For, While and do-while loops in C++ | C++ Tutorials for Begin...

Overview   **Q&A**   **Files**   **Announcements**

For, While and do-while loops in C++ | C++ Tutorials for Beginners #10

For, While and Do-While Loops in C++

In this series of our C++ tutorials, we will visualize for loop, while loop, and do-while loop in C++ language in this lecture. In our last lesson, we discussed the control structures, If-else statements, and switch statements in C++.

Loops in C++

Loops are block statements, which keeps on repeatedly executing until a specified condition is met. There are three types of loops in C++

- **For loop in C++**
- **While loop in C++**
- **Do While in C++**

For Loop in C++

For loop help us to run some specific code repeatedly until the specified condition is met. An example program **for the loop** is shown in figure 1.

```
for (int i = 0; i < 4; i++)
{
    /* code */
    cout<<i<<endl;
}
```

Figure 1: For Loop Program

As shown in figure 1, we created **for loop,** and inside its condition, there are three statements separated by a semicolon. The 1st statement is called "**initialization**", the 2nd statement is called "**condition**", and the 3rd statement is called "**updation**". After that, there is a loop body in which code is written, which needs to be repeated. Here is how our for loop will be executed:

- Initialize integer variable "**i**" with value "**0**"
- Check the condition if the value of the variable "**i**" is smaller than "**4**"
- If the condition is true go into loop body and execute the code
- Update the value of "**i**" by one
- Keep repeating this step until the condition gets false

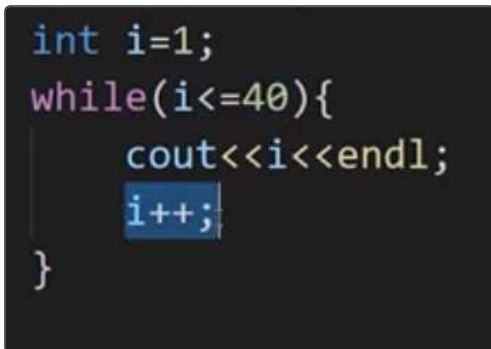The output for the following program is shown in figure 2.

Figure 2: For Loop Program Output

While Loop in C++

While loop helps us to run some specific code repeatedly until the specified condition is met. An example program of **while loop** is shown in figure 3.

```
int i=1;
while(i<=40){
    cout<<i<<endl;
    i++;
}
```
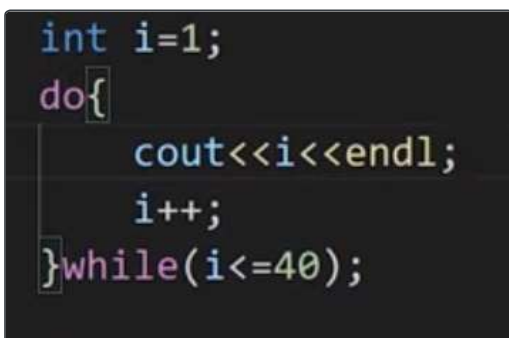
Figure 3: While Loop Program

As shown in figure 3, we created **a while loop,** and inside its condition, there is one statement. The statement is called "**condition**". Here is how our while loop will be executed:

- Initialize integer variable "**i**" with value "**1**"
- Check the condition if the value of the variable "**i**" is smaller or equal to "40."
- If the condition is true to go into loop body and execute the code
- Update the value of "**i**" by one
- Keep repeating this step until the condition gets false.

Do-While Loop in C++

The do-while loop helps us to run some specific code repeatedly until a specified condition is met. An example program of the **do-while loop** is shown in figure 1.

```
int i=1;
do{
    cout<<i<<endl;
    i++;
}while(i<=40);
```

Figure 4: Do-While Loop Program

As shown in figure 4, we created a **do-while loop,** and the syntax of the do-while loop is like write body with "**do**" keyword and at the end of body write "**while**" keyword with the condition. Here is how our do-while loop will be executed:

- Initialize integer variable "**i**" with value "**1**"

- Go into loop body and execute the code
- Check the condition if the value of the variable "**i**" is smaller or equal to "**40**"
- If the condition is true - go into loop body and execute the code
- Keep repeating this step until the condition gets false

Code as described/written in the video

```cpp
#include <iostream>

using namespace std;
int main()
{
    /*Loops in C++:
    There are three types of loops in C++:
        1. For loop
        2. While Loop
        3. do-While Loop
    */

    /*For loop in C++*/
    // int i=1;
    // cout<<i;
    // i++;

    // Syntax for for loop
    // for(initialization; condition; updation)
    // {
    //     loop body(C++ code);
    // }
```

← Previous

Next →