Continue Statement in C++ with example

Continue statement is used inside loops. Whenever a continue statement is encountered inside a loop, control directly jumps to the beginning of the loop for next iteration, skipping the execution of statements inside loop's body for the current iteration.

Syntax of continue statement

```
continue;
```

Example: continue statement inside for loop

As you can see that the output is missing the value 3, however the for loop iterate though the num value 0 to 6. This is because we have set a condition inside loop in such a way, that the continue statement is encountered when the num value is equal to 3. So for this iteration the loop skipped the cout statement and started the next iteration of loop.

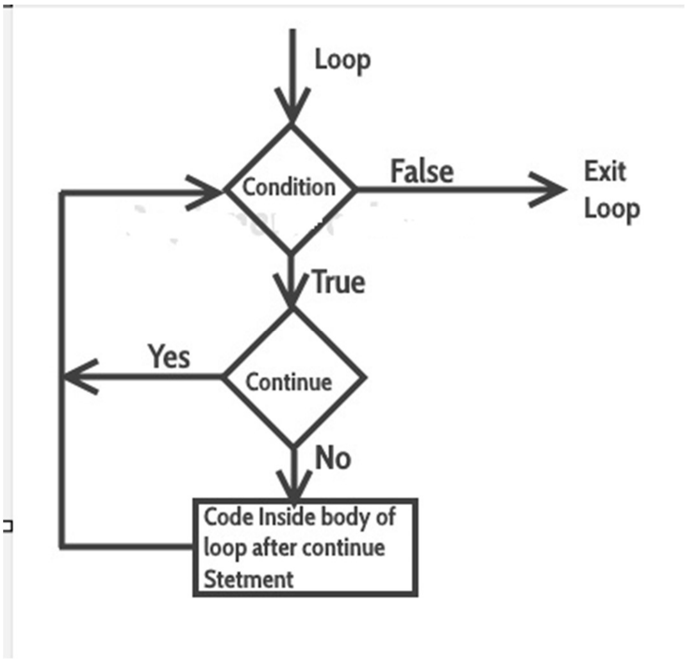
```
#include <iostream>
using namespace std;
int main(){
   for (int num=0; num<=6; num++) {
      /* This means that when the value of
       * num is equal to 3 this continue statement
       * would be encountered, which would make the
       * control to jump to the beginning of loop for
       * next iteration, skipping the current iteration
       */

      if (num==3) {
          continue;
      }
      cout<<num<<" ";
   }
   return 0;
}
```

Output:

```
0 1 2 4 5 6
```

Flow Diagram of Continue Statement



Example: Use of continue in While loop

```
#include <iostream>
using namespace std;
int main(){
   int j=6;
   while (j >=0) {
      if (j==4) {
         j--;
         continue;
      }
      cout<<"Value of j: "<<j<<endl;
      j--;
   }
   return 0;
}
```

Output:

```
Value of j: 6
Value of j: 5
Value of j: 3
Value of j: 2
Value of j: 1
Value of j: 0
```

Example of continue in do-While loop

```
#include <iostream>
using namespace std;
int main(){
   int j=4;
   do {
      if (j==7) {
         j++;
         continue;
      }
      cout<<"j is: "<<j<<endl;
      j++;
   }while(j<10);
   return 0;
}
```

Output:

```
j is: 4
j is: 5
j is: 6
j is: 8
j is: 9
```