

In C programming, the **do-while** loop is a control flow statement that is similar to the **while** loop but with a key difference: the **do-while** loop guarantees that the block of code inside the loop is executed at least once, regardless of whether the loop condition is true or false.

The syntax of a **do-while** loop is as follows:

```
do
{
    // Code to be executed at least once
}
while (condition);
```

In C programming, the **do-while** loop is a control flow statement that is similar to the **while** loop but with a key difference: the **do-while** loop guarantees that the block of code inside the loop is executed at least once, regardless of whether the loop condition is true or false. The syntax of a **do-while** loop is as follows:

```
do { // Code to be executed at least once }
while (condition);
```

Here's a breakdown of the components:

- **do**: The keyword that signifies the beginning of the **do-while** loop.
- **{ }**: The curly braces define the scope of the loop. The code inside the curly braces is the block of code that gets executed.
- **while (condition)**: The loop condition is a Boolean expression that is evaluated after each iteration of the loop. If the condition is true, the loop will continue to execute. If the condition is false, the loop is exited, and the program continues with the next statement after the loop.

The general flow of a **do-while** loop is as follows:

- The code inside the loop is executed.
- The condition is evaluated.
- If the condition is true, the loop repeats steps 1 and 2.
- If the condition is false, the loop is exited, and the program continues with the next statement after the loop.

### **Example 1: - Behavior of Do while**

```
#include <stdio.h>
int main()
{
    int i = 10;
    do
    {
        printf("%d ", i);
        i++;
    }
    while (i <= 5);

    printf("\n");

    return 0;
}
```

### **Example 2 :- Game Loops**

```
#include <stdio.h>
int main()
{
    char playAgain;

    Do
```

```
{  
    printf("Do you want to play again? (y/n): ");  
    scanf(" %c", &playAgain);  
  
}  
while (playAgain == 'y' || playAgain == 'Y');  
  
printf("Thanks for playing!\n");  
  
return 0;  
}
```

### **Example 3 : - Input Validation**

```
#include <stdio.h>  
  
int main()  
{  
    int number;  
  
    do  
    {  
        printf("Enter a positive number: ");  
        scanf("%d", &number);  
  
        // Check if the input is valid  
        if (number <= 0)  
        {  
            printf("Invalid input. Please enter a positive number.\n");  
        }  
    }  
}
```

```
        }  
    }  
    while (number <= 0);  
  
    // Perform operations with the valid input  
    printf("You entered a valid positive number: %d\n", number);  
  
    return 0;  
}
```