C++ continue statement with example

In C++, a continue statement is used to skip a statement in the program. using this statement with another control statement in C++, we can skip a statement in the program.

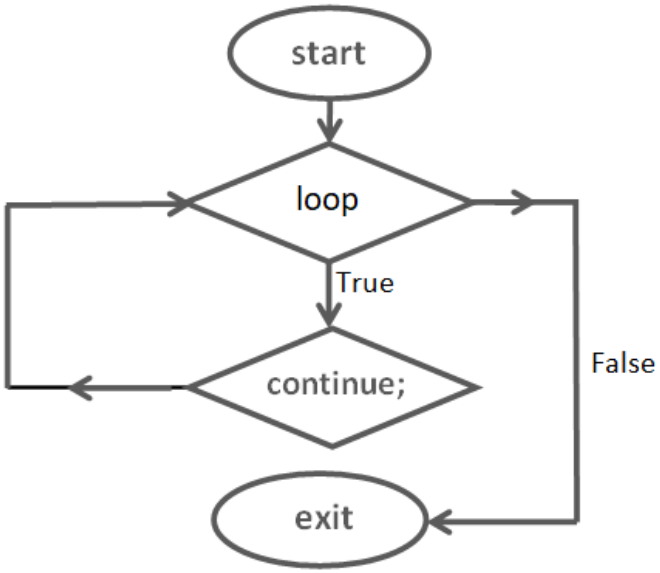
Remember, continue statement does not terminate the program it's only skipping the statement in the execution (if the given expression becomes true) that means the program does not terminate.

syntax

```
{

   continue;

   .........

}
```

Flow diagram of continue in C++

as you can see the above diagram continue-statement does not break the loop. After execution of continue-statement, execution goes back to the loop and loop will be executed



till then condition becomes false.

Example of continue statement in C++

In the below Program we Print number 0 – 10, in which we will skip a number using of continue statement.

```
#include<iostream>

using namespace std;

int main()

   {

     int num=10;
```

```
      for(int i = 0; i <= num; i++)

        {

           if(i==5)

              continue;

           cout<<"\t"<<i;

        }

   cout<<"Number 5 is skip";

  return 0;

  }
```

OUTPUT

```
0    1     2     3     4     6    7     8    9    10



Number 5 is skip.
```

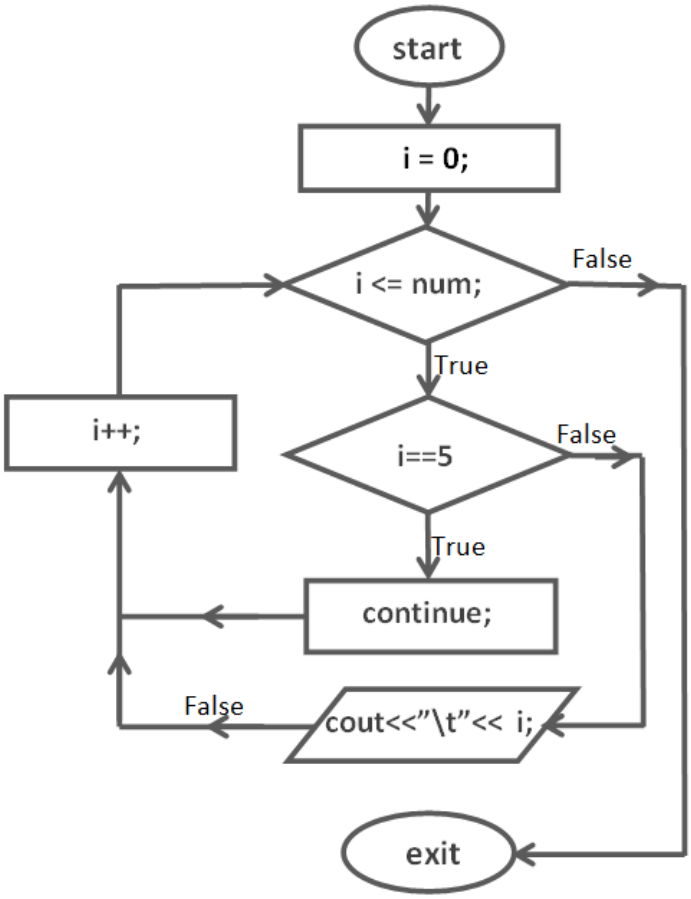
Explanation

In the Program, a **continue-statement** is body of an **if-statemen**t where an **if-statement** is a body of a **for-loop**.

Here the for-loop prints numbers from 0 to 10. But the condition will be checked before each number is printed. And as conditions in the if-statement become true. The `continue` statement will be executed and the number 5 skips(as condition given).

If here, the continue – statement would not have been used, if-else would be used to get the same output.

the following flow-diagram can explain this.



Print odd number using continue in C++

The program below is the same as above. But here we are printing odd numbers up to 20.

```
#include<iostream>

using namespace std;

  int main()

   {

     int num=20;



      for(int i = 0; i <= num; i++)

        {

           if(i%2==0)

             continue; // even number skip

           cout<<"\t"<<i; // odd number print

        }



return 0;

  }
```

OUTPUT

```
1  3  5  7  9  11  13  15  17   19
```

Explanation

Every time in the if-statement, even number will be skipped(as condition is given).

Print odd number table using continue in C++

In the below we printing only odd number table, here we use while loop as row and for loop as a column. others statements are similar to given the above programs.

```
#include<iostream>

using namespace std;

int main()

  {

    int num=20;

     int row=1,col;

   while(row<=10)

    {

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

         {

            if(col%2==0)
```

```
                continue;

            cout<<row*col<<"\t";

         }

      row++;

      cout<<endl;

    }

return 0;

 }
```

OUTPUT

```
1     3     5     7     9     11    13    15    17    19

2     6     10    14    18    22    26    30    34    38

3     9     15    21    27    33    39    45    51    57

4     12    20    28    36    44    52    60    68    76

5     15    25    35    45    55    65    75    85    95

6     18    30    42    54    66    78    90    102   114

7     21    35    49    63    77    91    105   119   133

8     24    40    56    72    88    104   120   136   152
```

```
9     27    45    63    81    99    117   135   153   171

10    30    50    70    90    110   130   150   170   190
```

difference between break and continue statement in C++

The break type statement is used to terminate a single statement or set of statements while the continue statement is used to skip the single statement.