

---

# Compound condition with AND

---

- A **compound** condition (or Boolean expression) is one where different relationships are combined using **logical operators**

The logical operators are:

- **or**
  - **and**
  - **not**
- It is rather **trivial** to handle **not** by **re-writing** a relationship...

Examples:

- "not (a < b)" is equivalent to "a >= b"
- "not (a == b)" is equivalent to "a != b"
- and so on...

- **Short circuit boolean evaluation**

- The "short circuit boolean evaluation" is a technique **commonly used** in programming languages (such as C, C++, Java) to evaluate **compound boolean expressions**
- The if-statement with an **or** logical operator:

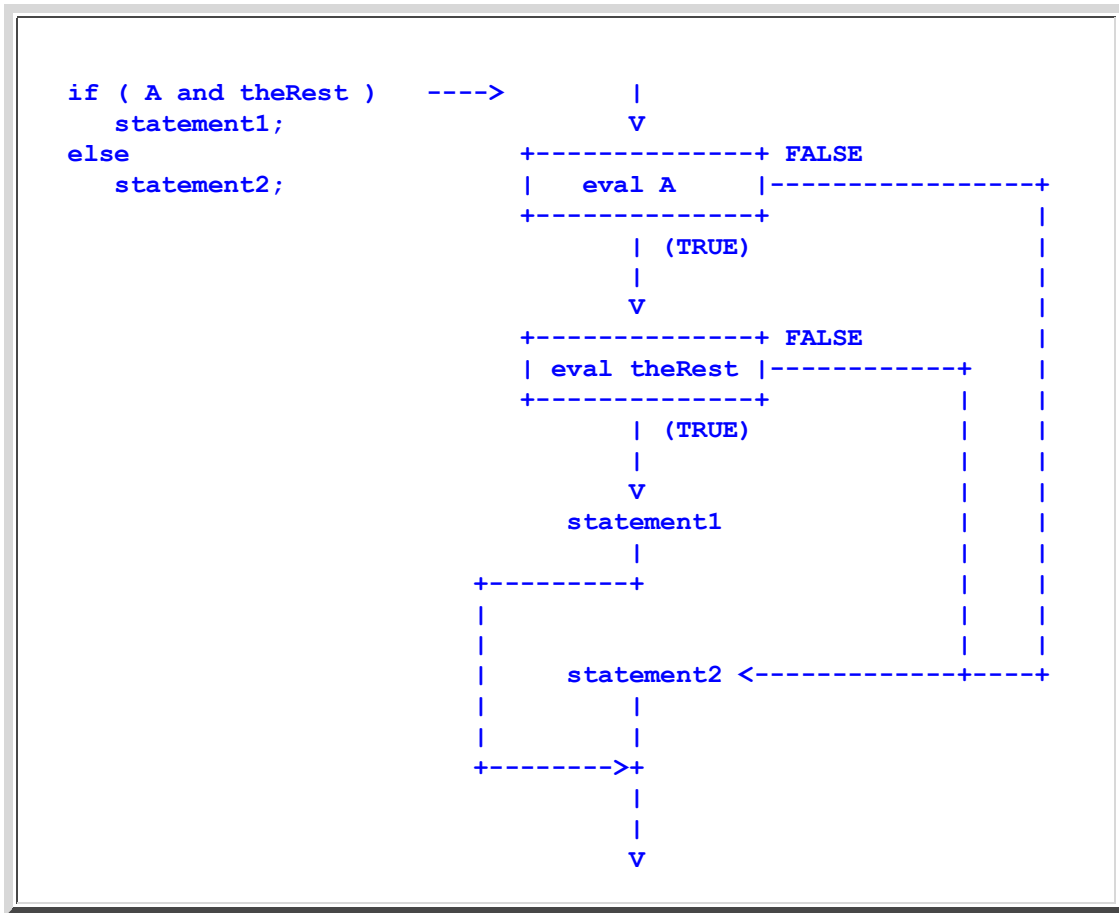
```
if ( A or "Rest of condition" )
    S1
else
    S2
```

Rest of the program

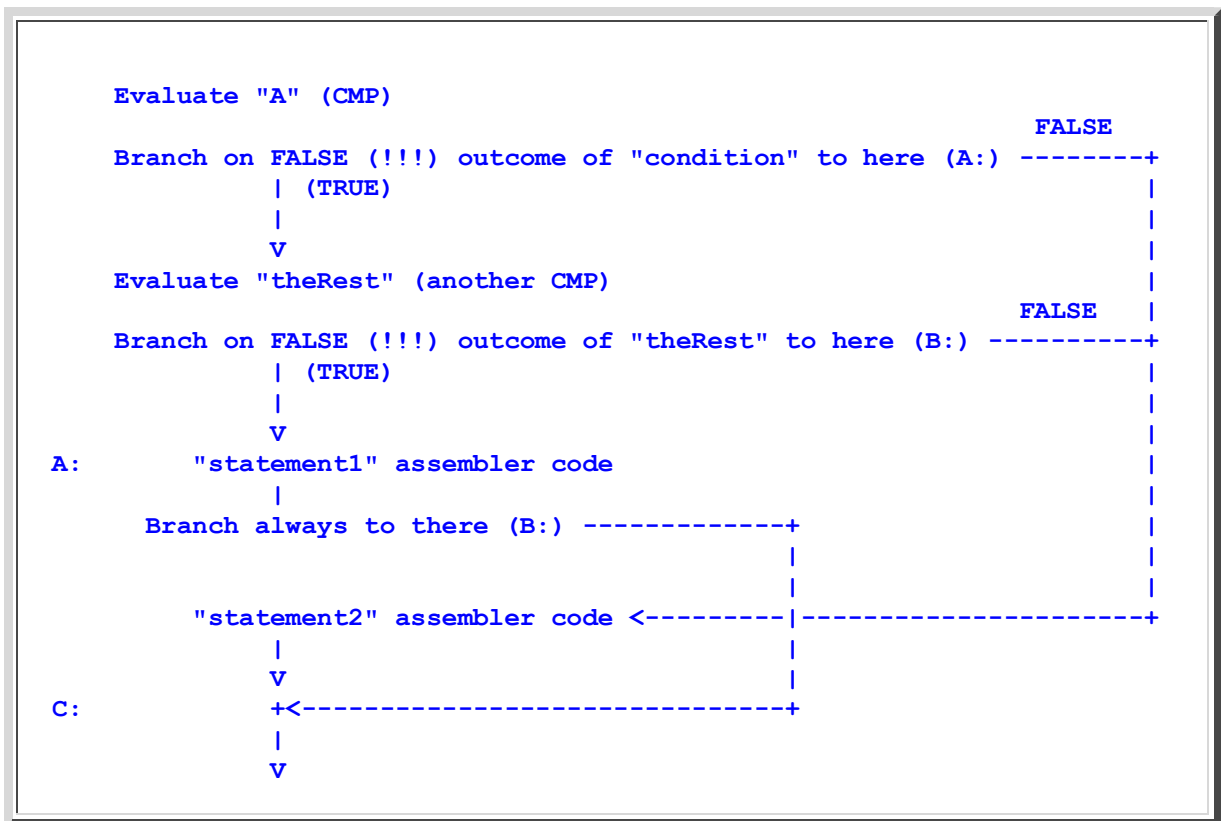
is execute as follows:

```
1. evaluate A
2. if TRUE then execute S1 and proceed to "Rest of the program"
3. if FALSE then CONTINUE with the evaluation of
   the "Rest of condition"
   3a. if "Rest of condition" is TRUE,
       then execute S1 and proceed to "Rest of the program"
   3b. if "Rest of condition" is FALSE,
       then execute S2 and proceed to "Rest of the program"
```

- **Program control** of the if-statement with an compound condition "A and theRest":



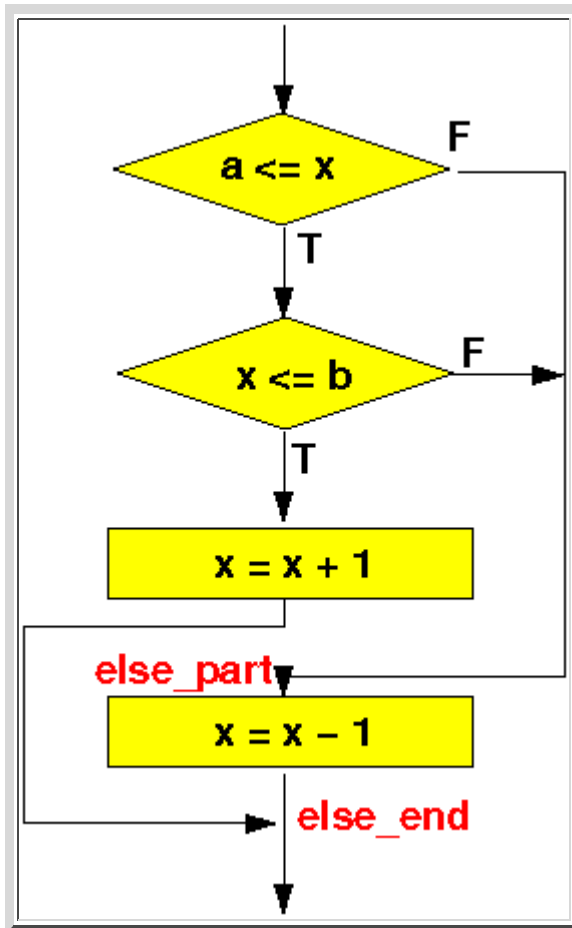
- Assembler construct that realizes the control flow of the `if`-statement with an compound condition "A or theRest" is as follows:



- **Example:**

```
int x, y, a;  
  
if (a <= x and x <= b)  
    x = x + 1;  
else  
    x = x - 1;
```

The **flow chart** of the above program is:



Assembler program for this compound if-statement: [click here](#)

---