
Cours 4

CONDITIONAL STATEMENTS

Dr.
Chénche

Conditiona l statement

S

Part 1

**OVERVIEW (LOGICAL
EXPRESSIONS)**

LOGICAL EXPRESSIONS

- All conditional statements in C make use of **logical expressions** that are **true/false** statements about data
- Simple logical expressions are of the form:
(data operator data)

Conditiona l statement



S Part 2

IF

STATEMENT
S

The **if** statement

- Sometimes we want to selectively execute a block of code (**here we use the if statement**)

The if statement

- The syntax of the if statement in C is :

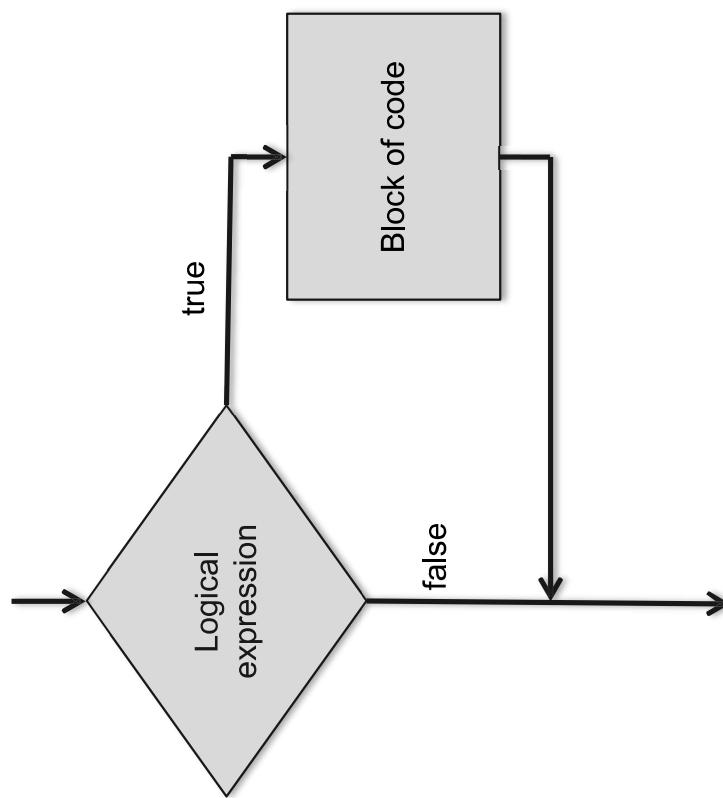
```
if ( logical expression )
```

```
{  
    // Block of code to execute if expression is true  
}
```

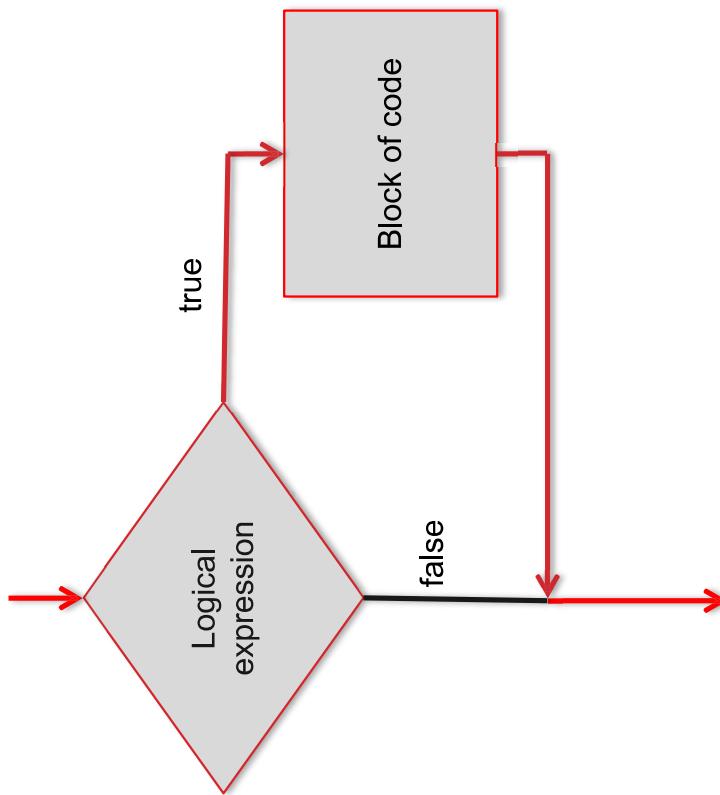
- When expression is true, the block of code is executed

- When expression is false, the block of code is skipped

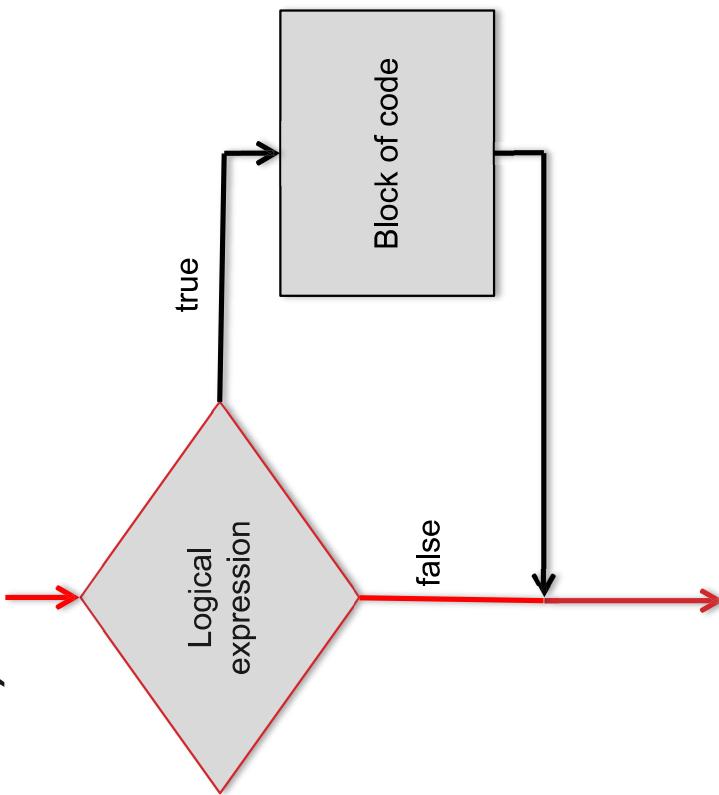
- We can visualize the program's if statement decision process, using a “flow chart” diagram



- If the logical expression is **true**, we take one path through the diagram (executing the block of code)



- If the logical expression is **false**, we take a different path through the diagram (skipping over the block of code)



```
int a, b;  
scanf ("%d %d", &a, &b);  
if (a == b)  
printf ("The two given numbers are equal \n");
```

Conditiona l statement

S

Part 3

if .. else statement

if..else statement

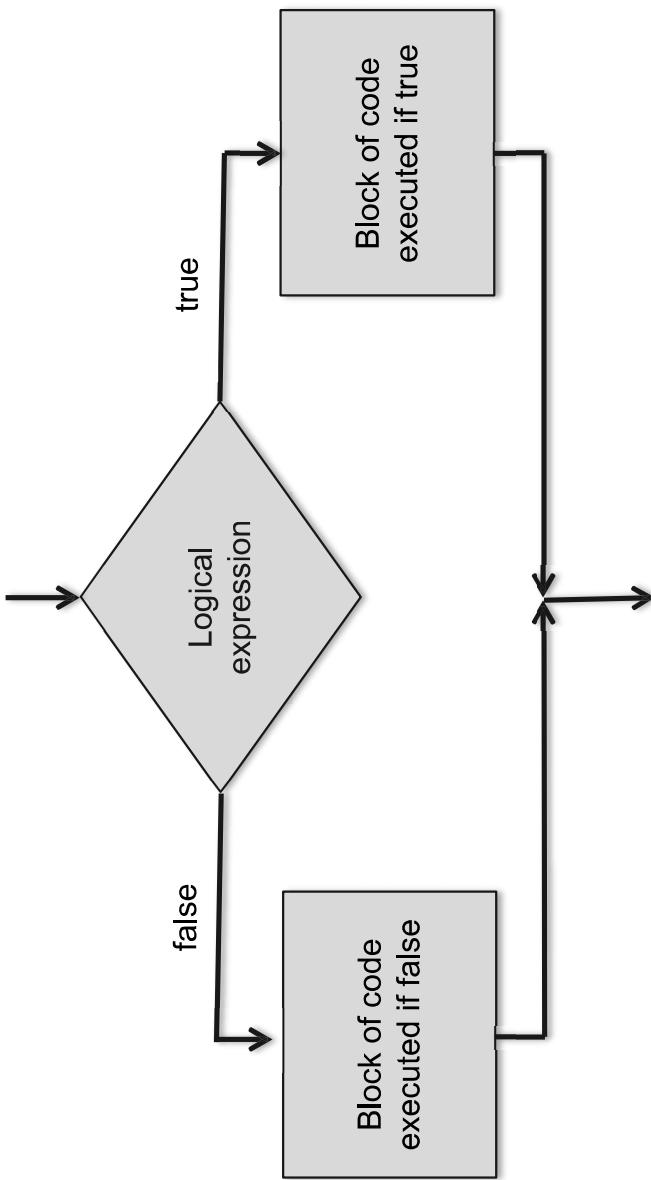
- Here a block of instructions is executed if a **condition is true**, and another block is executed if the **condition is false**.
- The syntax of the **if..else** statement is :

If..else statement

```
if( condition )
{
    /* bloc_d' instructions1 */
}
else
{
    /* bloc_d' instructions2 */
}
```

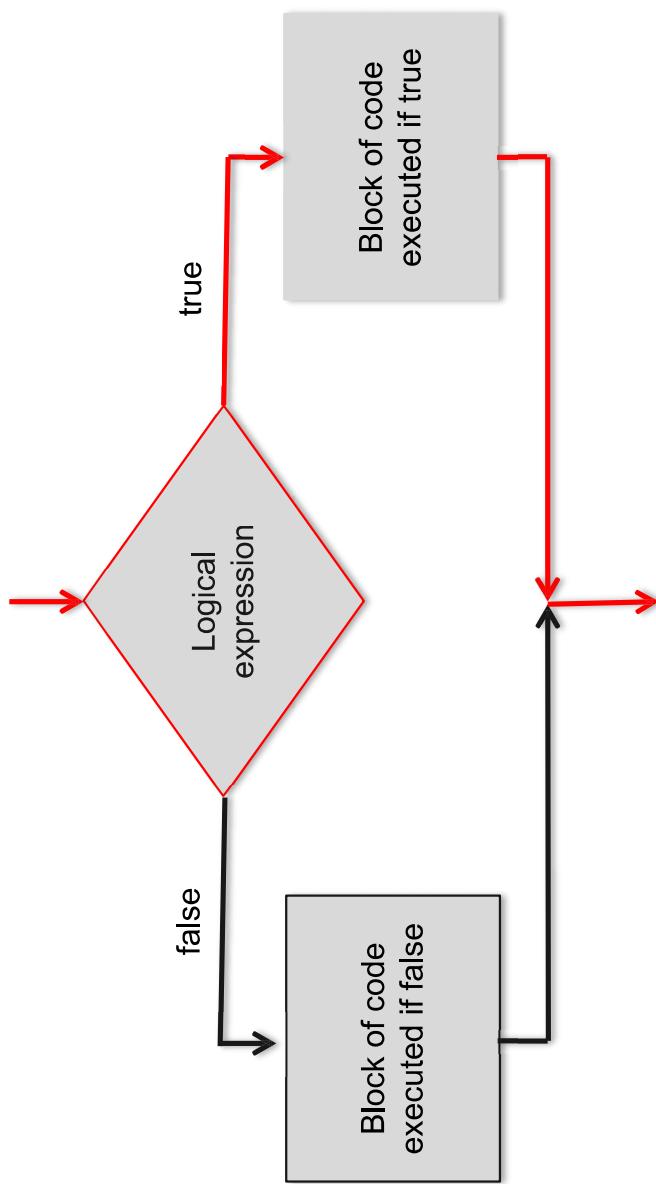
If..else statement

We can visualize the program's if-else statement decision process using a “flow chart” diagram



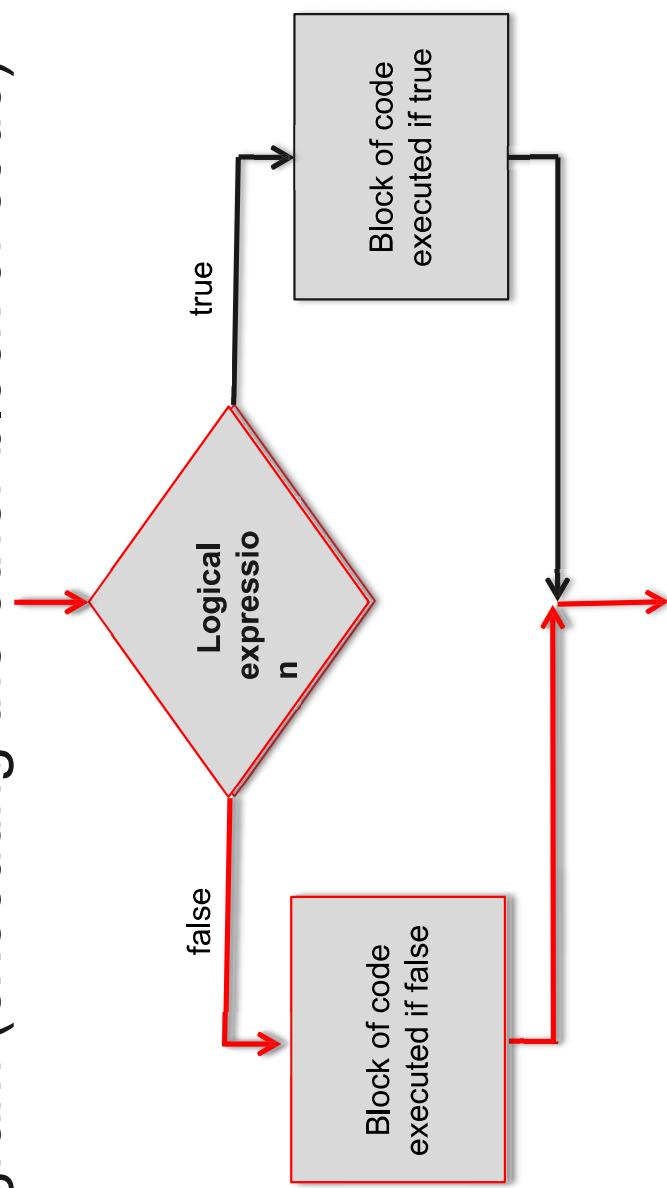
If..else statement

If the logical expression is true, we take one path through the diagram (executing one block of code)



If..else statement

If the logical expression is false, we take one path through the diagram (executing the other block of code)



Conditiona l statement



S Part 4

Nested if..else

Nested if..else

We can have **two or more if** statements inside each other to check multiple conditions :

- These are called **nested if..else statements**
- You need to take care when matching up { }

Nested if..else

```
if ( logical expression1 )
{
    if ( logical expression2 )
    {
        // Statements to execute if expressions1 and expression2 are true
    }
    else
    {
        // Statements to execute if expression1 true and expression2 false
    }
}
else
{
    // Statements to execute if expression1 false
}
```

Nested if..else :

```
if (age >= 17)
{
    if (gender == 'm')
        printf("You are an adult man \n");
    else
        printf("You are an adult woman.\n");
}
else
    printf("You are child \n");
```

Conditiona l statement

S

Part 5

Sequentiel if

Sequentiel if

- Sequential if means you have **several** if statements written one after another, each **if** is **checked separately** (they are *not connected with else*).
- That means **more than one condition can be true** and **multiple blocks can run**.

Sequentiel if :

```
if (number > 0)
    printf("The number is positive \n");
if (number % 2 == 0)
    printf("The number is even \n");
if (number % 5 == 0)
    printf("The number is divisible by 5 \n");
```

Exercise

Write the program that determines the maximum between two integers using the two forms of the if...else statement.

Exercise 02 :

Write a program that reads the temperature of the water then displays the state of the water (steam, liquid, or ice).

Exercise 03 :

Write a program that calculates the square root of a given integer, as well as its absolute value, then indicates whether it is even or odd