# C break statement

The **break statement** in C language is used to break the execution of loop (while, do while and for) and switch case.

In case of *inner loops*, it terminates the control of inner loop only.

There can be two usage of C break keyword:

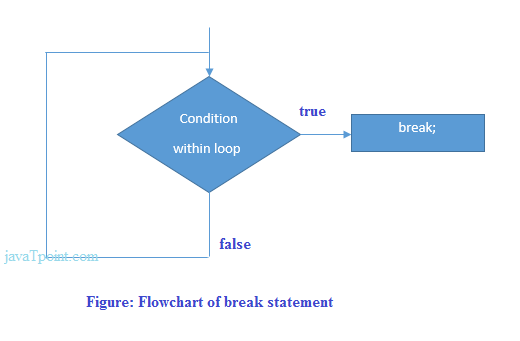
1. With switch case
2. With loop

#### Syntax:

1. jump-statement;
2. **break**;

The jump statement in c break syntax can be while loop, do while loop, for loop or switch case.

#### Flowchart of break in c



## **Example of C break statement with switch case**

Click here to see the example of [C break with switch statement](http://www.javatpoint.com/c-switch).

## **Example of C break statement with loop**

1. #include <stdio.h>
2. #include <conio.h>
3. **void** main(){
4. **int** i=1;//initializing a local variable
5. clrscr();
7. //starting a loop from 1 to 10
8. **for**(i=1;i<=10;i++){
9. printf("%d \n",i);
10. **if**(i==5){//if value of i is equal to 5, it will break the loop
11. **break**;
12. }
13. }//end of for loop
15. getch();
16. }

#### Output

1

2

3

4

5

As you can see on console output, loop from 1 to 10 is not printed after i==5.

## **C break statement with inner loop**

In such case, it breaks only inner loop, but not outer loop.

1. #include <stdio.h>
2. #include <conio.h>
3. **void** main(){
4. **int** i=1,j=1;//initializing a local variable
5. clrscr();
7. **for**(i=1;i<=3;i++){
8. **for**(j=1;j<=3;j++){
9. printf("%d &d\n",i,j);
10. **if**(i==2 && j==2){
11. **break**;//will break loop of j only
12. }
13. }
14. }//end of for loop
16. getch();
17. }

#### Output

1 1

1 2

1 3

2 1

2 2

3 1

3 2

3 3

As you can see the output on console, 2 3 is not printed because there is break statement after printing i==2 and j==2. But 3 1, 3 2 and 3 3 is printed because break statement works for inner loop only.

# C continue statement

The **continue statement** in C language is used to continue the execution of loop (while, do while and for). It is used with *if condition* within the loop.

In case of *inner loops*, it continues the control of inner loop only.

#### Syntax:

1. jump-statement;
2. **continue**;

The *jump statement* can be while, do while and for loop.

## **Example of continue statement in c**

1. #include <stdio.h>
2. #include <conio.h>
3. **void** main(){
4. **int** i=1;//initializing a local variable
5. clrscr();
7. //starting a loop from 1 to 10
8. **for**(i=1;i<=10;i++){
9. **if**(i==5){//if value of i is equal to 5, it will continue the loop
10. **continue**;
11. }
12. printf("%d \n",i);
13. }//end of for loop
15. getch();
16. }

#### 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. #include <conio.h>
3. **void** main(){
4. **int** i=1,j=1;//initializing a local variable
5. clrscr();
7. **for**(i=1;i<=3;i++){
8. **for**(j=1;j<=3;j++){
9. **if**(i==2 && j==2){
10. **continue**;//will continue loop of j only
11. }
12. printf("%d &d\n",i,j);
13. }
14. }//end of for loop
16. getch();
17. }

#### 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.

# C goto statement

The goto statement is known as jump statement in C language. It is used to unconditionally jump to other label. It transfers control to other parts of the program.

It is rarely used today because it makes program less readable and complex.

Syntax:

1. **goto** label;

## **goto example**

Let's see a simple example to use goto statement in C language.

1. #include <stdio.h>
2. #include <conio.h>
3. **void** main() {
4. **int** age;
5. clrscr();
6. ineligible:
7. printf("You are not eligible to vote!\n");
9. printf("Enter you age:\n");
10. scanf("%d", &age);
11. **if**(age<18)
12. **goto** ineligible;
13. **else**
14. printf("You are eligible to vote!\n");
16. getch();
17. }

Output:

You are not eligible to vote!

Enter you age:

11

You are not eligible to vote!

Enter you age:

44

You are eligible to vote!

# Type Casting in C

Type casting allows us to convert one data type into other. In C language, we use cast operator for type casting which is denoted by (type).

Syntax:

1. (type)value;

#### *Note: It is always recommended to convert lower value to higher for avoiding data loss.*

**Without Type Casting:**

1. **int** f= 9/4;
2. printf("f : %d\n", f );//Output: 2

**With Type Casting:**

1. **float** f=(**float**) 9/4;
2. printf("f : %f\n", f );//Output: 2.250000

## **Type Casting example**

Let's see a simple example to cast int value into float.

1. #include <stdio.h>
2. #include <conio.h>
3. **void** main(){
4. clrscr();
6. **float** f= (**float**)9/4;
7. printf("f : %f\n", f );
9. getch();
10. }

Output:

f : 2.250000