

# C++ Programming

## Relational Operators

**Mostafa S. Ibrahim**

*Teaching, Training and Coaching since more than a decade!*

*Artificial Intelligence & Computer Vision Researcher*

*PhD from Simon Fraser University - Canada*

*Bachelor / Msc from Cairo University - Egypt*

*Ex-(Software Engineer / ICPC World Finalist)*

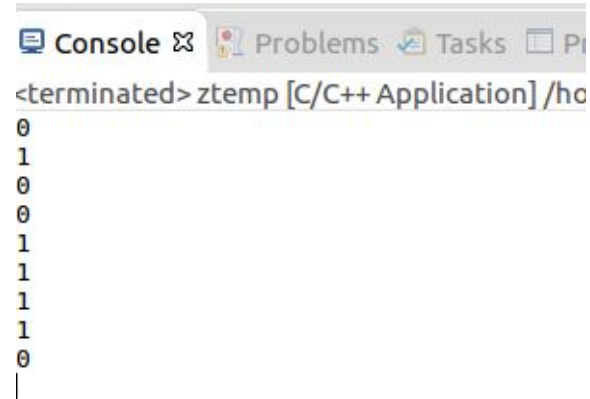


# True or False?

- 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

```
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
```



The screenshot shows a C++ IDE with a console window. The console output displays the results of the comparison operators used in the code: 0, 1, 0, 0, 1, 1, 1, 1, 0, and 1, each on a new line.

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

# We can also use variables!

```
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 Pre

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

# We can use bool variables

```
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 Pr

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

0  
1  
0  
0  
|

# Comparing strings

```
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

# Double comparisons

```
double a = 3.0 / 7.0;
double b = 1 + 3.0 / 7.0 - 1;

//0.428571 0.428571 0 SURPRISE SOMETIMES not 1
cout<<a<<" "<<b<<" "<<(a == b)<<"\n";

a = 5.0, b = 4.999999999999999999;

//5 5 1 SURPRISE SOMETIMES not 0
cout<<a<<" "<<b<<" "<<(a == b)<<"\n";

// Never compare doubles for equality directly
```

- C++ has an approximate representation for [real](#) values
- Never compare directly
- Later we will learn how to do it
  - Feel free to think about the possible semantic
  - Hint: Difference between 2 numbers

*“Acquire knowledge and impart it to the people.”*

*“Seek knowledge from the Cradle to the Grave.”*