

Lecture 4. Decision Control Statements in C

if statement

if-else and **else-if** statements

if Statement

if statement is used to make a decision based on a condition.

Condition can be an expression (examples: `if(x + y)` or `if(num1>5)`) or a direct value (example: `if(5)`)

If the expression produces zero, or direct value is zero, the condition becomes FALSE.
For all other values, the condition is TRUE.

Examples:

`if(10)` is TRUE

`if(0)` is FALSE

`if(x + y)` is TRUE if `(x+y)` is not equal to zero

`if(num1 > 5)` is TRUE if the value of variable **num** is greater than 5

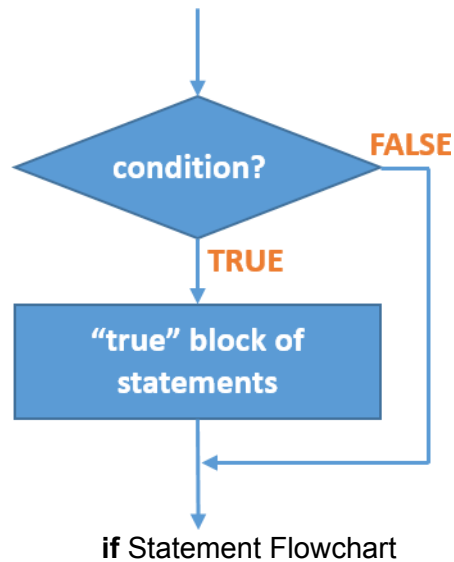
Simple if statement

The **if** statement evaluates the condition. If it is **TRUE**, it executes the block of code that follows. If the condition is **FALSE**, it will skip this code.

Used when there is only one option that is executed or not based on a condition.

Syntax

```
FALSE if (condition) TRUE
      {
        //statements to be executed if condition is TRUE
      }
      statements;
```



Example:

```
if (number < 0)    //checks if number is a negative number
{
    printf("You entered negative number: %d", number);
}
```

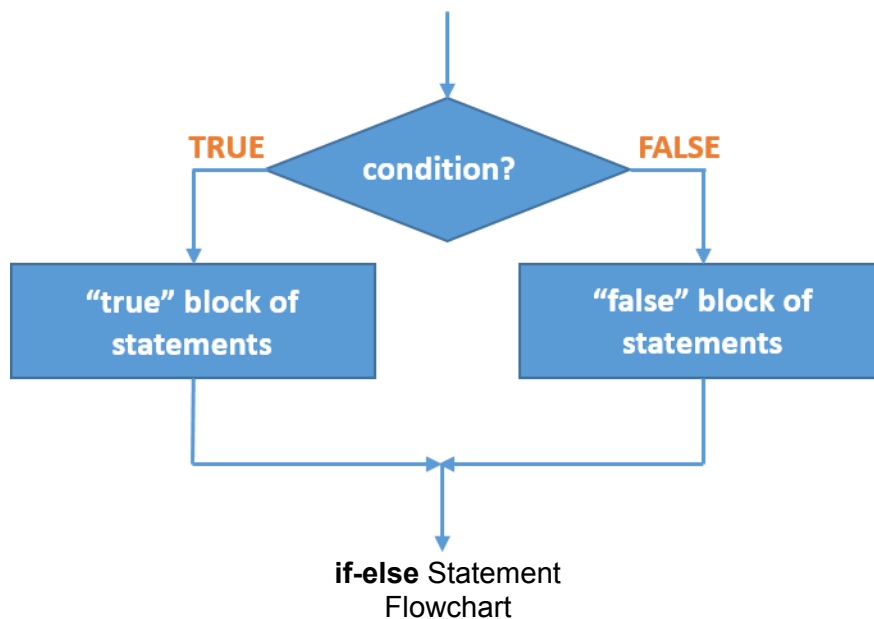
if - else statement

The **if-else** evaluates the condition. If it is TRUE, it executes the block of code that follows. If the condition is FALSE, it executes the second block of code.

Syntax

```

FALSE
if (condition) TRUE
{
    //statements to be executed if condition is TRUE ←
}
else
{
    //statements to be executed if condition is FALSE
}
statements;
    
```



Example:

```

if (number < 0)    //checks if a number is a negative number
{
    printf("You entered negative number: %d", number);
}
else
{
    printf("You entered zero or positive number: %d", number);
}
    
```


Nested if statement

The if-else statement is **inside** another if-else statement

Syntax

```
if (condition1)
{
    if (condition2)
    {
        //statements to be executed if condition1 & condition2 are TRUE
    }
    else
    {
        /*statements to be executed if condition1 is TRUE and condition2
        is FALSE */
    }
}
else
{
    //statements to be executed if condition1 is false
}
```

Example:

```
if (number < 0)           //checks if number is a negative number
{
    printf("You entered negative number: %d", number);
}
else
{
    if (number == 0)       //checks if number is equal to zero
    {
        printf("You entered zero");
    }
    else
    {
        printf("You entered positive number: %d", number);
    }
}
```

if – else – if statement (if-else ladder)

The if-else ladder allows for multiple test expressions to be checked and to execute different codes for more than two conditions.

Syntax

```
if (condition1)
{
//statements to be executed if condition1 is TRUE
}
else if (condition2)
{
//statements to be executed if condition1 is FALSE and condition2 is TRUE
}
else if (condition3)
{
//statements to be executed if condition1&condition2 are FALSE,
// and condition3 is TRUE
}
...
else
{
//statement to be executed if all conditions are FALSE
}
```

Example:

```
if(number < 0)           //checks if number is a negative number
{
    printf("You entered negative number: %d", number);
}
else if (number > 0)      //checks if number is a positive number
{
    printf("You entered positive number: %d", number);
}
else
{
    printf("You entered zero");
}
```

References

Tan, H.H., and T.B. D'Orazio. *C Programming for Engineering & Computer Science*. USA: WCB McGraw-Hill. 1999. Print.

C if, if...else and Nested if...else Statement. N.p., n.d. Web. 09 Feb. 2017.

<<https://www.programiz.com/c-programming/c-if-else-statement>>.