LECTURE 02

CONDITIONAL STATEMENT

Big-O Coding

Website: www.bigocoding.com

Giới thiệu

BT1 – SỐ LỚN NHẤT

- Bạn được cho đầu vào 2 số nguyên a và b, hãy tìm giá trị **lớn nhất** của hai số nguyên đó.
- Ví dụ:
  - Input: 2 3
  - Output: 3

BT1 – Giải thuật

1. Sử dụng lệnh nhập để nhận vào 2 biến a, b

```cpp
cin >> a >> b;
```

```java
a = sc.nextInt();
b = sc.nextInt();
```

```python
a,b = map(int, input().split())
```

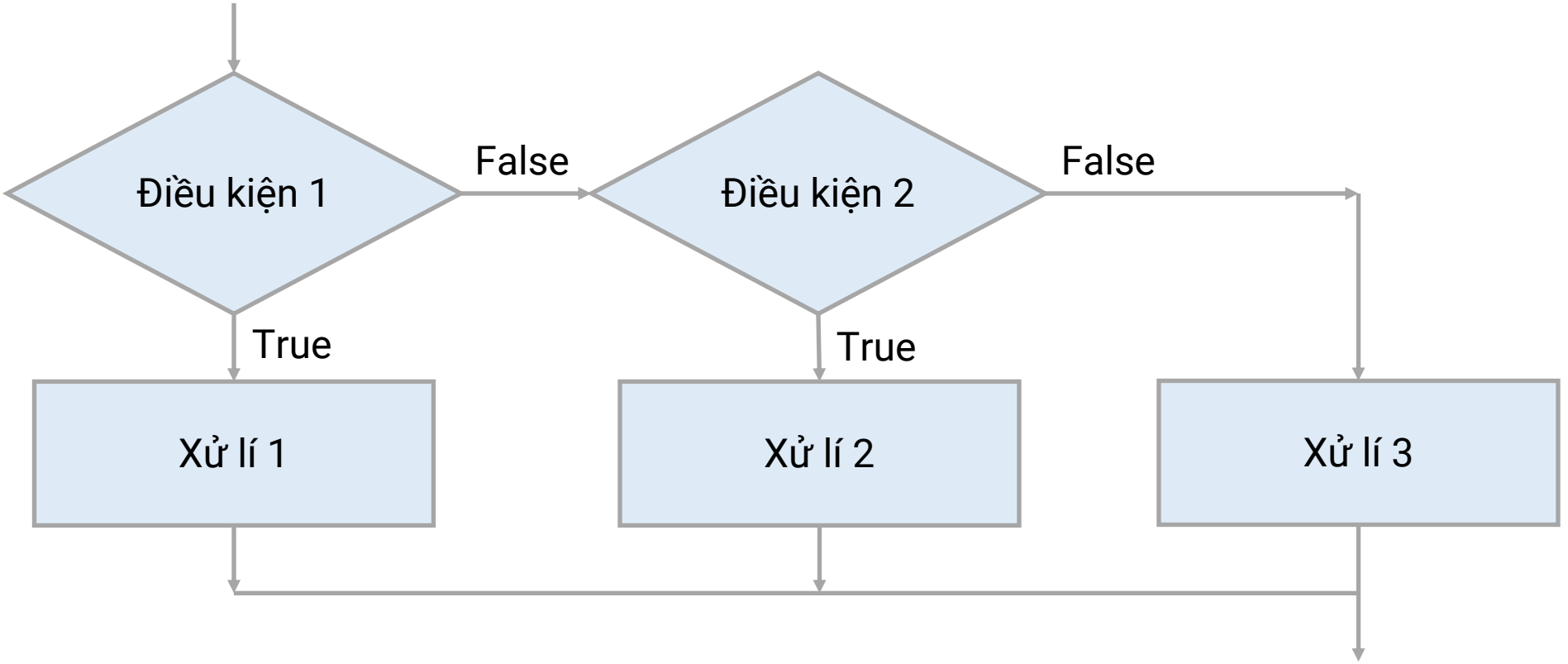
2. **Nếu** a lớn hơn b, in ra a

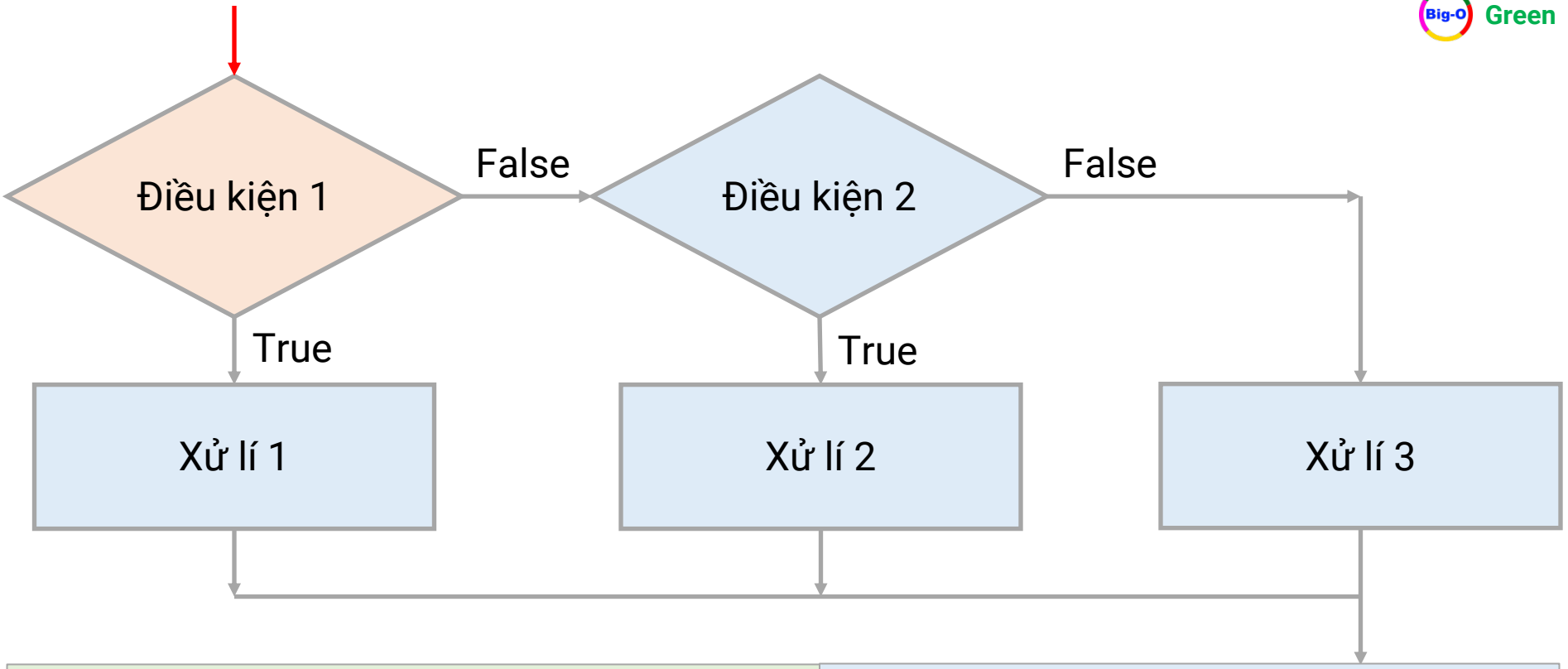
```cpp
cout << a;
```

```java
System.out.print(a);
```

```python
print(a)
```

3. **Ngược lại**, in ra b

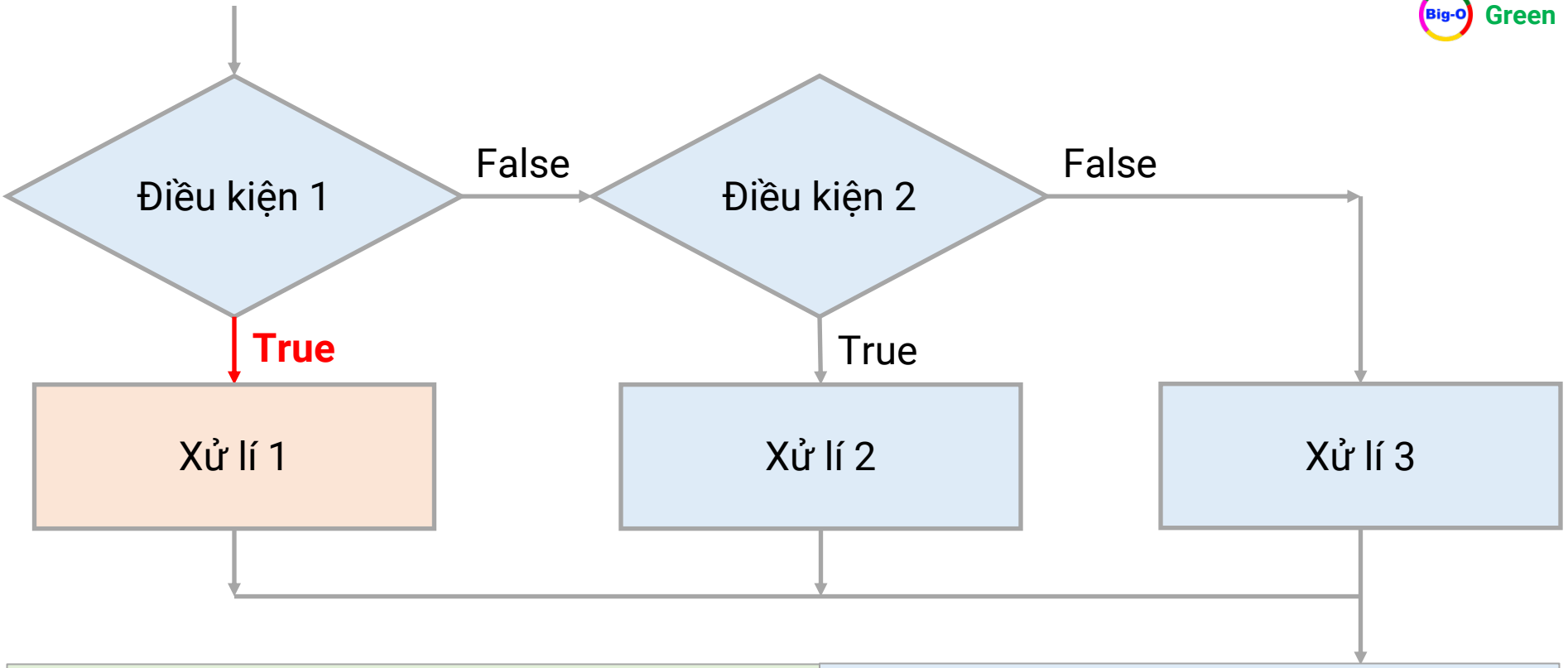

Điều kiện 1
False
Điều kiện 2
False
True
True
Xử lí 1
Xử lí 2
Xử lí 3




```java
if(điều kiện 1){

}
```

```python
if (điều kiện 1):
```

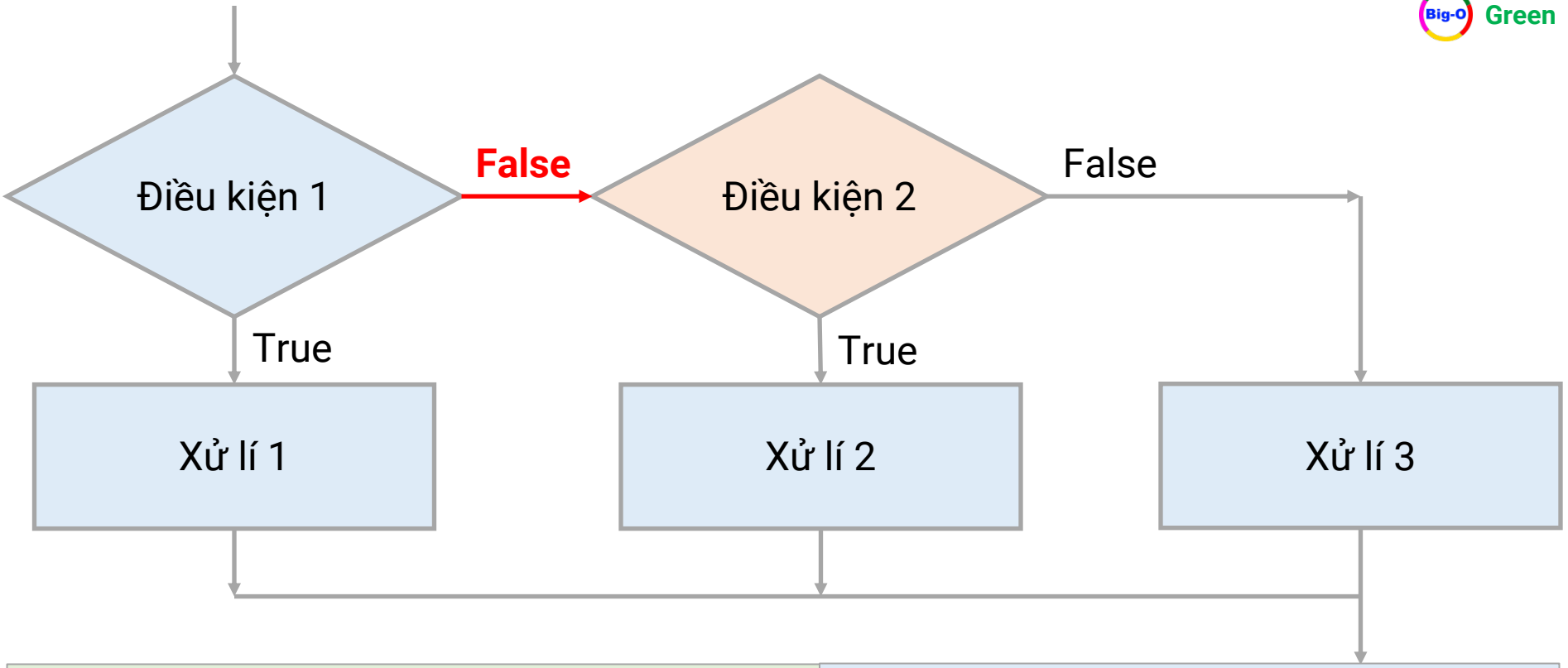



```
if(điều kiện 1){
    Xử lí 1;
}
```

- Cặp dấu { }
- Không có ; ở dòng if
- Không bắt buộc viết lùi "Xử lí 1" vào 1 tab

```
if (điều kiện 1):
  Xử lí 1
```

- Không dùng cặp dấu {}
- Phải có dấu :
- Bắt buộc viết lùi "Xử lí 1" vào 1 tab




```
if(điều kiện 1){
    Xử lí 1;
}
else if(điều kiện 2){

}
```

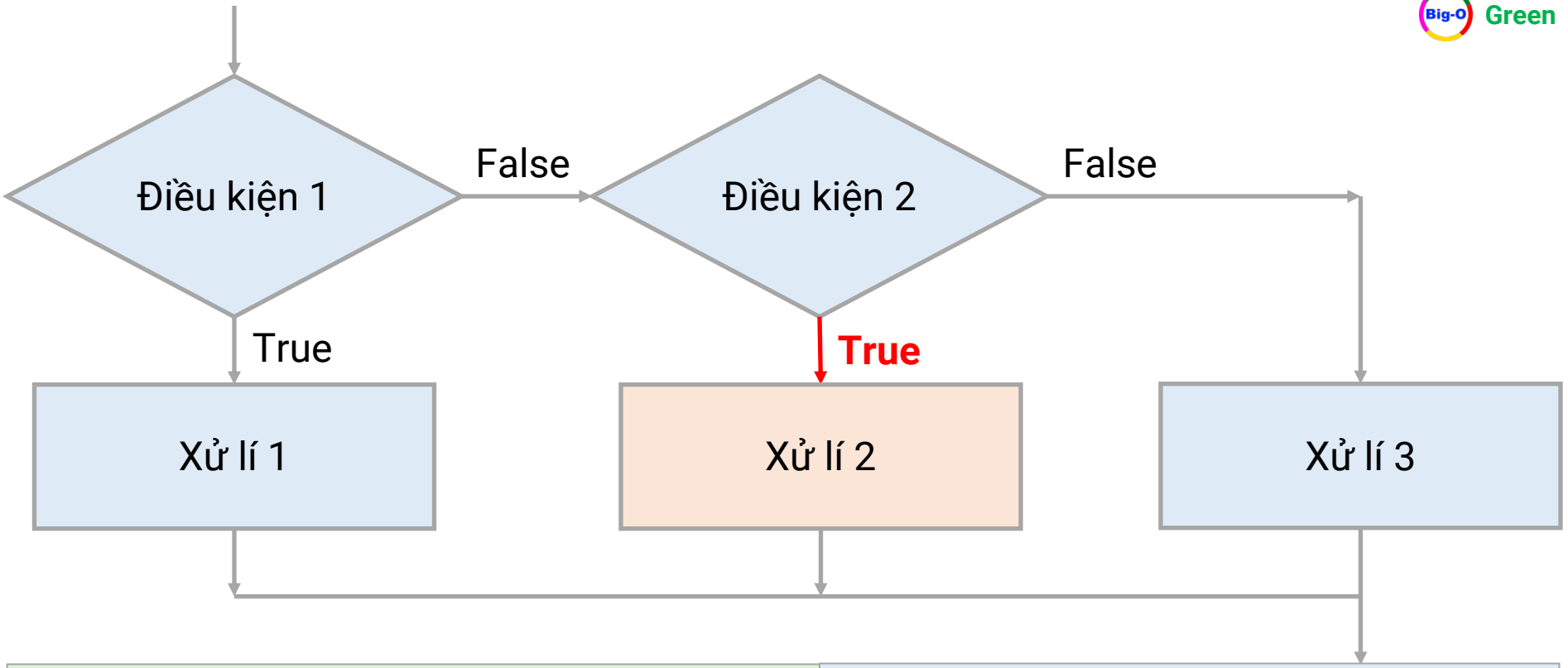
Java

C++

```
if (điều kiện 1):
  Xử lí 1

elif (điều kiện 2):
```

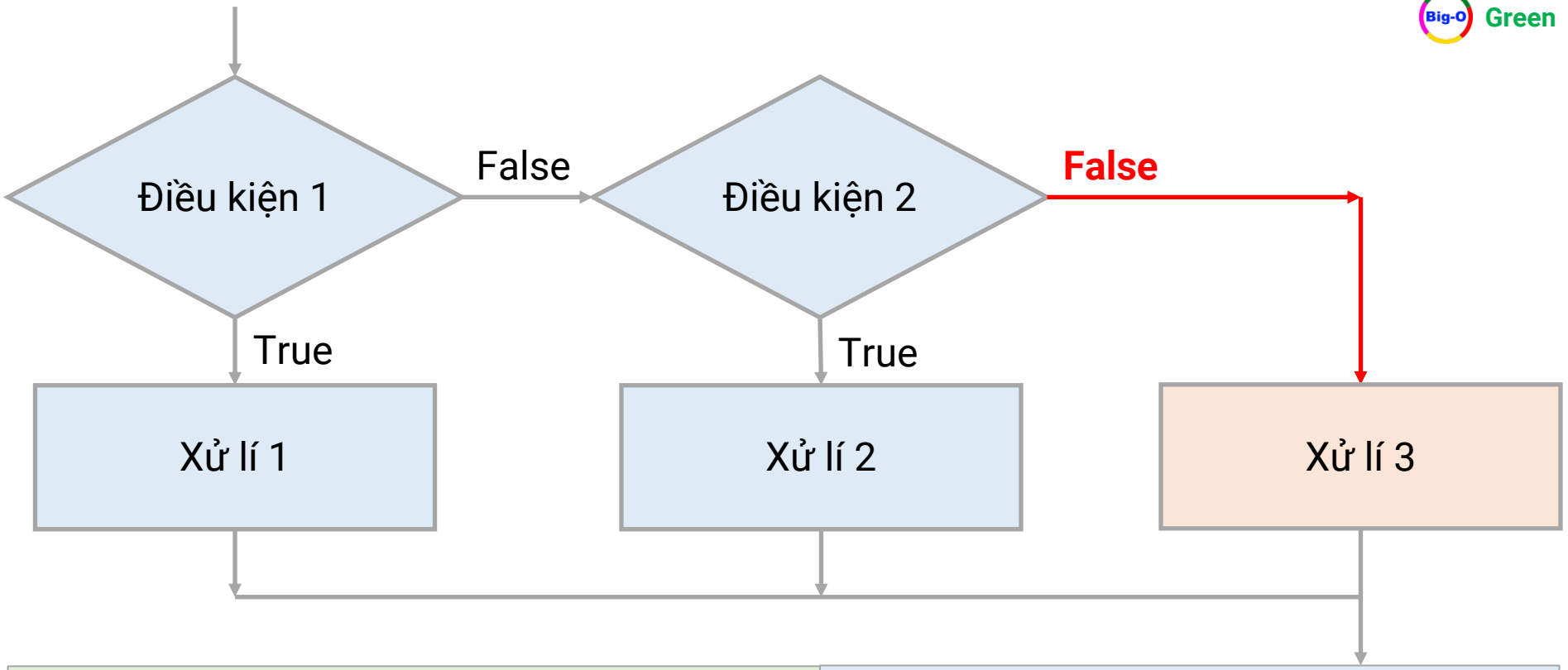
python




```java
if(điều kiện 1){
    Xử lí 1;
}
else if(điều kiện 2){
    Xử lí 2;
}
```

```python
if (điều kiện 1):
  Xử lí 1

elif (điều kiện 2):
  Xử lí 2
```

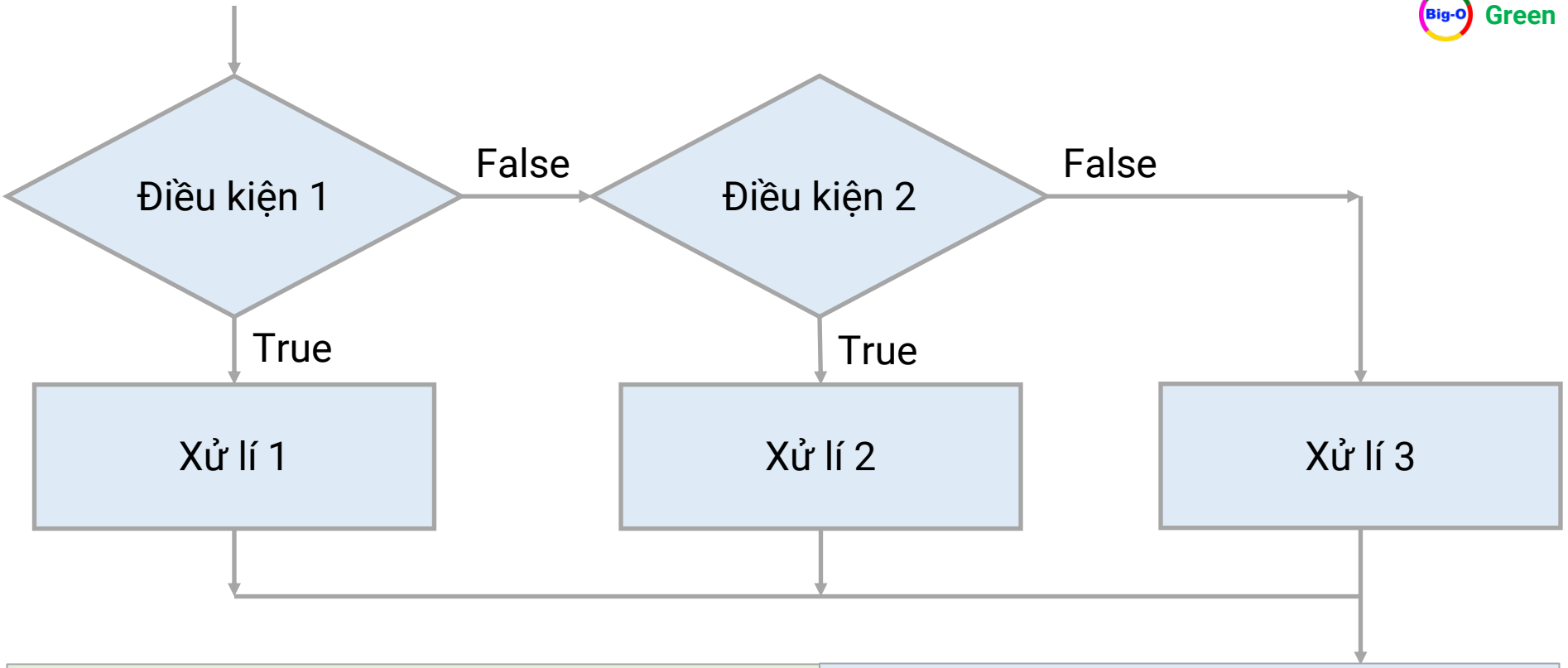



```
if(điều kiện 1){
    Xử lí 1;
}
else if(điều kiện 2){
    Xử lí 2;
}
else{
    Xử lí 3;
}
```

```
if (điều kiện 1):
  Xử lí 1

elif (điều kiện 2):
  Xử lí 2

else:
  Xử lí 3
```




```
if(điều kiện 1){
    Xử lí 1;
}
else if(điều kiện 2){
    Xử lí 2;
}
else{
    Xử lí 3;
}
```

```
if (điều kiện 1):
  Xử lí 1

elif (điều kiện 2):
  Xử lí 2

else:
  Xử lí 3
```

Ví dụ

```cpp
if(a > b){
    cout << a << endl;
}
else{
    cout << b << endl;
}
```


```java
if(a > b) {
    System.out.println(a);
}
else{
    System.out.println(b);
}
```


```python
if(a > b):
    print(a)
else:
    print(b)
```

Biểu thức điều kiện (boolean expression)

- Các biểu thức điều kiện được tạo thành từ các phép toán so sánh.
- Các phép toán so sánh sẽ trả về 1 trong 2 giá trị:
  - C++: true/false ➔ kiểu bool
  - Java: true/false ➔ kiểu boolean
  - Python: True/False ➔ kiểu bool
- a < b, a <= b, a > b, a >= b
- **a == b: so sánh bằng nhau.**
- **a != b: so sánh khác nhau.**

Ví dụ

```cpp
int a = 10;
int b = 2;
cout << (a == b);
cout << (a != b);
cout << (a < b);
cout << (a <= b);
cout << (a > b);
cout << (a >= b);
cout << (a % 2 == 0);
```


```python
a = 10
b = 2
print(a == b)
print(a != b)
print(a < b)
print(a <= b)
print(a > b)
print(a >= b)
print(a % 2 == 0)
```


```java
int a = 10;
int b = 2;
System.out.println(a == b);
System.out.println(a != b);
System.out.println(a < b);
System.out.println(a <= b);
System.out.println(a > b);
System.out.println(a >= b);
System.out.println(a % 2 == 0);
```

Một số toán tử khác

- Toán tử "và": trả ra True nếu cả a và b True. Ngược lại (nếu a False hoặc b False), trả ra False.
  - C++, Java: a **&&** b
  - Python: a **and** b

truth table

and	true	false
true	**true**	false
false	false	false

- Toán tử "hoặc": trả ra True nếu a hoặc b True. Ngược lại, (nếu a False và b False) trả ra False.
  - C++, Java: a || b
  - Python: a **or** b

truth table

or	true	false
true	true	true
false	true	**false**

- Toán tử "phủ định": trả ra True nếu a False. Ngược lại, trả về False (nếu a True).
  - C++, Java: **!**a
  - Python: **not** a

Ví dụ

```cpp
int a = 10;
int b = 2;
cout << ((a > 0) && (b < 0));
cout << ((a % 2 == 0) && (b % 2 == 0));
cout << ((a > 0) || (b < 0));
cout << (!(a % 2 == 0) && (b % 2 == 0));
```


```java
int a = 10;
int b = 2;
System.out.println((a > 0) && (b < 0));
System.out.println((a % 2 == 0) && (b % 2 == 0));
System.out.println((a > 0) || (b < 0));
System.out.println(!(a % 2 == 0) && (b % 2 == 0));
```


```python
a = 10
b = 2
print((a > 0) and (b < 0))
print((a % 2 == 0) and (b % 2 == 0))
print((a > 0) or (b < 0))
print(not(a % 2 == 0) and (b % 2 == 0))
```

Bài tập

BT1 – SỐ LỚN NHẤT

- Bạn được cho đầu vào 2 số nguyên a và b, hãy tìm giá trị **lớn nhất** của hai số nguyên đó.
- Ví dụ:
  - Input: 2 3
  - Output: 3

BT2 – XÉT DẤU

- Viết chương trình nhập vào hai số thực a và b khác 0. Kiểm tra xem chúng có cùng dấu hay không.
- Hai số thực được gọi là "**cùng dấu**" nếu như chúng **cùng dương hoặc cùng âm**.

BT3 – TÍNH QUÝ TRONG NĂM

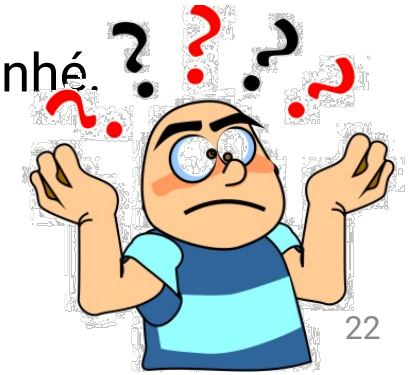
- Hôm nay cô giáo dạy Laura về các quý trong năm, nhưng cô vẫn không thực sự hiểu rõ về nó. Cô phải hoàn thành bài tập trước khi ngày mai tới lớp.
- Hãy giúp Laura bằng cách viết một chương trình nhập vào một tháng và trả về tháng đó thuộc quý nào trong năm.

BT4 – KIỂM TRA NĂM NHUẬN

- Beta cảm thấy rất rối với các khái niệm “năm nhuận”, hãy giúp anh ấy xác định một năm có phải năm nhuận hay không nhé!

BT5 – SỐ MAY MẮN

- Upan và Ipan là đôi bạn thân. Hai bạn muốn tìm cho mình chung một con số may mắn. Nhưng với Upan thì một số là số may mắn phải chia hết cho a. Còn với Ipan một số được gọi là may mắn nếu nó chia hết cho b.
- Biết hai bạn đang cùng nhau tìm một con số may mắn cho riêng mình. Ông bụt hiện ra và trao tặng hai bạn một con số x. Số x có thể thỏa mãn cho Ipan hoặc Upan. Hoặc thật tuyệt vời nó khớp với yêu cầu số may mắn của cả hai. Nhưng số x cũng có thể không phải là số may mắn của cả 2 bạn.
- Bạn hãy đánh giá giúp hai bạn số x mà bụt đã trao tặng nhé.



Lưu ý

Lưu ý 1: ko có dấu ; phía sau if, else if và else

- C++, Java: cẩn thận lỗi này.
- Python: ko có dấu ; ➔ ko bị lỗi này.

```cpp
if(a > b);
{
    cout << a;
}
```

C++

sẽ trở thành

```cpp
if(a > b)
{

}
cout << a;
```

C++

a = 3
b = 5

```java
if(a > b);
{
    System.out.print(a);
}
```

Java

sẽ trở thành

```java
if(a > b)
{

}
System.out.print(a);
```

Java

Lưu ý 2: ko viết điều kiện sau else

- Lỗi cú pháp.

```
if(a > b) {
    cout << a;
}
else (b >= a) {
    cout << b;
}
```

```
if(a > b){
    System.out.print(a);
}
else (b >= a) {
    System.out.print(b);
}
```

```
if(a > b):
    print(a)

else (b >= a):
    print(b)
```

Lưu ý 3: cẩn thận với trường-hợp-ngược-lại

- Nên nhớ: ngược lại của học-giỏi là học-ko-giỏi, chứ ko phải là học-dở.

```cpp
if(a > 0){
    cout << "So duong";
}
else{
    cout << "So am";
}
```

C++

```python
if(a > 0):
    print("So duong")

else:
    print("So am")
```

python

```java
if(a > 0){
    System.out.print("So duong");
}
else{
    System.out.print("So am");
}
```

a = 0

Java

Lưu ý 4: coi chừng xét thiếu biên

- Nếu sinh viên từ 5đ trở lên là qua môn học.

```cpp
if(mark > 5){
    cout << "Passed";
}
else{
    cout << "Failed";
}
```

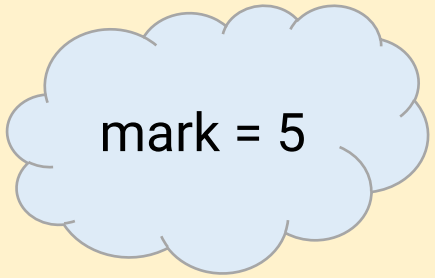
C++

```python
if(mark > 5):
    print("Passed")

else:
    print("Failed")
```

python

```java
if(mark > 5){
    System.out.print("Passed");
}
else{
    System.out.print("Failed");
}
```



Java

Lưu ý 5: so sánh bằng là 2 dấu bằng

- a sẽ được gán bằng 0.

```cpp
if(a = 0) {
    cout << "Invalid";
}
```


```python
if(a = 0):
    print("Invalid")
```


```java
if(a = 0) {
    System.out.print("Invalid");
}
```

Lưu ý 6: Các if có thể lồng nhau

```cpp
if(a > 0){
    if(b > 0){
        cout << "YES";
    }
}
```


```java
if(a > 0){
    if(b > 0){
        System.out.println("YES");
    }
}
```


```python
if(a > 0):
    if( b > 0):
        print("YES")
```

Lưu ý 7: Có thể bỏ dấu { }

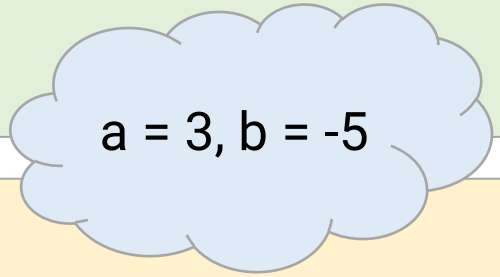
- C++, Java: nếu trong if chỉ có 1 lệnh xử lí, ta có thể bỏ cặp dấu { và }.

```cpp
if(a > b)
    cout << a << endl;
else
    cout << b << endl;
```


```java
if(a > b)
    System.out.println(a);
else
    System.out.println(b);
```

Lưu ý 8: else được tính cho if gần nhất

```cpp
if(a > 0)
    if(b > 0)
        cout << "a > 0, b > 0";
else
    cout << "a > 0, b <= 0";
```




```java
if(a > 0)
    if(b > 0)
        System.out.println("a > 0, b > 0");
else
    System.out.println("a > 0, b <= 0");
```

- Lỗi này dễ bị ở C++, Java.
- Python: không lo.
- Tốt nhất là lúc nào cũng phải có { và } dù chỉ có 1 lệnh xử lí.

Lưu ý 9: kết hợp if...else if...else...

- Nên sử dụng lệnh else if / elif thay vì 2 if rời nhau.
  - Giảm số lượng tính toán.
  - Đôi khi gây ra lỗi sai.
- Một câu lệnh if nên có 1 lệnh else tương ứng.

Ví dụ trong C++

```
if (a > 0 && b > 0){
    cout << "YES";
}
if(a < 0 && b < 0){
    cout << "YES";
}
if(a > 0 && b < 0){
    cout << "NO";
}
if(a < 0 && b > 0){
    cout << "NO";
}
```

SHOULD NOT

```
if (a > 0 && b > 0){
    cout << "YES";
}
if(a < 0 && b < 0){
    cout << "YES";
}
if(a > 0 && b < 0){
    cout << "NO";
}
else{
    cout << "NO";
}
```

WRONG!!!

a = 3, b = 5
YES, NO

```
if (a > 0 && b > 0){
    cout << "YES";
}
else if(a < 0 && b < 0){
    cout << "YES";
}
else{
    cout << "NO";
}
```

SHOULD

a = 3, b = 5
YES

Ví dụ trong Java

```
if(a > 0 && b > 0){
    System.out.print("YES");
}
if(a < 0 && b < 0){
    System.out.print("YES");
}
if(a > 0 && b < 0){
    System.out.print("NO");
}
if(a < 0 && b > 0){
    System.out.print("NO");
}
```

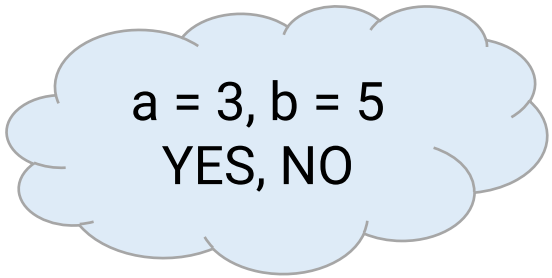
SHOULD NOT

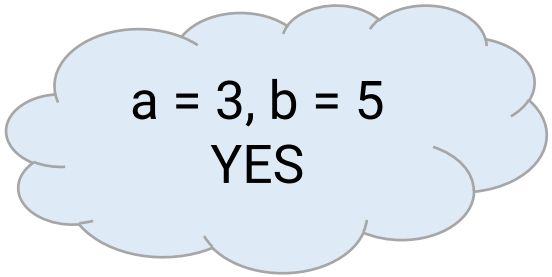
```
if(a > 0 && b > 0){
    System.out.print("YES");
}
if(a < 0 && b < 0){
    System.out.print("YES");
}
if(a > 0 && b < 0){
    System.out.print("NO");
}
else{
    System.out.print("NO");
}
```

WRONG!!!

```
if(a > 0 && b > 0){
    System.out.print("YES");
}
else if(a < 0 && b < 0){
    System.out.print("YES");
}
else{
    System.out.print("NO");
}
```

SHOULD





Ví dụ trong Python

```
if (a > 0 and b > 0):
    print("YES")

if(a < 0 and b < 0):
    print("YES")

if(a > 0 and b < 0):
    print("NO")

if(a < 0 and b > 0):
    print("NO")
```

SHOULD NOT

```
if (a > 0 and b > 0):
    print("YES")

if(a < 0 and b < 0):
    print("YES")

if(a > 0 and b < 0):
    print("NO")

else:
    print("NO")
```

WRONG!!!

a = 3, b = 5
YES, NO

```
if (a > 0 and b > 0):
    print("YES")

elif(a < 0 and b < 0):
    print("YES")

else:
    print("NO")
```

SHOULD

a = 3, b = 5
YES

Lưu ý 10: nếu ko quen and, hãy dùng 2 if lồng nhau

```cpp
if(a > 0 && b > 0){
    cout << "YES";
}
```

viết lại thành

```cpp
if(a > 0){
    if(b > 0){
        cout << "YES";
    }
}
```

```java
if(a > 0 && b > 0){
    System.out.print("YES");
}
```

viết lại thành

```java
if(a > 0){
    if(b > 0){
        System.out.print("YES");
    }
}
```

```python
if(a > 0 and b > 0):
    print("YES")
```

viết lại thành

```python
if(a > 0):
    if(b > 0):
        print("YES")
```

Lưu ý 11: nếu ko quen or, hãy dùng if rời nhau

```
if(mark < 5 || copy == true){
    cout << "Failed";
}
```

viết lại thành

```
if(mark < 5){
    cout << "Failed";
}
else if(copy == true){
    cout << "Failed";
}
```

```
if(mark < 5 || copy == true){
    System.out.print("Failed");
}
```

viết lại thành

```
if(mark < 5){
    System.out.print("Failed");
}
else if(copy == true){
    System.out.print("Failed");
}
```

```
if(mark < 5 or copy == True):
    print("Failed")
```

viết lại thành

```
if(mark < 5):
    print("Failed")
elif(copy == True):
    print("Failed")
```

Lưu ý 12: thứ tự so sánh (order of precedence)

- Toán tử not
- Toán tử * / %
- Toán tử + -
- Toán tử so sánh
- Toán tử and
- Toán tử or

```
x = 10
```

```
x + 1 > 2 || x + 1 < -3
```

```
10 + 1 > 2 || 10 + 1 < -3
```

```
11 > 2 || 11 < -3
```

```
true || false
```

```
true
```

Lưu ý 13: nên dùng dấu ()

```
time = 36
limit = 60
```

```
!time > limit
```

```
!36 > 60
```

```
false > 60
```

```
0 > 60
```

```
false
```

```
time = 36
limit = 60
```

```
!(time > limit)
```

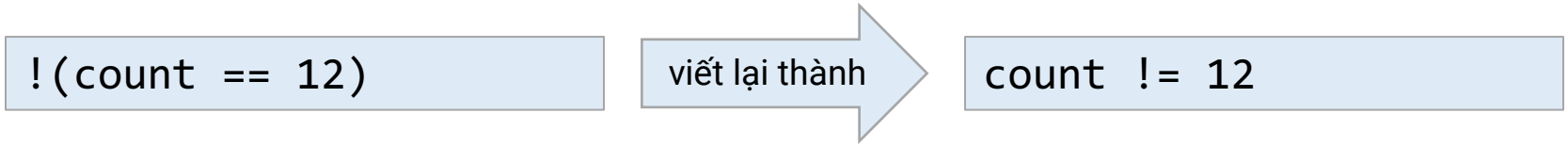
```
!(36 > 60)
```

```
!false
```

```
true
```

Lưu ý 14: Hạn chế dùng toán tử phủ định !

- Vì gây khó hiểu và thường làm developer hiểu sai.



Lưu ý 15: short-circuit evaluation

- Chương trình có thể rút ra kết quả true/false của 1 biểu thức so sánh mà không cần tính toán toàn bộ giá trị của biểu thức.

```
x = 10
```

```
x + 1 > 2 || x + 1 < -3
```

```
10 + 1 > 2 || x + 1 < -3
```

```
11 > 2 || x + 1 < -3
```

```
true || x + 1 < -3
```

```
true
```

Lưu ý 15: short-circuit evaluation

- Short-circuit evaluation giúp tránh các lỗi run-time error có thể xảy ra.

```cpp
int kids = 0;
int pieces = 100;
if((kids != 0) && (pieces / kids >= 2)){
    cout << ">= 2 pieces/kid";
}
```


```java
int kids = 0;
int pieces = 100;
if((kids != 0) && (pieces / kids >= 2)){
    System.out.print(">= 2 pieces/kid");
}
```


```python
kids = 0
pieces = 100
if((kids != 0) and (pieces / kids >= 2)):
    print(">= 2 pieces/kid")
```

Lưu ý 16: Giản lược điều kiện nhờ else if

```cpp
if(mark >= 9){
    cout << "Outstanding";
}
else if(mark < 9 && mark >=8){
    cout << "Excellent";
}
```

viết lại thành

```cpp
if(mark >= 9){
    cout << "Outstanding";
}
else if(mark >=8){
    cout << "Excellent";
}
```

```java
if(mark >= 9){
    System.out.print("Outstanding");
}
else if(mark < 9 && mark >=8){
    System.out.print("Excellent");
}
```

viết lại thành

```java
if(mark >= 9){
    System.out.print("Outstanding");
}
else if(mark >=8){
    System.out.print("Excellent");
}
```

```python
if(mark >= 9):
    print("Outstanding")
else if(mark < 9 && mark >=8):
    print("Excellent")
```

viết lại thành

```python
if(mark >= 9):
    print("Outstanding")
else if(mark >=8):
    print("Excellent")
```

Lưu ý 17: toán tử 3 ngôi (ternary operator) (C++)

```cpp
int a = 3;
int b = 5;
int c = 0;

c = a > b ? a : b;



cout << c << endl;
```

```cpp
int a = 3;
int b = 5;
int c = 0;

if(a > b)
    c = a;
else
    c = b;

cout << c << endl;
```

Lưu ý 17: toán tử 3 ngôi (ternary operator) (Java)

```java
int a = 3;
int b = 5;
int c = 0;

c = a > b ? a : b;



System.out.println(c);
```

```java
int a = 3;
int b = 5;
int c = 0;

if(a > b)
    c = a;
else
    c = b;

System.out.println(c);
```

Lưu ý 17: toán tử 3 ngôi (ternary operator) (Python)

```
a = 3
b = 5


c = a if a > b else b




print(c)
```

```
a = 3
b = 5
c = 0

if(a > b):
    c = a
else:
    c = b

print(c)
```

Lưu ý 18: Câu lệnh switch (C++, Java)

```
int x = 3;
switch(x)
{
    case 1:
        cout << "One" << endl;
        break;
    case 2:
        cout << "Two" << endl;
        break;
    case 3:
        cout << "Three" << endl;
        break;
    case 4:
        cout << "Four" << endl;
        break;
    default:
        cout << "Dont know" << endl;
        break;
}
```

```
Three
```

Python ko có switch...case

Không sử dụng switch cho 1 khoảng giá trị

```
int x = 1;
switch(x)
{
    case x >= 0 && x <= 9:
        cout << "1 digit" << endl;
        break;
    case 11, 12, 13, 14, 15:
        cout << "2 digits" << endl;
        break;
    default:
        cout << "Dont know" << endl;
        break;
}
```

Không sử dụng switch với số thực


```
double x = 3.14;
switch(x)
{
    case 2.71:
        cout << "E" << endl;
        break;
    case 3.14:
        cout << "PI" << endl;
        break;
    default:
        cout << "Dont know" << endl;
        break;
}
```

Cẩn thận khi thiếu lệnh break

```
int x = 3;
switch(x)
{
    case 1:
        cout << "One" << endl;
    case 2:
        cout << "Two" << endl;
    case 3:
        cout << "Three" << endl;
    case 4:
        cout << "Four" << endl;
        break;
    default:
        cout << "Dont know" << endl;
        break;
}
```

```
Three
Four
```

Lưu ý 19: Lợi dụng việc thiếu lệnh break

```
int x = 1;
switch(x)
{
    case 1:
    case 3:
        cout << "Odd" << endl;
        break;
    case 2:
    case 4:
        cout << "Even" << endl;
        break;
    default:
        cout << "Dont know" << endl;
        break;
}
```


```
Odd
```

Lưu ý 20: Cách viết gọn trong Python

```
print (1 < 2 < 3)
print ((1 < 2) and (2 < 3))

print(1 < 3 > 2)
print((1 < 3) and (3 > 2))
```

Lưu ý 21: Toán tử in

```
x = 97
if(x in [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]):
    print("1 digit")
elif (x in range(10, 100)):
    print("2 digits")
else:
    print("Dont know")
```


```
2 digits
```

Terminology

Conditional statement	Boolean expression
Order of precedence	Short-circuit evaluation
Ternary operator	

Hỏi đáp

Gợi ý

BT1 – Gợi ý

1. Sử dụng lệnh nhập, đọc vào 2 số nguyên a, b.
2. **Nếu** a > b,
   - Sử dụng lệnh xuất để in a.
3. **Ngược lại,**
   - Sử dụng lệnh xuất để in b.

BT2 – Gợi ý

1. Sử dụng lệnh nhập, để đọc vào 2 số thực a, b.
2. **Nếu** a > 0 và b > 0,
   - a, b cùng dấu.
3. **Ngược-lại-nếu** a < 0 và b < 0,
   - a, b cùng dấu.
4. **Ngược lại**,
   - a, b trái dấu.

BT3 – Gợi ý

1. Sử dụng lệnh nhập để nhập vào tháng.
2. **Nếu** tháng == 1, **hoặc** tháng == 2, **hoặc** tháng == 3,
   - Quý 1.
3. **Ngược-lại-nếu** tháng == 4, **hoặc** tháng == 5, **hoặc** tháng == 6,
   - Quý 2.
4. **Ngược-lại-nếu** tháng == 7, **hoặc** tháng == 8, **hoặc** tháng == 9,
   - Quý 3.
5. **Ngược lại,**
   - Quý 4.

BT4 – Gợi ý

1. Sử dụng lệnh nhập, để nhập vào năm.
2. **Nếu** năm chia hết cho 400, chắc chắn đó là năm nhuận. VD: năm 400, năm 800, năm 2000…
3. **Ngược-lại-nếu** năm chia hết cho 4 và năm đó cũng chia hết cho 100, đó không phải là năm nhuận. VD: năm 100, năm 300, năm 1000, năm 1900…
4. **Ngược-lại-nếu** năm đó chia hết cho 4, năm đó là năm nhuận.
5. **Ngược lại**, không phải là năm nhuận.

BT5 – Gợi ý

- Sử dụng operator %. Chia làm 4 trường hợp:
  1. **Nếu** x chia hết số của Upan và x chia hết số của Ipan.
  2. **Ngược-lại-nếu** x chỉ chia hết số của Upan.
  3. **Ngược-lại-nếu** x chỉ chia hết số của Ipan.
  4. **Ngược lại**.