



C PROGRAMMING

Flow Controls

What are Flow Controls?

The control statements used in the C language help a user to specify a program control's flow. In simpler words, the control statements help users specify the order of execution of the instructions present in a program. These make it possible for the program to make certain decisions, perform various tasks repeatedly, or even jump from any one section of the code to a different section.



List of flow Controls

- if statement
- if...else statement
- for loop
- while loop
- break and continue

Syntax of if statement

- if (test expression)
 {
 Code
 }

How if statement works

- The if statement evaluates the test expression inside the parenthesis ().
- If the test expression is evaluated to true, statements inside the body of **if** are executed.
- If the test expression is evaluated to false, statements inside the body of **if** are not executed.

How if statement works


Expression is true.

```
int test = 5;  
  
if (test < 10)  
{  
    // codes  
}  
  
// codes after if
```

A flowchart showing a horizontal line from the 'if' statement to a vertical line, which then has an arrow pointing down to the code block inside the if statement.

Expression is false.

```
int test = 5;  
  
if (test > 10)  
{  
    // codes  
}  
  
// codes after if
```

A flowchart showing a horizontal line from the 'if' statement to a vertical line, which then has an arrow pointing down to the code block after the if statement.

Example of if statement

Program to display a number if it is negative

```
int number;  
printf("Enter an integer: ");  
scanf("%d", &number);  
  
if (number < 0)  
{  
    printf("You entered a negative number");  
}
```

Syntax of if...else statement

```
if (test expression)
```

```
{
```

```
    Code
```

```
}
```

```
else
```

```
{
```

```
    Code
```

```
}
```


How if...else statement works?

- If the test expression is evaluated to **true**,
statements inside the body of **if** are executed.
statements inside the body of **else** are skipped from execution.
- If the test expression is evaluated to **false**,
statements inside the body of **else** are executed.
statements inside the body of **if** are skipped from execution.

How if...else statement works?

Expression is true.

```
int test = 5;

if (test < 10)
{
    // body of if
}
else
{
    // body of else
}
```

A flowchart illustrating the execution of an if...else statement when the condition is true. It starts with the variable declaration 'int test = 5;'. An arrow points to the 'if (test < 10)' condition. From there, the path goes down into the 'if' block, then up to the 'else' block, and finally down to the end of the code block.

Expression is false.

```
int test = 5;

if (test > 10)
{
    // body of if
}
else
{
    // body of else
}
```

A flowchart illustrating the execution of an if...else statement when the condition is false. It starts with the variable declaration 'int test = 5;'. An arrow points to the 'if (test > 10)' condition. From there, the path goes down into the 'if' block, then up to the 'else' block, and finally down to the end of the code block.

Example of if...else statement

Program to display a number if it is positive or negative

```
int number;  
printf("Enter an integer: ");  
scanf("%d", &number);  
  
if (number < 0)  
{  
    printf("You entered a negative number");  
}  
else  
{  
    printf("You entered a positive number");  
}
```