for Loop

Get introduced to the for loop in C++.

We'll cover the following

Introduction#

Suppose you have a coupon to buy five ice-creams free of cost. You know in advance how many free ice-creams you can buy.

I have no coupon.
Can you please give me free ice-cream?
Sorry kid! I cannot do that.
6 of 6

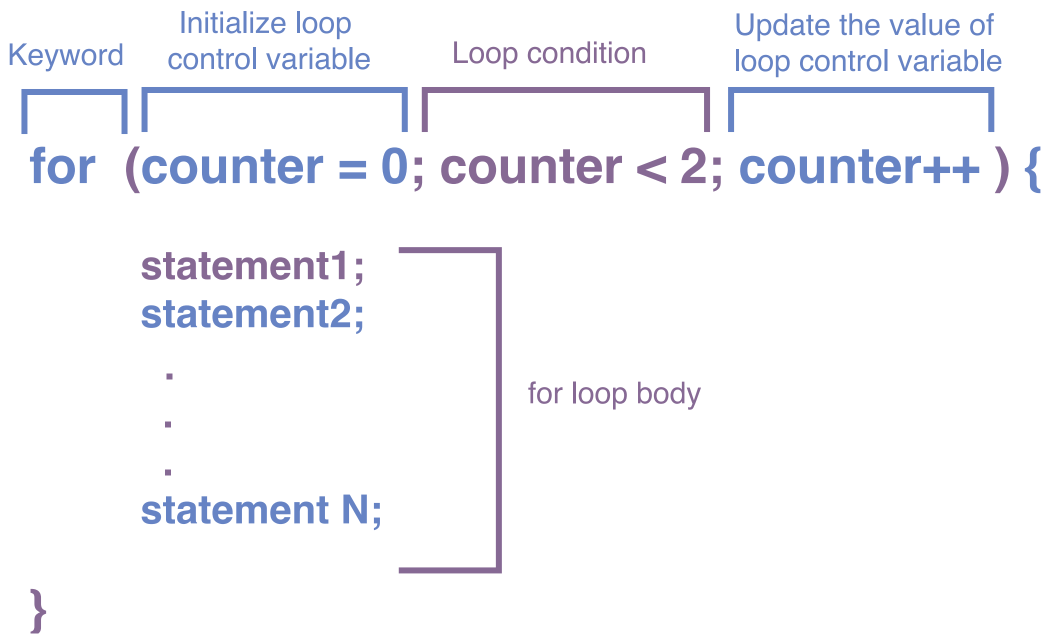
In the era of programming, we can use the `for` loop for such situations.

> *The `for loop` keeps executing a particular code block as long as the given condition is true. It knows in advance the number of times the loop body should be executed.*

> 📝 The `for loop` is a **count controlled loop** since the program knows in advance the number of times the loop body should be executed.

Syntax#

Let’s go over the syntax of the `for` loop.



```
for  (counter = 0; counter < 2; counter++ ) {

        statement1;
        statement2;
          .
          .
          .
        statement N;

}
```

Keyword
Initialize loop control variable
Loop condition
Update the value of loop control variable
for loop body

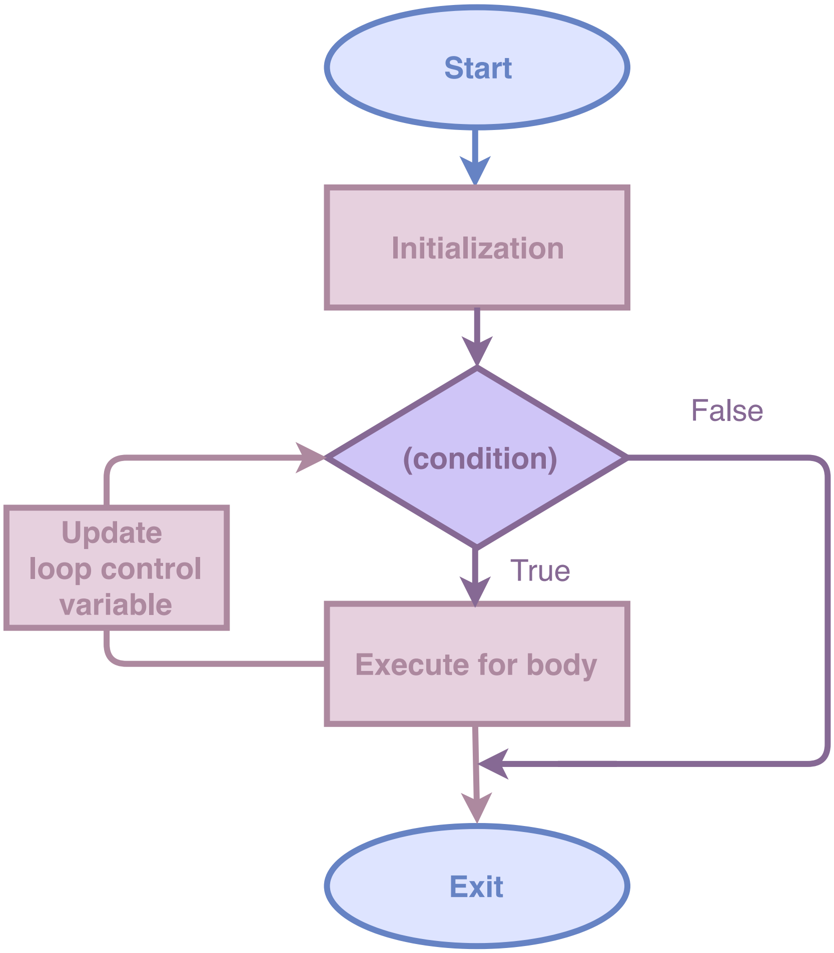
The general syntax of the for loop consists of a `for` keyword followed by round brackets `( )`. Inside the round brackets, the following three operations take place:

1. Initialization of the loop control variable
2. Evaluation of the loop condition
3. Increment or decrement of the loop control variable

The curly brackets `{ }` contain statements to be executed while the condition is `true`.

Flowchart#

Let’s look at the flowchart of the `for` loop.



- The `for` loop first initializes the loop control variable.
- Then, it evaluates the given condition.
- If the condition evaluates to `true`, the code inside the body of the `for loop` is executed.
- After that, it updates the value of the loop control variable and again evaluates the condition. This process continues as long as 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!

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

```
 1  #include <iostream>
 2
 3  using namespace std;
 4
 5  int main() {
 6    // Initialize variable icecream
 7    int icecream;
 8    // for loop start
 9    for (icecream = 5; icecream > 0; icecream--) {
10      // loop body
11      cout << "Number of free icecream = " << icecream << endl;
12      cout << "Buy an icecream" << endl;
13    }
14    // Exit loop
15    return 0;
16  }
```

Output 1.26s

```
Number of free icecream = 5
Buy an icecream
Number of free icecream = 4
Buy an icecream
Number of free icecream = 3
Buy an icecream
Number of free icecream = 2
Buy an icecream
Number of free icecream = 1
```

Explanation#

Line No. 7: Declares a variable `icecream`.

Line No. 9:

Line No. 9:

- **`icecream = 5:`** The initial value of `icecream` is set to `5`. Here, `icecream` is a loop control variable.

- **`icecream > 0:`** It is the loop continuation condition. It ensures the repetitive execution of the body of for loop until it evaluates to `true`.

- In the code above, loop statements are repeated until the value of the `icecream` is greater than `0`. When the loop condition evaluates to true, it executes the statements from **Lines No. 11 to 13**. After executing the loop block, it jumps back to **Line No. 9**. At this point, it updates the value of the icecream and again evaluates the condition.

- **`icecream--:`** This statement decrements the value of the `icecream` by `1`.

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

Line No. 12: Prints `Buy an icecream` to the console.

It's about time to wrap up our discussion of the `for` loop. Let's discuss infinite loops in the upcoming lesson.

Stay tuned!

← Back
do-while Loop

Next →
Infinite Loop

☑ Mark as Completed

Report an Issue