

Contents

- [while Loop - Coding Exercises](#)
- [do-while Loop - Coding Exercises](#)

Section 1: **while** Loop

Problem Statement 1: Repeat a Message

Task: Write a C++ program using a **while** loop to display your name five times, with each repetition on a new line.

Expected Output:

```
Hamza
Hamza
Hamza
Hamza
Hamza
```

*Note: Replace **Hamza** with your own name.*

Problem Statement 2: Print a Sequence

Task: Write a C++ program using a **while** loop to display all integers from 1 to 50 in ascending order.

Expected Output:

```
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
```

Problem Statement 3: Print Odd Numbers

Task: Write a C++ program using a **while** loop to display all odd numbers from 1 to 25 in ascending order.

Expected Output:

```
1 3 5 7 9 11 13 15 17 19 21 23 25
```

Problem Statement 4: Countdown

Task: Write a C++ program using a **while** loop to display numbers from 50 down to 1 in descending order.

Expected Output:

```
50 49 48 47 46 45 44 43 42 41
40 39 38 37 36 35 34 33 32 31
30 29 28 27 26 25 24 23 22 21
20 19 18 17 16 15 14 13 12 11
10 9 8 7 6 5 4 3 2 1
```

Problem Statement 5: Calculate a Sum

Task: Write a C++ program using a **while** loop to calculate and display the sum of all integers from 1 to 20.

Expected Output:

```
Sum = 210
```

Hint: Use a variable to store the running sum and update it during each loop iteration.

Problem Statement 6: Multiplication Table

Task: Write a C++ program that asks the user to enter an integer and displays its multiplication table from 1 to 10 using a **while** loop.

Sample Input:

```
Enter a number: 7
```

Expected Output:

```
Multiplication Table of 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

Section 2: **do-while** Loop

Problem Statement 7: Repeat a Message

Task: Write a C++ program using a **do-while** loop to display your name five times.

Expected Output:

```
Hamza
Hamza
Hamza
Hamza
Hamza
```

*Note: Replace **Yasir** with your own name.*

Problem Statement 8: Print Numbers from 1 to 20

Task: Write a C++ program using a **do-while** loop to display numbers from 1 to 20 in ascending order.

Expected Output:

```
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
```

Problem Statement 9: Print Numbers in Reverse Order

Task: Write a C++ program using a **do-while** loop to display numbers from 20 down to 1.

Expected Output:

```
20 19 18 17 16 15 14 13 12 11
10 9 8 7 6 5 4 3 2 1
```

Problem Statement 10: Display Even Numbers

Task: Write a C++ program using a **do-while** loop to display all even numbers from 2 to 50.

Expected Output:

```
2 4 6 8 10 12 14 16 18 20
22 24 26 28 30 32 34 36 38 40
42 44 46 48 50
```

Problem Statement 11: Multiplication Table

Task: Write a C++ program that asks the user to enter a number and displays its multiplication table from 1 to 10 using a **do-while** loop.

Sample Input:

```
Enter a number: 5
```

Expected Output:

```
Multiplication Table of 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

Problem Statement 12: Calculate the Sum from 1 to n

Task: Write a C++ program that asks the user to enter a positive integer **n** and calculates the sum of all numbers from 1 to **n** using a **do-while** loop.

Sample Input:

```
Enter a positive integer: 5
```

Expected Output:

```
Sum = 15
```

Problem Statement 13: Display Odd Numbers

Task: Write a C++ program that asks the user to enter a positive integer and displays all odd numbers from 1 up to that number using a **do-while** loop.

Sample Input:

```
Enter a positive integer: 15
```

Expected Output:

```
Odd numbers: 1 3 5 7 9 11 13 15
```

Problem Statement 14: Sum of Even Numbers

Task: Write a C++ program that asks the user to enter a positive integer **n** and calculates the sum of all even numbers from **1** to **n** using a **do-while** loop. Use an **if** statement to check whether each number is even.

Hint: Use `if (number % 2 == 0)` to identify even numbers, add them to `sum`, and increment `number` in each iteration.

Sample Input:

100

Expected Output:

```
Enter a positive integer: 100
Sum of even numbers from 1 to 100 = 2550
```

Problem Statement 15: Sum of Odd Numbers

Task: Write a C++ program that asks the user to enter a positive integer `n` and calculates the sum of all odd numbers from `1` to `n` using a `do-while` loop. Use an `if` statement to check whether each number is odd.

Hint: Use `if (number % 2 != 0)` to identify odd numbers, add them to `sum`, and increment `number` in each iteration.

Sample Input:

100

Expected Output:

```
Enter a positive integer: 100
Sum of odd numbers from 1 to 100 = 2500
```

Problem Statement 16: Identify Even and Odd Numbers

Task: Write a C++ program that asks the user to enter a positive integer `n` and displays every number from `1` to `n` with its status, `Even` or `Odd`, using a `do-while` loop and an `if-else` statement.

Hint: Use `if (number % 2 == 0)` to identify even numbers and `else` to identify odd numbers.

Sample Input:

100

Expected Output (first 10 and last 5 lines shown):

```
Enter a positive integer: 100
```

```
1 is Odd
2 is Even
3 is Odd
4 is Even
5 is Odd
6 is Even
7 is Odd
8 is Even
9 is Odd
10 is Even
...
96 is Even
97 is Odd
98 is Even
99 is Odd
100 is Even
```

Note: Your program must display all numbers from 1 to 100, not just the numbers shown above.

Problem Statement 17: Repeating Menu

Task: Write a C++ program that displays the following menu repeatedly using a **do-while** loop until the user selects Exit (option 3).

Menu:

```
===== Main Menu =====
1. Say Hello
2. Display Welcome Message
3. Exit
Enter your choice:
```

Requirements:

- If the user selects **1**, display **Hello!**.
- If the user selects **2**, display **Welcome to C++ Programming!**.
- If the user selects **3**, display **Program terminated.** and stop the loop.
- For any other choice, display **Invalid choice. Please try again..**

Sample Input:

```
1
2
3
```

Expected Output:

```
===== Main Menu =====
1. Say Hello
2. Display Welcome Message
3. Exit
Enter your choice: 1
Hello!

===== Main Menu =====
1. Say Hello
2. Display Welcome Message
3. Exit
Enter your choice: 2
Welcome to C++ Programming!

===== Main Menu =====
1. Say Hello
2. Display Welcome Message
3. Exit
Enter your choice: 3
Program terminated.
```

Problem Statement 18: Validate Positive Input

Task: Write a C++ program that asks the user to enter a positive number. If the number is zero or negative, keep asking until the user enters a positive number.

Sample Input:

```
Enter a positive number: -5
Enter a positive number: 0
Enter a positive number: -2
Enter a positive number: 8
```

Expected Output:

```
Valid number entered: 8
```

Problem Statement 19: Simple Calculator Menu

Task: Write a C++ program that repeatedly displays a calculator menu using a **do-while** loop until the user selects Exit.

Menu Options:

1. Addition
2. Subtraction
3. Exit

Requirements:

- If the user selects **1**, ask for two numbers and display their sum.
- If the user selects **2**, ask for two numbers and display their difference.
- Display the menu again after each calculation.
- If the user selects **3**, display **Calculator closed.** and terminate the loop.
- For an invalid choice, display an appropriate error message.

Sample Input:

```
1
10 5
2
10 5
3
```

Expected Output:

```
===== Simple Calculator =====
1. Addition
2. Subtraction
3. Exit
Enter your choice: 1
Enter two numbers: 10 5
Sum = 15

===== Simple Calculator =====
1. Addition
2. Subtraction
3. Exit
Enter your choice: 2
Enter two numbers: 10 5
Difference = 5

===== Simple Calculator =====
1. Addition
2. Subtraction
3. Exit
Enter your choice: 3
Calculator closed.
```