C continue statement

The **continue statement** in C language is used to bring the program control to the beginning of the loop. The continue statement skips some lines of code inside the loop and continues with the next iteration. It is mainly used for a condition so that we can skip some code for a particular condition.

Syntax:

1. //loop statements
2. **continue**;
3. //some lines of the code which is to be skipped

Continue statement example 1

1. #include<stdio.h>
2. **void** main ()
3. {
4. **int** i = 0;
5. **while**(i!=10)
6. {
7. printf("%d", i);
8. **continue**;
9. i++;
10. }
11. }

**Output**

infinite loop

Continue statement example 2

1. #include<stdio.h>
2. **int** main(){
3. **int** i=1;//initializing a local variable
4. //starting a loop from 1 to 10
5. **for**(i=1;i<=10;i++){
6. **if**(i==5){//if value of i is equal to 5, it will continue the loop
7. **continue**;
8. }
9. printf("%d \n",i);
10. }//end of for loop
11. **return** 0;
12. }

**Output**

1

2

3

4

6

7

8

9

10

As you can see, 5 is not printed on the console because loop is continued at i==5.

C continue statement with inner loop

In such case, C continue statement continues only inner loop, but not outer loop.

1. #include<stdio.h>
2. **int** main(){
3. **int** i=1,j=1;//initializing a local variable
4. **for**(i=1;i<=3;i++){
5. **for**(j=1;j<=3;j++){
6. **if**(i==2 && j==2){
7. **continue**;//will continue loop of j only
8. }
9. printf("%d %d\n",i,j);
10. }
11. }//end of for loop
12. **return** 0;
13. }

**Output**

1 1

1 2

1 3

2 1

2 3

3 1

3 2

3 3

As you can see, 2 2 is not printed on the console because inner loop is continued at i==2 and j==2.