CONDITIONAL OPERATOR:

What is conditional operator ?

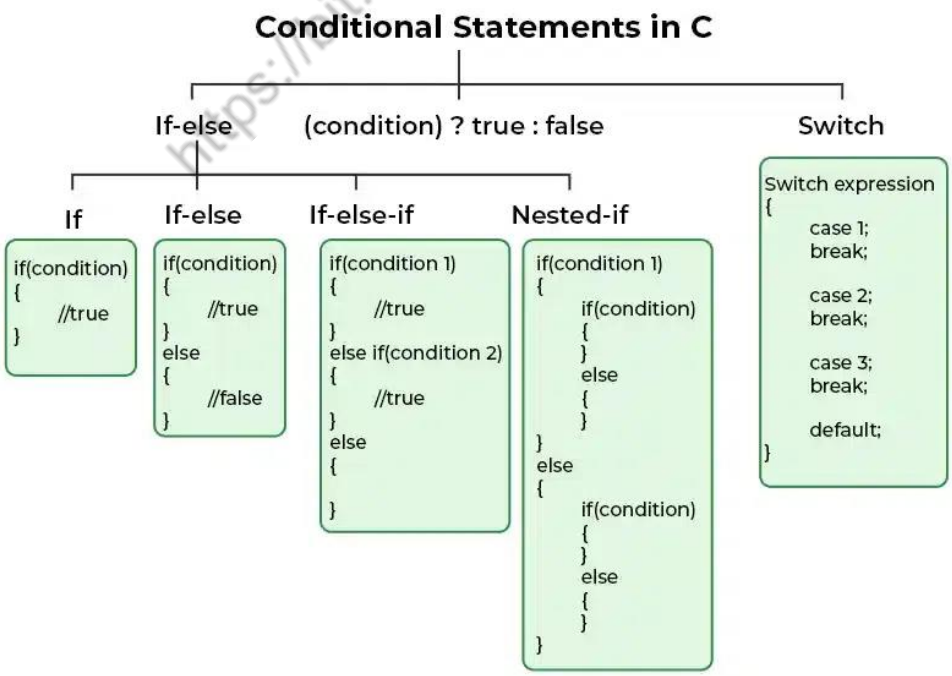
The **conditional statements** (also known as decision control structures) such as if, if else, switch, etc. are used for decision-making purposes in programs.

They are also known as Decision-Making Statements and are used to evaluate one or more conditions and make the decision whether to execute a set of statements or not.

Importance of conditional operator:

In real life, we often make decisions that determine our next actions. Similarly, in programming, decisions dictate the next block of code to execute. For example, if condition x occurs, execute y; otherwise, execute z. Multiple conditions can also be handled using else-if statements: if x occurs, execute p; else if y occurs, execute q; otherwise, execute r.

Types of conditional operator:



1. IF statement:

The if statement is the most simple decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e. if a certain condition is true then a block of statements is executed otherwise not.

Syntax of if condition:

if(*condition*)

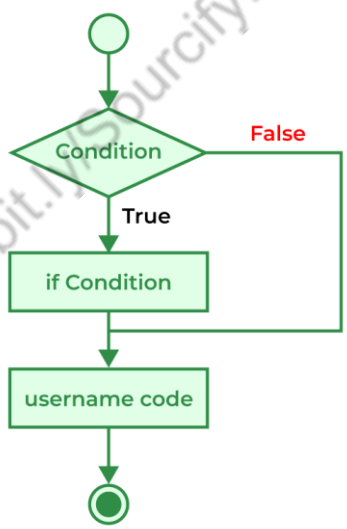
{

// Statements to execute if

// condition is true

}

Flowchart of IF:



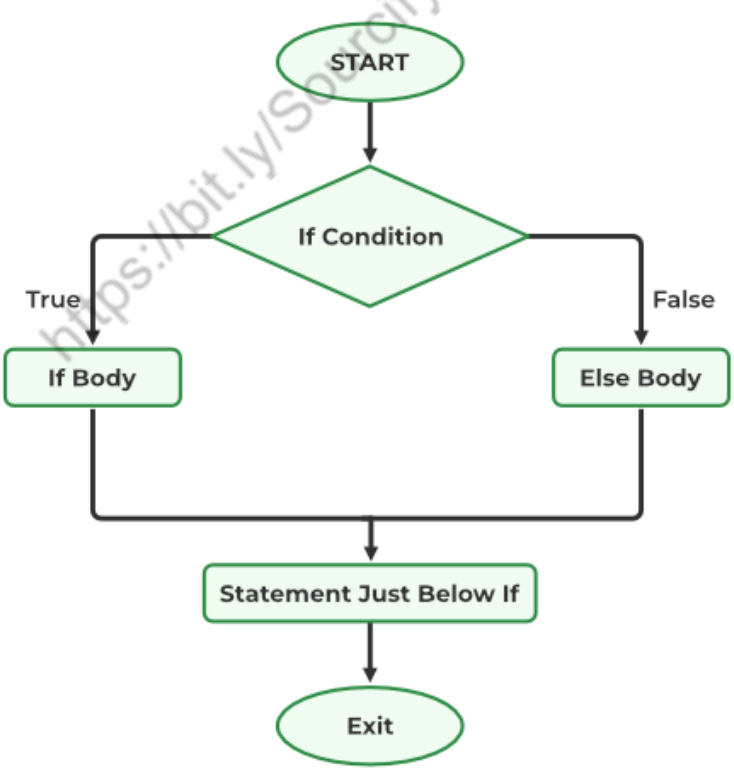
2. IF-Else statement:

if statement executes a block of code if a condition is true. If the condition is false, it does nothing. To handle false conditions, we use the else statement. The if-else statement has two blocks: one for when the condition is true and another for when it is false.

Syntax of IF-ELSE:

```
if (condition)
{
   // Executes this block if
   // condition is true
}
else
{
   // Executes this block if
   // condition is false
}
```

Flowchart :



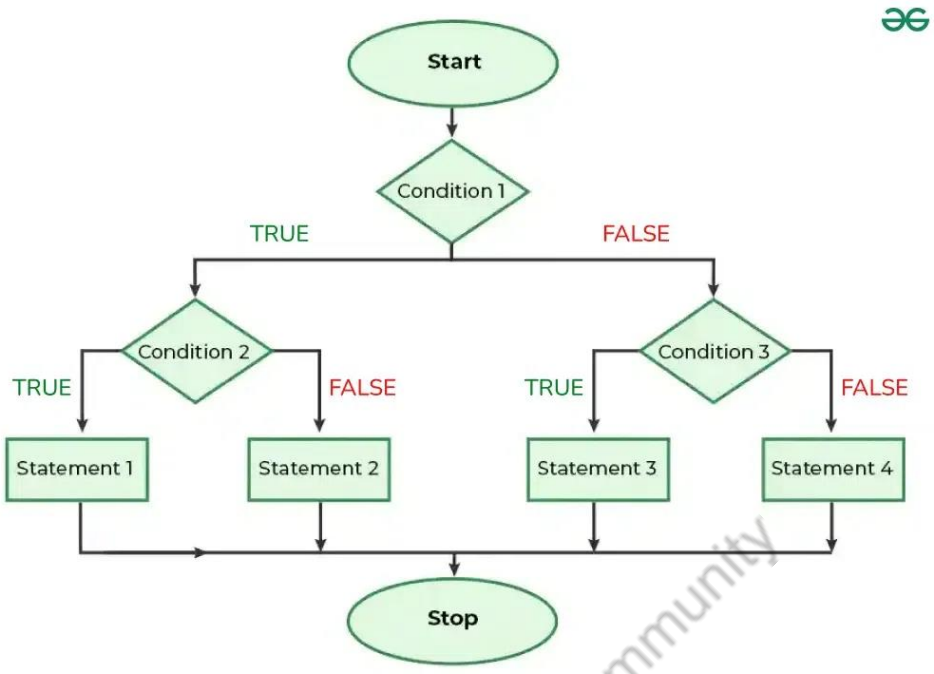
3. NESTED IF-ELSE statement:

A nested if is an if statement that is the target of another if statement. Nested if statements mean an if statement inside another if statement. It allow us to nested if statements within if statements, i.e., we can place an if statement inside another if statement.

Syntax:

```
if (condition1)
{
  // Executes when condition1 is true
  if (condition_2)
  {
    // statement 1
  }
  else
  {
      // Statement 2
   }
}
else {
    if (condition_3)
  {
    // statement 3
  }
  else
  {
      // Statement 4
   }
}
```

flowchart of Nested If-else statement :



4. switch Statement :

The switch- case statement is an alternative to the if else if ladder that can be used to execute the conditional code based on the value of the variable specified in the switch statement. The switch block consists of cases to be executed based on the value of the switch variable.

Syntax of switch-case statement :

switch (*expression*) {

case *value1:*

statements;

case *value2:*

statements;

....

....

....

default:

statements;

}

Flowchart: