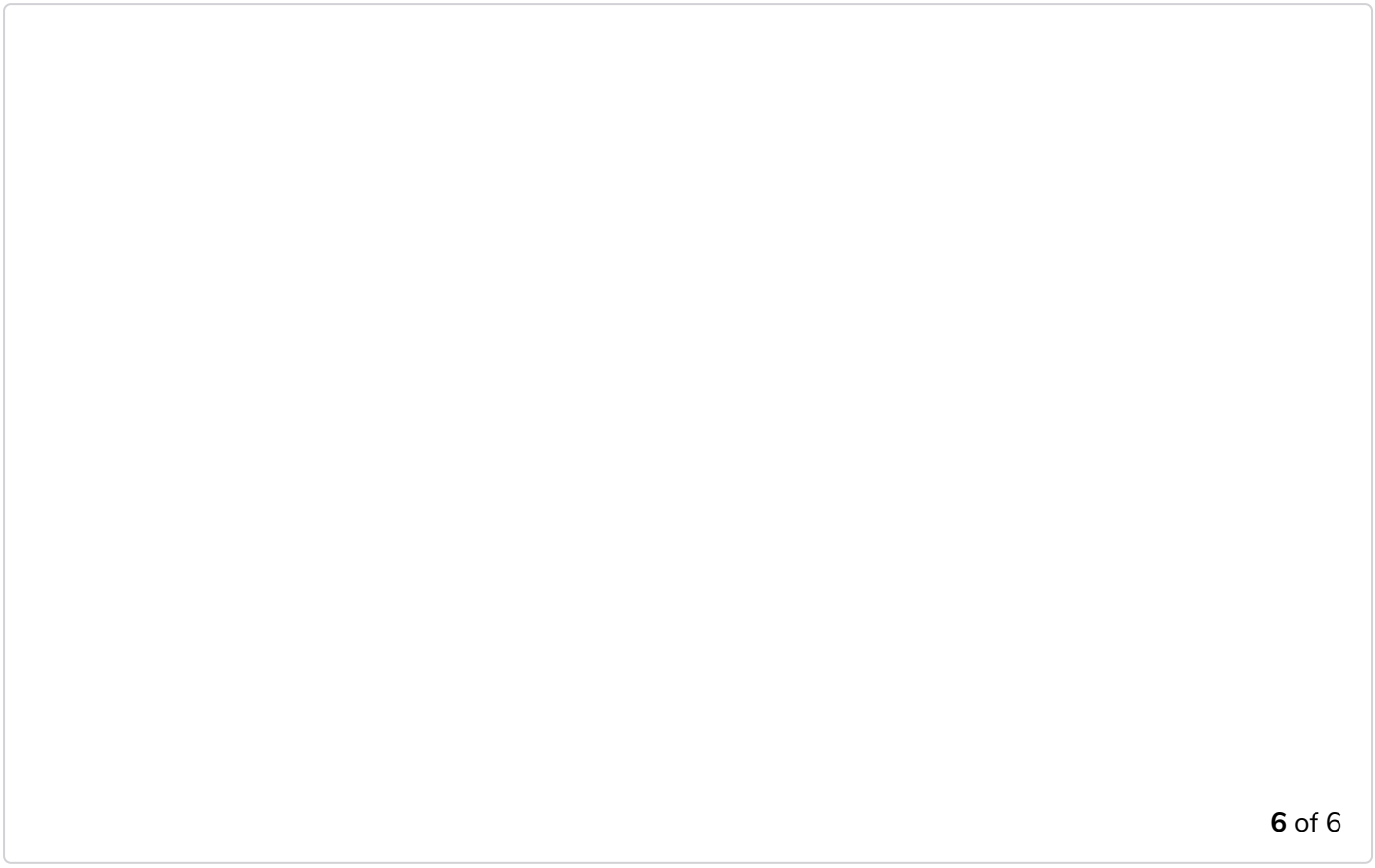
for Loop

In this lesson, you will be 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. Here, you know in advance how many free ice-creams you can buy.



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

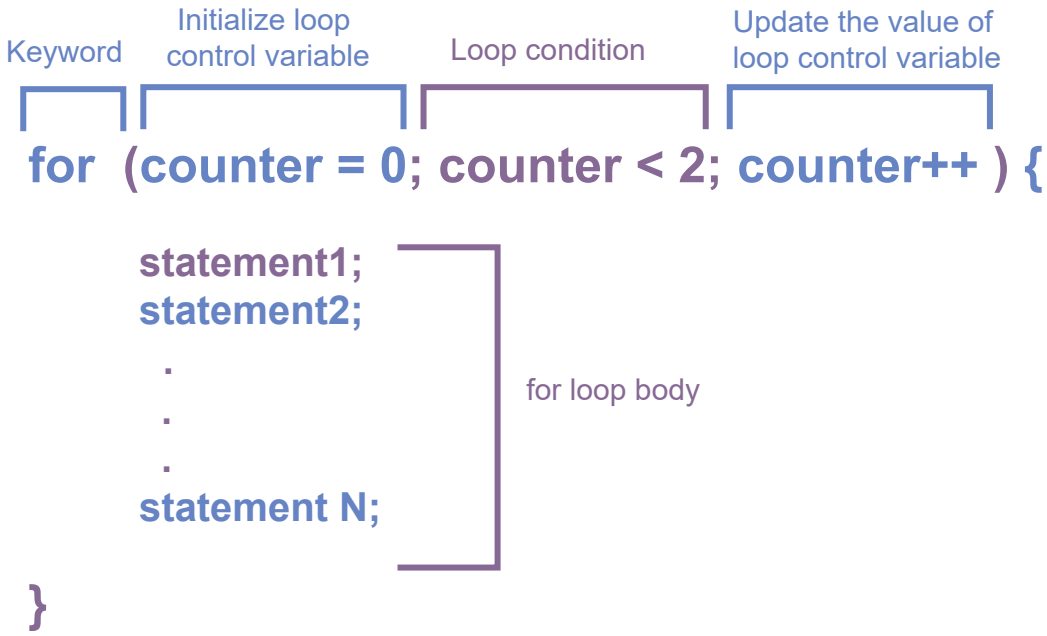
---

*The **for loop** keeps executing a particular code block until 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.



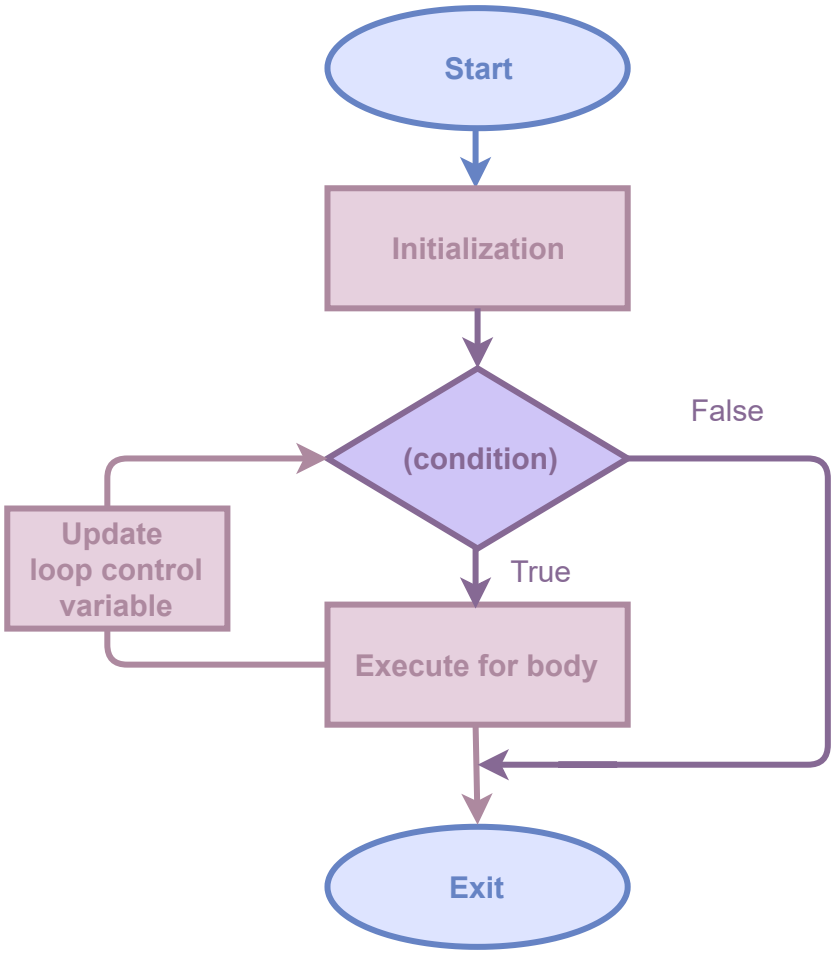
The general syntax of the for loop consists of a `for` keyword followed by round brackets `( )`. Inside the round brackets, three operations take place:

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 until 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 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!

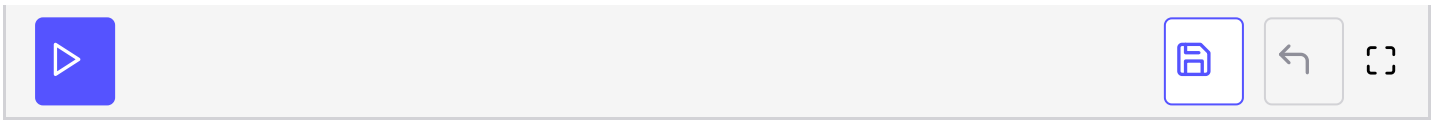
```
#include <iostream>

using namespace std;

int main() {
  // Initialize variable icecream
```

```
  int icecream;
  // for loop start
  for (icecream = 5; icecream > 0; icecream--) {

    // loop body
    cout << "Number of free icecream = " << icecream << endl;
    cout << "Buy an icecream" << endl;
  }
  // Exit loop
  return 0;
}
```



Explanation

Line No. 7: Declares a variable `icecream`

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

`for` loop knows in advance how many times the loop body should be executed.

Retake Quiz

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

Stay tuned!