'Y');

cout << "\nProgram ended." << endl;
return 0;
}

```

```

main.cpp | (1) Area of Circle
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5
6 int main() {
7     int choice;
8     float radius, length, width, base, height, side, area;
9     char again;
10
11    do {
12        cout << "\nEnter Computation Menu:\n";
13        cout << "(1) Area of Circle\n";
14        cout << "(2) Area of Rectangle\n";
15        cout << "(3) Area of Triangle\n";
16        cout << "(4) Area of Square (in feet)\n";
17        cout << "Enter your choice (1-4): ";
18        cin >> choice;
19
20        cout << fixed << setprecision(2);
21
22        if (choice == 1) {
23            cout << "Enter radius: ";
24            cin >> radius;
25            area = M_PI * radius * radius;
26            cout << "Area of Circle = " << area << endl;
27        } else if (choice == 2) {
28            cout << "Enter length: ";
29            cin >> length;
30            cout << "Enter width: ";
31            cin >> width;
32            area = length * width;
33            cout << "Area of Rectangle = " << area << endl;
34        } else if (choice == 3) {
35            cout << "Enter base: ";
36            cin >> base;
37            cout << "Enter height: ";
38            cin >> height;
39            area = 0.5 * base * height;
40            cout << "Area of Triangle = " << area << endl;
41        } else if (choice == 4) {
42            cout << "Enter side (in feet): ";
43            cin >> side;
44            area = side * side;
45            cout << "Area of Square = " << area << " square feet" << endl;
46        } else {
47            cout << "Invalid choice." << endl;
48        }
49
50        cout << "\nDo you want to compute another? (y/n): ";
51        cin >> again;
52
53    } while (again == 'y' || again == 'Y');
54
55    cout << "\nProgram ended." << endl;
56
57 }

```

## 8. Conclusion

- I tried applying while, Do-while, loop and if. I'm understanding it more and gaining more knowledge of conditions to solve real life problems using coding.

## 9. Assessment Rubric

Rubric for SO 7 (B)						
Criteria	Ratings					Pts
④ SO 7 PI 1 III.L04 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	4 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
⑤ SO 7 PI 2 III.L04 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	4 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
⑥ SO 7 PI 3 III.L04 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
⑦ SO 7 PI 4 III.L04 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