

Programming 4kids

Logical Operators

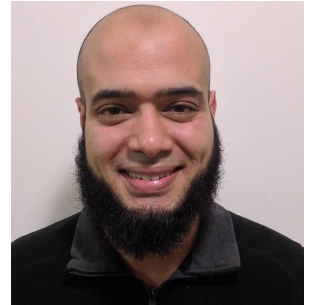
[Mostafa Saad Ibrahim](#)

Computer Vision Researcher @ Huawei Canada

PhD - Simon Fraser University

Bachelor / Msc - FCI Cairo University

Ex-(Software Engineer / Teaching Assistant)



True or False?

- Is 3 greater than 5? False
 - Is 3 less than 5? True
 - Is 3 equal to 5? False
 - Is 3 greater than or equal to 5? False
 - Is 3 greater than or equal to 3? True
 - Is 3 equal to 3? True
 - Is 3 greater than 1? True
 - Is 3 not equal to 4? True
 - Is 3 not equal to 3? False
 - Remember, we use bool for True and False conditions
- $3 > 5$
 - $3 < 5$
 - $3 == 5$
 - $3 >= 5$
 - $3 >= 3$
 - $3 == 3$
 - $3 > 1$
 - $3 != 4$
 - $3 != 3$

Let's code them

```
05_01.cpp
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     cout << (3 > 5) << "\n";
6     cout << (3 < 5) << "\n";
7     cout << (3 == 5) << "\n";
8     cout << (3 >= 5) << "\n";
9     cout << (3 >= 3) << "\n";
10    cout << (3 == 3) << "\n";
11    cout << (3 > 1) << "\n";
12    cout << (3 != 4) << "\n";
13    cout << ((3 != 3) << "\n";
14
15    return 0;
16 }
17
```

```
Console Problems Tasks
<terminated> ztemp [C/C++ Application] /ho
0
1
0
0
1
1
1
1
0
|
```

We can also use variables!

05_02.cpp

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int x, y;
6     x = 3, y = 5;
7
8     cout << (x > y) << "\n";
9     cout << (x < y) << "\n";
10    cout << (x == y) << "\n";
11    cout << (x >= y) << "\n";
12
13    return 0;
14 }
15
```

Console

Problems Tasks

<terminated> ztemp [C/C++ Application] /hor

```
0
1
0
0
|
```

We can use bool variables

```
05_2_B.cpp
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int x, y;
6     x = 3, y = 5;
7
8     bool result = (x > y);
9     cout << result << "\n";
10
11     result = (x < y);
12     cout << result << "\n";
13
14     cout << !result << "\n";
15     cout << !(x < y) << "\n";
16
17     return 0;
18 }
```

Console Problems Tasks Prj

<terminated> ztemp [C/C++ Application] /hor

0
1
0
0
|

Comparing strings

05_3.cpp

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     string name1 = "ali", name2 = "ali mostafa";
6     string name3 = "ziad", name4 = "ali", name5 = "ALI";
7
8     cout<<(name1 < name2) <<"\n";
9     cout<<(name1 > name3) <<"\n";
10    cout<<(name1 != name4) <<"\n";
11    cout<<(name1 == name4) <<"\n";
12
13    cout<<(name1 == name5) <<"\n";
14    cout<<(name1 > name5) <<"\n";
15
16    return 0;
17 }
```

Console

Problems Tasks Properties 1010 0101 Call Graph Search

<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/ecli

```
1
0
0
1
0
1
|
```

- Names are sorted in dictionary
- So comparison based on it
- Upper case comes first before lower case
 - Letter A smaller than a

AND Logical Operators

- Let say I am 30 years old, salary = 7000 and weight = 110 kg
- Mostafa > 25 years and salary < 8000? True
- Mostafa > 27 years and salary > 9000? False
- Mostafa > 35 years and salary < 8500? False
- Mostafa > 35 years and salary > 9000? False
- Summary
 - Only 1 case is True: when both conditions are true
 - In discrete mathematics: $p \wedge q$

OR Logical Operators

- Let say I am 30 years old, salary = 7000 and weight = 110 kg
- Mostafa > 25 years or salary < 8000? True
- Mostafa > 27 years or salary > 9000? True
- Mostafa > 35 years or salary < 8500? True
- Mostafa > 35 years or salary > 9000? False
- Summary
 - Only 1 case is False: when both conditions are false
 - In discrete mathematics: $p \vee q$

Logical Tables 2 variables

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	$\sim p$
T	F
F	T

Logical Tables 3 variables

AND truth table

INPUTS			OUTPUT
W	X	Y	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

OR truth table

INPUTS			OUTPUT
W	X	Y	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Logical Tables 3 variables

- Let say I am 30 years old, salary = 7000 and weight = 110 kg
- Mostafa > 25 years and salary < 8000 and weight < 150kg? True
 - T and T and T
- Mostafa > 25 years and salary < 8000 and weight > 70kg? True
 - T and T and T
- Mostafa > 25 years and salary < 8000 and weight > 200kg? False
 - T and T and F
- Mostafa > 35 years or salary < 8000 **or** weight > 200kg? True
 - F or T or F
- Mostafa > 35 years or salary > 9000 **or** weight > 200kg? False
 - F or F or F

Mixing Logical Operators

- Let say I am 30 years old, salary = 7000 and weight = 110 kg
- Mostafa > 35 years or salary > 6000 **or** weight > 200kg? True
- Mostafa > 35 years and salary > 6000 **or** weight > 200kg? False
 - Reduce every subgroup of ANDS first
 - **F and T** or F
 - F or F \Rightarrow F
- Mostafa > 20 years and salary > 10 and salary < 8000 **or** weight > 200kg?
 - **T and T and T** or F
 - T or F \Rightarrow T
- Precedence: Means what to apply first. Here AND before OR
 - What about $3 + 4 * 5$: is it $7 * 5$ or computed $3 + 20$? $*$ before $+$

C++ operators

- && for and
- || for or

```
05_4.cpp
1  #include<iostream>
2  using namespace std;
3
4  int main() {
5      int age = 30, salary = 7000;
6
7      bool result = (age > 25) && (salary < 8000);
8      cout<<result<<"\n";
9
10     cout<<( (age > 25) && (salary > 9000) )<<"\n";
11
12     cout<<( (age > 35) || (salary < 8500) )<<"\n";
13     cout<<( (age > 35) || (salary > 9000) )<<"\n";
14
15     return 0;
16 }
17
```

Console Problems Tasks Properties 1010 0101 Call Graph

<terminated> ztemp [C/C++ Application] /home/moustafa/workspace

```
1
0
1
0
|
```

Mixing Logical Operators in C++

```
3
4 int main() {
5     int age = 30, salary = 7000, weight = 110;
6
7     cout<<( (age > 25) && (salary < 8000) && (weight < 150) )<<"\\n";
8     cout<<( (age > 25) && (salary < 8000) && (weight > 200) )<<"\\n";
9
10    cout<<( (age > 35) || (salary > 6000) || (weight > 200) )<<"\\n";
11
12    cout<<( (age > 35) && (salary > 6000) || (weight > 200) )<<"\\n";
13    cout<<( (age > 20) && (salary > 6000) || (weight > 200) )<<"\\n";
14
15    return 0;
16 }
17
18
```

Console Problems Tasks Properties Call Graph Search

<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/eclipse_cpp/ztemp/C

```
1
0
1
0
1
|
```

So far

- All conditions are **ANDed**
 - True IFF all **ANDed** conditions are true
- All conditions are **ORed**
 - True if any condition is true. False if all are false
- Mixed ORs and ANDs
 - Find each subgroup of ANDs and evaluate first. Do normal ORing for the remaining
- Advanced: What if I want to **force** specific priority? Use ()
 - Every group of () is computed first. Find the simplest and reduce first
 - *Rarely used in practice*

() applied first

05_6.cpp

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int age = 30, salary = 7000, weight = 110;
6
7     // ANDs are evaluated
8     cout << ( age > 35 || salary > 6000 && weight > 200) << "\n";
9
10    // () are evaluated FIRST even before some ANDS
11    cout << ((age > 35 || salary > 6000) && weight > 200) << "\n";
12
13    return 0;
14 }
```


Let's try simplifying

- Let's simplify this expression $T \ \&\& \ T \ \&\& \ (F \ || \ (T \ \&\& \ T)) \ || \ T$
- $T \ \&\& \ T \ \&\& \ (F \ || \ (T \ \&\& \ T)) \ || \ T \Rightarrow (T \ \&\& \ T)$ is the simplest (). Its value is T
- $T \ \&\& \ T \ \&\& \ (F \ || \ T) \ || \ T \Rightarrow (F \ || \ T)$ is the simplest (). Its value is T
- $T \ \&\& \ T \ \&\& \ T \ || \ T \Rightarrow$ No more (). Next is group ands
- $T \ \&\& \ T \ \&\& \ T \ || \ T \Rightarrow T \ \&\& \ T \ \&\& \ T$ is group of ands. Evaluate to T
- $T \ || \ T$. Now final expression is set of conditions ORed $\Rightarrow T$

Coding mistakes

- Writing `< =` NOT `<=` (extra spaces)
- Writing `&` NOT `&&`
- Writing `& &` NOT `&&` (extra space)
- Writing `|` not `||`
 - `&` and `|` are called bits operators (later topic)
- Writing `=` not `==`
 - `=` is assignment. `==` is for comparing
- Writing `! result` NOT `!result` (extra space)
- `cout<<x < 5<<"\n";`
 - Compiler get confused. Use `()` \Rightarrow `cout<<(x < 5)<<"\n";`
- Imbalanced expression: `(T || (T && F))` \Rightarrow `(T || (T && F))`

Summary

- Comparisons creates for us conditions (each is true or false)
 - < <= > >= == != [over numbers or characters or strings]
- And Table: True IFF all **AND**ed conditions are true
- Or Table: True if any **OR**ed condition is true. False if all are false
- We can mix ands and ors. **Priority rules:**
 - First: Conditions inside () are evaluated. Find the simplest and replace it first
 - Second: each groups of ands [&&] are evaluated first
 - Remaining is either one result or ORed conditions
- In practice we usually have 1 or 2 simple conditions
 - If you did not understand () rules, it is ok for now.

Homework 1: Guess the output

03_homework1.cpp

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5
6     int a = 10, b = 20, c = 30, d = 40;
7
8     cout << (a + b == c) << "\n";
9     cout << (a + b + c >= 2 * d) << "\n";
10
11     cout << (a > 5 || d < 30) << "\n";
12     cout << (a > 5 && d < 30) << "\n";
13     cout << (a <= b && b <= c) << "\n";
14
15     cout << (a > 5 && d < 30 || c - b == 10) << "\n";
16     cout << (a <= b && b <= c && c <= d) << "\n";
17
18     cout << (a > 5 && d < 30 || c > d || d % 2 == 0) << "\n";
19     cout << (a > 5 && d < 30 || c > d && d % 2 == 0) << "\n";
20
21     cout << ( a == 10 || b != 20 && c != 30 || d != 40) << "\n";
22     cout << ((a == 10 || b != 20) && c != 30 || d != 40) << "\n";
23
24     return 0;
25 }
```

Homework 2: Create logic!

- Write a program that reads 3 integers about the class room
 - Number of boys (nb), number of girls (ng), number of teachers (nt)
- Prepare and **print** a boolean variable for these cases:
- nb greater than 25
- ng less than or equal to 30
- $nb > 20$ and $nt > 2$ or $ng > 30$ and $nt > 4$
- Either $nb < 60$ or $ng < 70$
- Neither $nb \geq 60$ nor $ng \geq 70$
- nb is 10 more students than ng
- Difference between nb and ng is more than 10 or $nt > 5$
- Either **nb is 10 more students than ng** or **ng is 15 more students than nb**

Homework 3 (optional): Simplify expressions

- For each expression:
 - Write a line of code that evaluate it to see its final value
 - Simplify it step by step to finally be a T or F
- $T \ \&\& \ T \ \&\& \ F \ \&\& \ T$
- $T \ \&\& \ T \ \&\& \ F \ \&\& \ T \ || \ T \ \&\& \ T$
- $T \ \&\& \ T \ \&\& \ T \ \&\& \ T \ || \ T \ \&\& \ (T \ || \ F)$
- $T \ \&\& \ T \ \&\& \ T \ || \ T \ \&\& \ (F \ || \ (T \ \&\& \ (T \ \&\& \ T)))$
- $T \ \&\& \ T \ || \ T \ \&\& \ F \ \&\& \ T \ || \ T \ \&\& \ T \ \&\& \ F \ || \ (T \ \&\& \ (T \ || \ F))$
- $T \ \&\& \ T \ || \ T \ \&\& \ F \ \&\& \ T \ || \ (T \ \&\& \ T \ \&\& \ F \ || \ (T \ \&\& \ (T \ || \ F)))$
- $(T \ \&\& \ T \ || \ T \ \&\& \ F \ \&\& \ T \ || \ T) \ \&\& \ T \ \&\& \ F \ || \ (T \ \&\& \ (T \ || \ F))$
- $T \ \&\& \ T \ || \ T \ \&\& \ (F \ \&\& \ T \ || \ T \ \&\& \ T) \ \&\& \ F \ || \ (T \ \&\& \ (T \ || \ F))$

تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً

