



# while Loop in C++

Get acquainted with the while loop and its basic syntax.

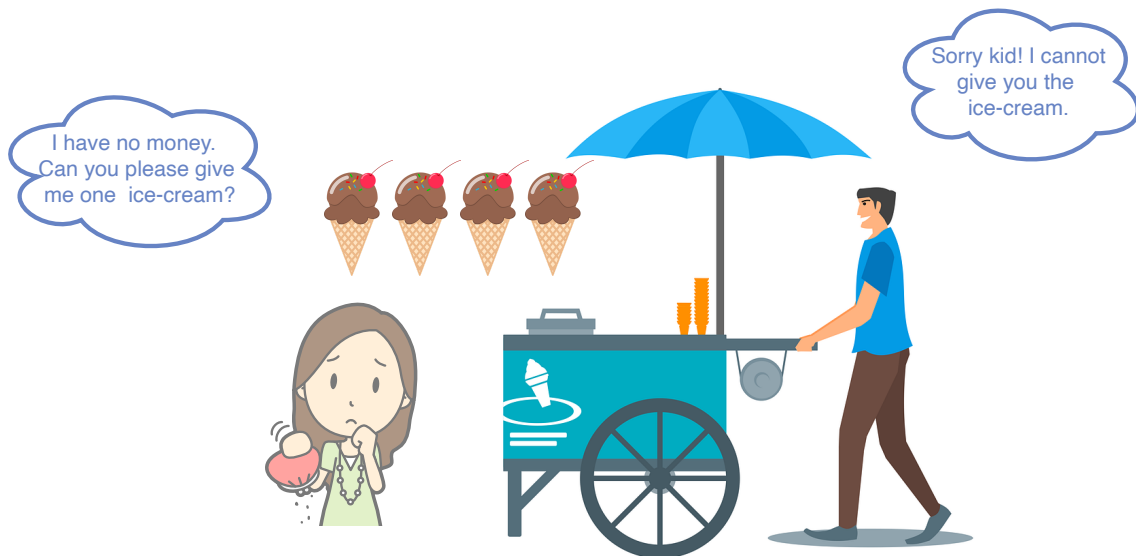
We'll cover the following



- Introduction
  - Syntax
  - Flowchart
  - Example program
  - Explanation

## Introduction#

Suppose you have \$20, and the price of an ice-cream is \$5. You want to keep buying the ice-cream until you have no money left. This task is repetitive, and you don't know in advance how many ice-creams you can buy.




5 of 5



In the era of programming, we can use the `while` loop to implement repetitive tasks.

*The **while loop** keeps executing a particular code block until the given condition is true. It does not know in advance how many times the loop body should be executed.*

 The condition in the `while` loop is evaluated before executing the statements inside its body. Therefore, the `while` loop is called an entry-controlled loop.



## Syntax #

Let's go over the syntax of the `while` loop.

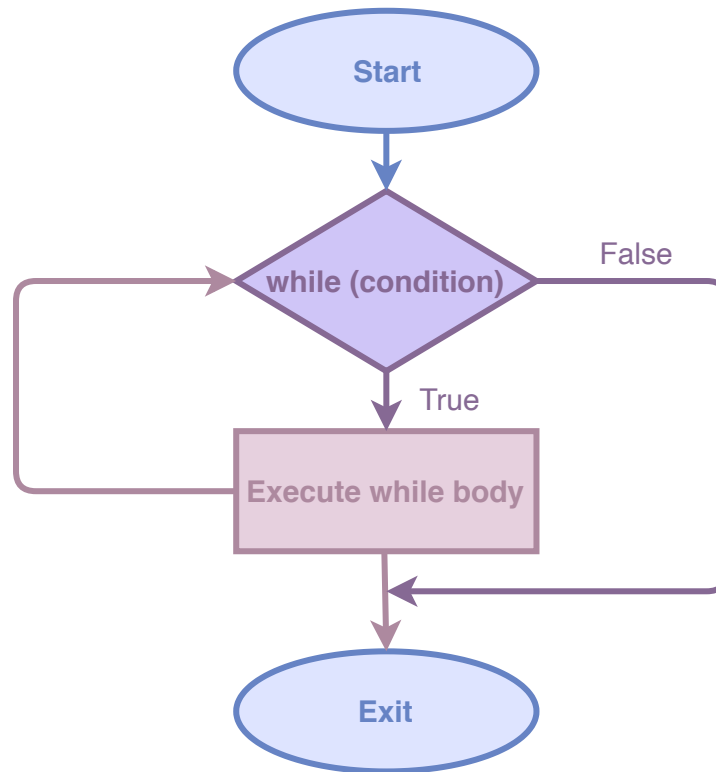
```
Keyword      Must return boolean
while ( condition ) {
    statement1;
    statement2;
    .
    .
    statement N;
}
```

while loop body

The general syntax of the `while` loop consists of a `while` keyword followed by a condition to be checked. The closing curly bracket is followed by the `while` keyword and the condition that is to be checked.

## Flowchart#

Let's look at the flowchart of the `while` loop.



- The while loop first evaluates the given condition.
- If the condition evaluates to true, the code inside the body of the while loop is executed.
- After that, the while loop again evaluates the condition. This process continues until the given condition remains true.

## Example program#

Let's translate the example given above into a C++ program.

Press the **RUN** button and see the output!

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
```



```

6 // Initialize the variable money
7 int money = 20;
8 // Initialize the variable icecream_price
9 int icecream_price = 5;
10 // Prints value of variables
11 cout << "Intial money = " << money << endl;
12 cout << "Ice-cream price = " << icecream_price << endl;
13 // Start of the while loop
14 while (money >= icecream_price){
15     // Body of the while loop
16     cout << "Buy an ice-cream" << endl;
17     money = money - icecream_price;
18     cout << "Remaining money = " << money << endl;
19 }
20 // End of the while loop
21 cout << "You can't buy an ice-cream" << endl;
22
23 return 0;
24 }

```



Output

1.15s

```

Intial money = 20
Ice-cream price = 5
Buy an ice-cream
Remaining money = 15
Buy an ice-cream
Remaining money = 10
Buy an ice-cream
Remaining money = 5
Buy an ice-cream

```



## Explanation#

**Line No. 7:** Initializes the value of money .

**Line No. 9:** Initializes the value of `icecream_price`.



**Line No. 11:** Prints the value of `money` to the console.

**Line No. 12:** Prints the value of `icecream_price` to the console.

**Line No. 14:** Checks if the value of `money` is greater than or equal to `icecream_price`. If true, then execute **Lines No. 16 to 19**. If false, then it executes **Line No. 21**.

**Line No. 16:** Prints Buy an ice-cream to the console.

**Line No. 17:** Subtracts `icecream_price` from the `money`.

**Line No. 18:** Prints the new value of `money`.

**Line No. 19:** Jumps to **Line No. 14**.

**Line No. 21:** Prints You can't buy an ice-cream to the console.



(/learn)

## Quiz



Q What is the output of the following code?

```
int main() {  
    int number = 1;  
    while (number <= 10) {  
        number = number + 1;  
    }  
    cout << "Number = " << number;  
}
```



AN Number = 10



Your Answer



**B) Number = 11**

Explanation

Initially number = 1 ,  
condition = true

After first iteration:  
number = 2 , condition =  
true

After second iteration:  
number = 3 , condition =  
true

After third iteration:  
number = 4 , condition =  
true

...

After tenth iteration:  
number = 11 , condition  
= false



**C) Number = 22**



**D) Number = 9**

Submit Answer

Reset Quiz ↻



This sums up our discussion of the `while` loop. Let's discuss the `do-while` loop in the upcoming lesson.

[← Back](#)[Next →](#)

Introduction to Loops

do-while Loop



Mark as Completed



Report an Issue