

# C++ While Loop

## C++ While Loop

---

While Loop can execute a block of statements in a loop based on a condition.

In this tutorial, we learn the syntax of while loop in C++, its algorithm, flowchart, then some examples illustrating the usage of it. Later we shall go through Infinite While Loop and Nested While Loop.

## Syntax of While Loop

---

Following is the syntax of while loop in C++.

```
while (condition) {  
    // statement(s)  
}
```

At the start of while loop execution, the **condition** is checked. If the **condition** is true, **statement(s)** inside while block are executed. The **condition** is checked again. If it evaluates to true, the **statement(s)** inside the while loop are executed. This cycle goes on. If at all, the condition evaluates to false, execution comes out of while loop, meaning while loop is completed. And the program continues with the execution of statements after while loop if any.

## Algorithm

---

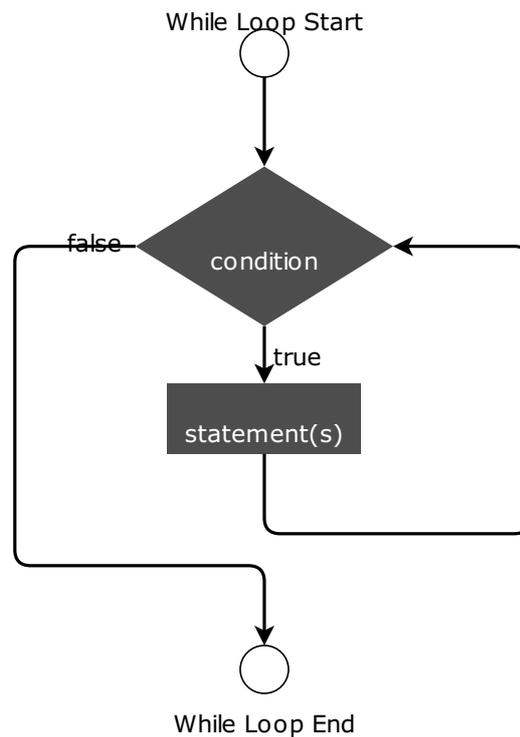
Following would be the algorithm of while loop.

1. Start.
2. Check the condition. If the condition is false, go to **step 4**.
3. Execute statement(s). Go to **step 2**.
4. Stop.

You have to take care of the initialization and update of the variables present in condition. And make sure that the condition would break after a definite number of iterations. If the condition is never going to be false, then the while loop is going to execute indefinitely.

## Flow Diagram

Following is the flow chart of flow diagram of while loop in C++.



### Example 1: While Loop

In this example, we shall write a while loop that prints numbers from 1 to 5. The while loop contains statement to print a number, and the condition checks if the number is within the limits.

#### C++ Program

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

int main() {
    int n = 5;
    int i = 1;
    while (i<=n) {
        cout << i << "\n";
        i++;
    }
}
```

#### Output

```
1
2
3
4
5
```

## Example 2: While Loop to Compute Factorial

---

In this example, we shall use while loop to compute factorial of a number.

### C++ Program

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

int main() {
    int n = 5;
    int factorial = 1;

    int i = 1;
    while (i<=n) {
        factorial *= i;
        i++;
    }

    cout << factorial;
}
```

### Output

```
120
```

## Example 3: While Loop to Compute Sum of first N Natural Numbers

---

In this example, we shall use while loop to compute the sum of first N natural numbers. We shall write a while loop with condition that it is true until it reaches given number, and during each iteration, we shall add this number to the `sum` .

### C++ Program

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

int main() {
    int n = 5;
    int sum = 0;

    int i = 1;
    while (i<=n) {
        sum += i;
        i++;
    }

    cout << sum;
}
```

## Output



## While Loop with break Statement

---

You can break the while loop abruptly using **break** statement. **break** statement ends the execution of the wrapping loop.

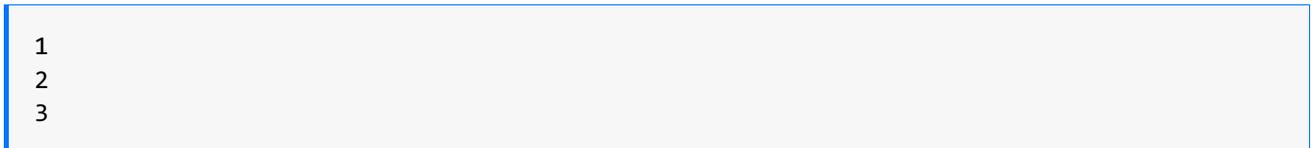
In the following example, we shall write a while loop that prints numbers from 1 to 10 . But, then we include a break statement such that when `i` is 4 , we break the while loop.

### C++ Program

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

int main() {
    int i = 1;
    while (i <= 10) {
        if (i == 4) {
            break;
        }
        cout << i << "\n";
        i++;
    }
}
```

## Output



## While Loop with continue Statement

---

You can skip the execution of statements in a while loop during an iteration using **continue** statement. **continue** statement takes the control to the condition without executing further statements in the loop.

In the following example, we shall write a while loop that prints numbers from 1 to 6 . But, then we include a **continue** statement such that when `i` is 4 , we skip the execution of further statements inside while loop.

### C++ Program

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

```
int main() {
    int i = 1;
    while (i <= 7) {
        if (i == 4) {
            i++;
            continue;
        }
        cout << i << "\n";
        i++;
    }
}
```

### Output

```
1
2
3
5
6
7
```

## Infinite While Loop

If the condition in while loop is going to be always true, then this loop runs indefinitely. This kind of loop is called infinite while loop.

Just a simple condition like `1==1` or `true`, will make the while loop to run indefinitely.

### C++ Program

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

int main() {

    while (true) {
        cout << "hello";
    }

}
```

### Output

The string "hello" is printed to the terminal indefinitely, until you interrupt and stop the program execution.

## While Loop with Update of Variable in Condition

You can update the loop control variable in the condition itself.

In the following example, we shall print the numbers from 1 to 5 .

### C++ Program

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

int main() {

    int i = 0;
    while (++i<=5) {
        cout << i << "\n";
    }

}
```

### Output

```
1
2
3
4
5
```

## Nested While Loop

While is just like another statement in C++. So, you can include a while loop inside the body a while loop, just like a statement.

In the following example program, we shall print a pattern that resembles a triangle, using nested while loop.

### C++ Program

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

int main() {
    int i = 1;
    while (i <= 5) {
        int j = 1;
        while (j <= i) {
            cout << " *";
            j++;
        }
        cout << "\n";
        i++;
    }
}
```

### Output

```
*
* *
* * *
* * * *
* * * * *
```

Outer while loop is used to traverse the rows, and inner while loop is used to traverse the columns.

## Conclusion

In this [C++ Tutorial](#), we learned the syntax of while loop in C++, its algorithm, flowchart, and usage with the help of example C++ programs.

### C++ Tutorials

- ◆ [C++ Tutorial](#)
- ◆ [C++ Hello World Program](#)
- ◆ [C++ If Else](#)
- ◆ [C++ Switch](#)
- ◆ [C++ Ternary Operator](#)
- ◆ [C++ Logical Operations](#)
- ◆ [C++ Arithmetic Operations](#)
- ⇒ **C++ While Loop**
- ◆ [C++ Do-While Loop](#)
- ◆ [C++ For Loop](#)
- ◆ [C++ ForEach](#)
- ◆ [C++ Continue](#)
- ◆ [C++ Break](#)
- ◆ [C++ Comments](#)
- ◆ [C++ Recursion](#)
- ◆ [C++ Try Catch](#)
- ◆ [C++ String Operations](#)
- ◆ [C++ Array Operations](#)
- ◆ [C++ Vector Operations](#)
- ◆ [C++ Input Output Operations](#)

▼ C++ Input Output Operations

◆ C++ Class

◆ C++ Programs