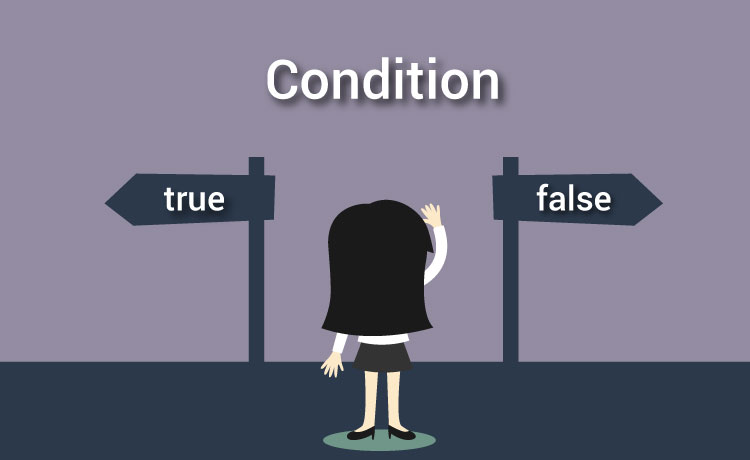
C if...else Statement



## C if statement

if (testExpression)

{

// statements

}

### Example #1: C if statement

// Program to display a number if user enters negative number

// If user enters positive number, that number won't be displayed

#include <stdio.h>

int main()

{

int number;

printf("Enter an integer: ");

scanf("%d", &number);

// Test expression is true if number is less than 0

if (number < 0)

{

printf("You entered %d.\n", number);

}

printf("You entered –ve Number.");

return 0;

}

**Output :**

Enter an integer: -2

You entered -2.

You entered –ve Number.

**Output 2**

Enter an integer: 5

The if statement in C programming is easy.

## C if...else statement

The if...else statement executes some code if the test expression is true (nonzero) and some other code if the test expression is false (0).

### Syntax of if...else

if (testExpression) {

// codes inside the body of if

}

else {

// codes inside the body of else

}

### Example #2: C if...else statement

// Program to check whether an integer entered by the user is odd or even

#include <stdio.h>

int main()

{

int number;

printf("Enter an integer: ");

scanf("%d",&number);

// True if remainder is 0

if( number%2 == 0 )

printf("%d is an even integer.",number);

else

printf("%d is an odd integer.",number);

return 0;

}

**Output**

Enter an integer: 7

7 is an odd integer.

## Nested if...else statement (if...elseif....else Statement)

The if...else statement executes two different codes depending upon whether the test expression is true or false. Sometimes, a choice has to be made from more than 2 possibilities.

The nested if...else statement allows you to check for multiple test expressions and execute different codes for more than two conditions.

### Example #3: C Nested if...else statement

// Program to relate two integers using =, > or <

#include <stdio.h>

int main()

{

int number1, number2;

printf("Enter two integers: ");

scanf("%d %d", &number1, &number2);

//checks if two integers are equal.

if(number1 == number2)

{

printf("Result: %d = %d",number1,number2);

}

//checks if number1 is greater than number2.

else if (number1 > number2)

{

printf("Result: %d > %d", number1, number2);

}

// if both test expression is false

else

{

printf("Result: %d < %d",number1, number2);

}

return 0;

}

**Output**

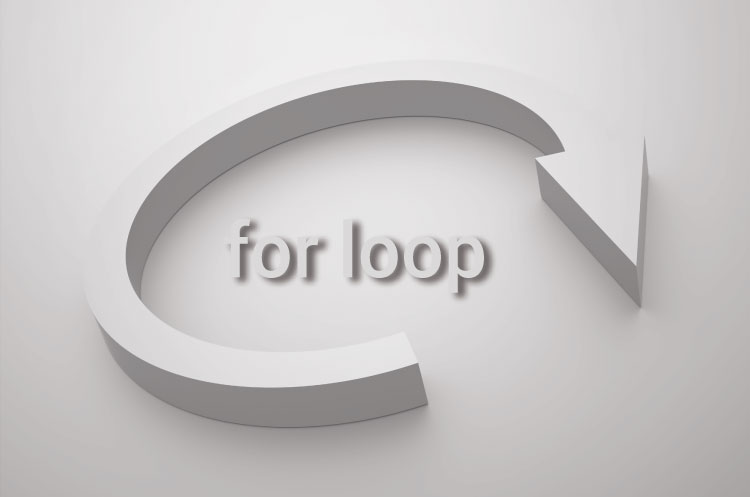
Enter two integers: 12

23

Result: 12 < 23

# C Programming for Loop

**Loops are used in programming to repeat a specific block of code.**



Loops are used in programming to repeat a specific block until some end condition is met. There are three loops in C programming:

1. for loop
2. [while loop](https://www.programiz.com/c-programming/c-do-while-loops)
3. [do...while loop](https://www.programiz.com/c-programming/c-do-while-loops)

## for Loop

 The syntax of for loop is:

for (initializationStatement; testExpression; updateStatement)

{

// codes

}

### Example: for loop

// Program to calculate the sum of first n natural numbers

// Positive integers 1,2,3...n are known as natural numbers

#include <stdio.h>

int main()

{

int num, count, sum = 0;

printf("Enter a positive integer: ");

scanf("%d", &num);

// for loop terminates when n is less than count

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

{

sum += count;

}

printf("Sum = %d", sum);

return 0;

}

**Output**

Enter a positive integer: 10

Sum = 55

# C Programming while and do...while Loop

L**oops are used in programming to repeat a specific block of code**

## while loop

The syntax of a while loop is:

while (testExpression)

{

//codes

}

where, testExpression checks the condition is true or false before each loop.

### Example #1: while loop

// Program to find factorial of a number

// For a positive integer n, factorial = 1\*2\*3...n

#include <stdio.h>

int main()

{

int number;

long long factorial;

printf("Enter an integer: ");

scanf("%d",&number);

factorial = 1;

// loop terminates when number is less than or equal to 0

while (number > 0)

{

factorial \*= number; // factorial = factorial\*number;

--number;

}

printf("Factorial= %lld", factorial);

return 0;

}

**Output**

Enter an integer: 5

Factorial = 120

## do...while loop

The do..while loop is similar to the while loop with one important difference. The body of do...while loop is executed once, before checking the test expression. Hence, the do...while loop is executed at least once.

### do...while loop Syntax

do

{

// codes

}

while (testExpression);

### Example #2: do...while loop

// Program to add numbers until user enters zero

#include <stdio.h>

int main()

{

double number, sum = 0;

// loop body is executed at least once

do

{

printf("Enter a number: ");

scanf("%lf", &number);

sum += number;

}

while(number != 0.0);

printf("Sum = %f",sum);

return 0;

}

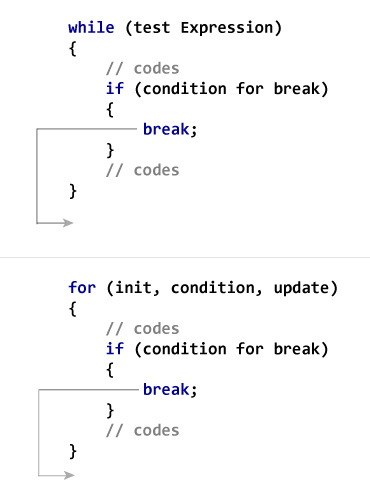
# C Programming break and continue Statement

**you will learn how to use break and continue statements to alter the program flow of loops.**

## break Statement

The break statement terminates the loop ([for](https://www.programiz.com/c-programming/c-for-loop), [while and do...while loop](https://www.programiz.com/c-programming/c-do-while-loops)) immediately when it is encountered. The break statement is used with decision making statement such as [if...else](https://www.programiz.com/c-programming/c-if-else-statement).

### How break statement works?



### Example #1: break statement

// Program to calculate the sum of maximum of 10 numbers

// Calculates sum until user enters positive number

# include <stdio.h>

int main()

{

int i;

double number, sum = 0.0;

for(i=1; i <= 10; ++i)

{

printf("Enter a n %d: ",i);

scanf("%lf",&number);

// If user enters negative number, loop is terminated

if(number < 0.0)

{

break;

}

sum += number; // sum = sum + number;

}

printf("Sum = %f",sum);

return 0;

}

**Output**

Enter a n1: 2.4

Enter a n2: 4.5

Enter a n3: 3.4

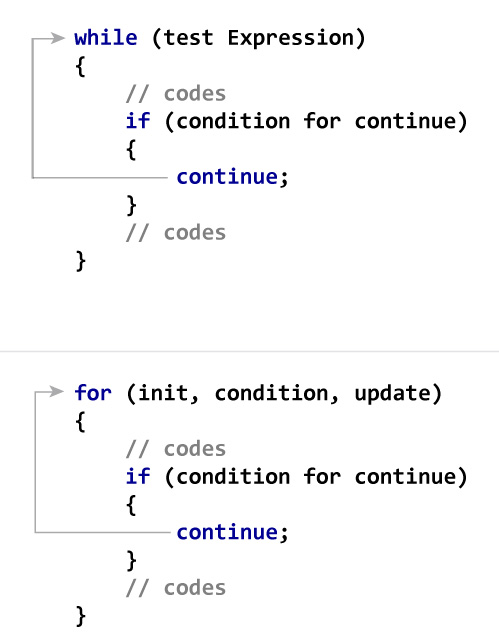
Enter a n4: -3

Sum = 10.30

## continue Statement

The continue statement skips some statements inside the loop. The continue statement is used with decision making statement such as if...else.

### How continue statement works?



### Example #2: continue statement

// Program to calculate sum of maximum of 10 numbers

// Negative numbers are skipped from calculation

# include <stdio.h>

int main()

{

int i;

double number, sum = 0.0;

for(i=1; i <= 10; ++i)

{

printf("Enter a n%d: ",i);

scanf("%lf",&number);

// If user enters negative number, loop is terminated

if(number < 0.0)

{

continue;

}

sum += number; // sum = sum + number;

}

printf("Sum = %.20f",sum);

return 0;

}

Enter a n1: 1.1

Enter a n2: 2.2

Enter a n3: 5.5

Enter a n4: 4.4

Enter a n5: -3.4

Enter a n6: -45.5

Enter a n7: 34.5

Enter a n8: -4.2

Enter a n9: -1000

Enter a n10: 12

Sum = 59.70

# C switch...case Statement



The if..else..if ladder allows you to execute a block code among many alternatives. If you are checking on the value of a single variable in if...else...if, it is better to use switch statement.

The switch statement is often faster than nested if...else (not always). Also, the syntax of switch statement is cleaner and easy to understand.

## Syntax of switch...case

switch (n)

​{

case constant1:

// code to be executed if n is equal to constant1;

break;

case constant2:

// code to be executed if n is equal to constant2;

break;

.

.

.

default:

// code to be executed if n doesn't match any constant

}// Following is a simple program to demonstrate syntax of switch.

#include <stdio.h>

int main()

{

int x = 2;

switch (x)

{

case 1: printf("Choice is 1");

break;

case 2: printf("Choice is 2");

break;

case 3: printf("Choice is 3");

break;

default: printf("Choice other than 1, 2 and 3");

break;

}

return 0;

}

Output:

Choice is 2

1. ***The expression used in switch must be integral type ( int, char and enum).*** Any other type of expression is not allowed.

|  |
| --- |
| **// float is not allowed in switch**  **#include <stdio.h>**  **int main()**  **{**  **float x = 1.1;**  **switch (x)**  **{**  **case 1.1: printf("Choice is 1");**  **break;**  **default: printf("Choice other than 1, 2 and 3");**  **break;**  **}**  **return 0;** |

Output:

Compiler Error: switch quantity not an integer