

while loop in C++ with examples

The execution of while-loop is similar to for-loop in C++. but wherein the while-loop up-dation is a part of the **body of loop** while in the **for-loop** is not.

[1. while-loop flow diagram in C++](#)

[1.1. Example of while loop in C++](#)

[2. nested while-loop in C++](#)

[2.1. Example of nested while loop in C++](#)

[3. while-loop with for-loop Example in C++](#)

[4. while-loop with if-else statement in C++](#)

In both loops, the execution sequence is in the same order. means, the first is the **variable initialization**, then the **condition is given** if the condition is true then the **body of loop** is executed then the initialization value is **updated or increased**. thus a while-loop is executed. but as soon as the condition is false the loop gets terminated.

syntax

```
initialization;

while(condition)

{

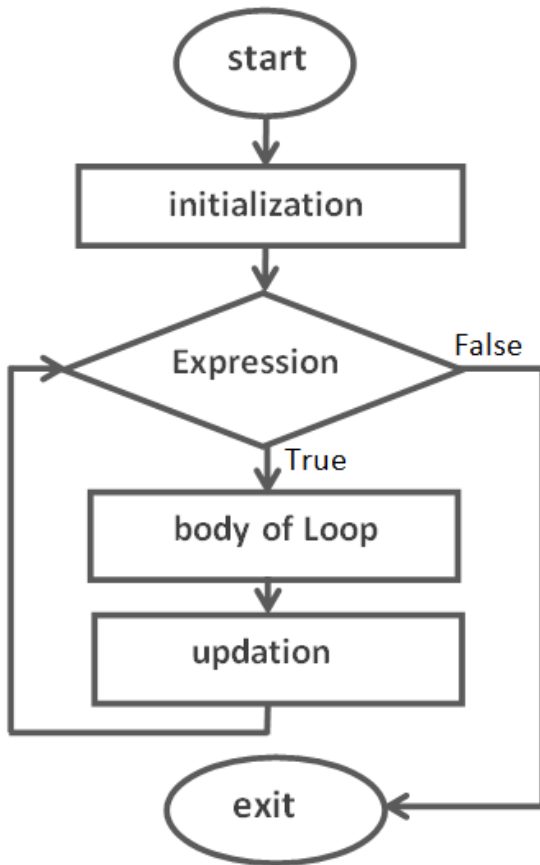
    body of loop;

    updation;

}
```

while-loop flow diagram in C++

a while-loop flow-diagram as follows,



Here is the program,

Example of while loop in C++

```
#include<iostream>

using namespace std;

int main()

{

    int num=5;
```

```
int i=0;

while(i<num)

{

    cout<<i+1<<"statement executed\n"; // body of loop

    i++;

}

return 0;

}
```

OUTPUT

```
1.statement executed

2.statement executed

3.statement executed

4.statement executed

5.statement executed
```

Explanation

In the Program, we declared loop size by **num = 5**, so here loop is executed 5 times,

After initialization variable **i** to **0**, then after the condition is checked which is becoming true (**0<5**) so **body of loop** is executed.

In the last, variable **i** value is increased by 1, Thus a while-loop will be executed.

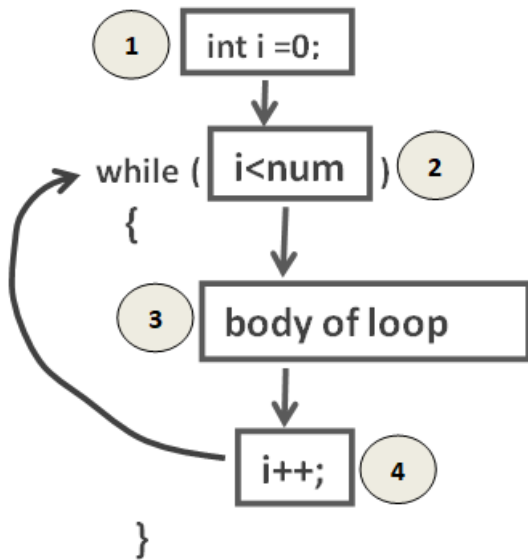
Now the whole procedure begins again until the condition becomes false (**5<5**) meaning that the condition will be checked again because this times variable value increased by 1 so the condition will be (**1<5**) and we know this time condition also becoming true so **body of loop** is executed again.

Here is another way,

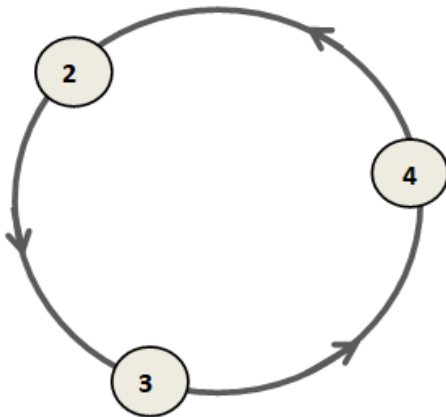
above program, implementation is given in the below table

initialization value (i=1)	condition check (i<=5)	body of loop executed
0	0<5	1.statement executed
1	1<5	2.statement executed
2	2<5	3.statement executed
3	3<5	4.statement executed
4	4<5	5.statement executed
5	5<5	Exit

the execution sequence of while loop will as,



in the while-loop loop following are will repeat, until the condition becomes false,



nested while-loop in C++

when a while-loop is body of another while-loop is called nested while loop in C++.

syntax

```
while (expression)
```

```
{  
  
// outer-while body  
  
while(expression)  
  
    {  
  
        // inner-while body  
  
        updation;  
  
    }  
  
    updation;  
  
|
```

Let's try with an example,

Example of nested while loop in C++

In the program outer while execute 3 times while inner while-loop execute 2 times,

```
#include<iostream.h>  
  
using namespace std;  
  
int main()  
  
{
```

```
int i=1,j;

while(i<=3)

{

//outer-while body

    cout<<i<<"outer-while loop\n";

    j=1;

    while(j<=2)

        {

            cout<<"\t"<<j<<"inner-while loop\n";

            j++;

        }

    i++;

}

return 0;

}
```

OUTPUT

1.outer-while loop

1.inner-while loop

2.inner-while loop

2.outer-while loop

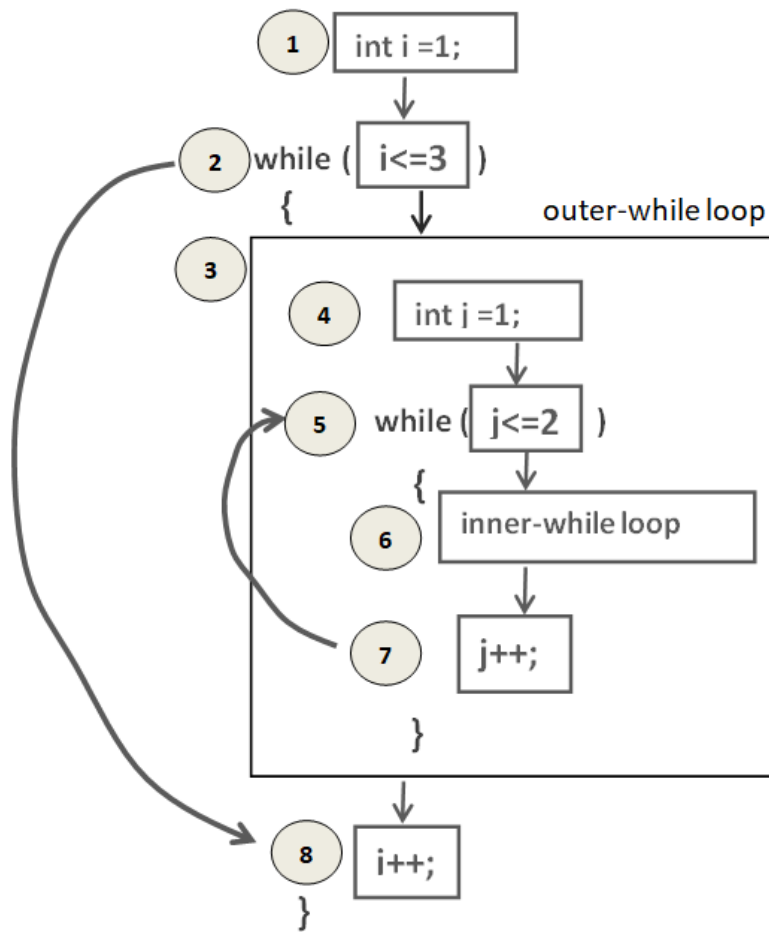
1.inner-while loop

2.inner-while loop

3.outer-while loop

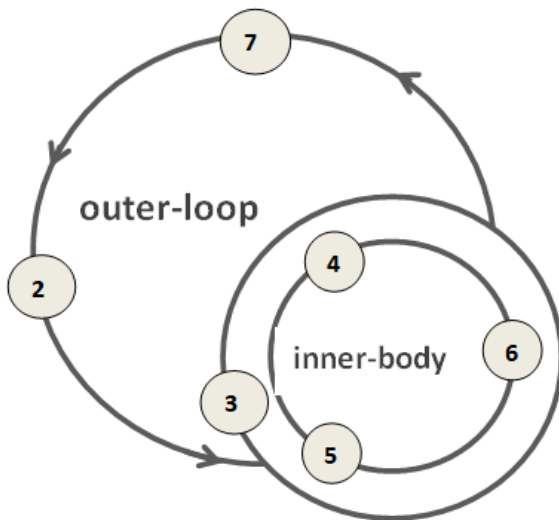
1.inner-while loop

2.inner-while loop



Explanation

repetition of nested while-loop will as follows,



while-loop with for-loop Example in C++

syntax

```
while (expression)

{

    // outer-while body

    for(.....)

    {

        // inner-while body updation;

    }

    updation;

|
```

while-loop with for-loop will execute same as the nested while-loop,

```
#include<iostream>

using namespace std;

int main()

{
```

```
int i=1,j;

while(i<=3)

{

//outer-while body

    cout<<i<<".while-loop executed\n";

    for(j=1; j<=2; j++)

        cout<<"\t"<<j<<".for-loop executed\n";

    i++;

}

return 0;

}
```

OUTPUT

1.while-loop executed

1.for-loop executed

2.for-loop executed

2.while-loop executed

1.for-loop executed

2.for-loop executed

3.while-loop executed

1.for-loop executed

2.for-loop executed

Let's try another example

```
#include<iostream>

using namespace std;

int main()

{

    int num;

    cout<<"Enter number: ";

    cin>>num;

    int row=1,col;
```

```
while(row<=10)

{

//while-loop body

    for(col=1; col<=num; col++)

        cout<<row<<"x"<<col<<"="<<row*col<<"\t";// for-loop body

    row++;

    cout<<endl;

}

return 0;

}
```

OUTPUT

Enter number: 6

```
1x1=1   1x2=2   1x3=3   1x4=4   1x5=5   1x6=6

2x1=2   2x2=4   2x3=6   2x4=8   2x5=10  2x6=12
```

```
3x1=3   3x2=6   3x3=9   3x4=12  3x5=15  3x6=18
4x1=4   4x2=8   4x3=12  4x4=16  4x5=20  4x6=24
5x1=5   5x2=10  5x3=15  5x4=20  5x5=25  5x6=30
6x1=6   6x2=12  6x3=18  6x4=24  6x5=30  6x6=36
7x1=7   7x2=14  7x3=21  7x4=28  7x5=35  7x6=42
8x1=8   8x2=16  8x3=24  8x4=32  8x5=40  8x6=48
9x1=9   9x2=18  9x3=27  9x4=36  9x5=45  9x6=54
10x1=10 10x2=20 10x3=30 10x4=40 10x5=50 10x6=60
```

while-loop with if-else statement in C++

in the below codes, we will print even and odd numbers in differently.

```
#include<iostream>

using namespace std;

int main()

{

    int count=0,even=0,odd=0,size=11;

    cout<<"even"<<"\t"<<"odd"<<endl;
```

```
while(count<=size)

{

//while body

    if(count%2==0) {

        cout<<count<<"\t";

        even+=count; //sum of even

    }

    else {

        cout<<count<<"\n";

        odd+=count; //sum of odd

    }

    count++;

}

cout<<"_____ \n";

cout<<even<<"\t"<<odd;

return 0;
```

```
}
```

OUTPUT

```
even    odd

0       1

2       3

4       5

6       7

8       9

10      11

_____

30      36
```

for-loop and while-loop, both are similar. C++ also provides a different loop is called the do-while loop.

Related Exercise

- [Print Fibonacci Series in C++](#)
- [Print large form Table in C++](#)