

FAST

**National University of Computer and
Emerging Sciences Peshawar**

OOP Lab # 2.4

DEPARTMENT OF COMPUTER SCIENCE

C++ Programming

Computer Instructor: Muhammad Abdullah Orakzai



Prepared By: Muhammad Abdullah Orakzai (Lab Instructor CS)

Contents



- 1) C++ Booleans
- 2) Boolean Values
- 3) Boolean expressions



C++ Booleans

Very often, in programming, you will need a data type that can only have one of two values, like:

- YES / NO
- ON / OFF
- TRUE / FALSE

For this, C++ has a **bool** data type, which can take the values **true** (1) or **false** (0).



Boolean Values

A boolean variable is declared with the **bool** keyword and can only take the values **true** or **false**:

Example

```
bool isCodingFun = true;
```

```
bool isFishTasty = false;
```

```
cout << isCodingFun; // Outputs 1 (true)
```

```
cout << isFishTasty; // Outputs 0 (false)
```



Boolean Values...

```
#include<iostream>
using namespace std;
#include<cmath>

int main()
{
    bool isCodingFun=true;

    bool isFishTasty=false;

    cout<<isCodingFun<<endl;
    cout<<isFishTasty<<endl;
    return 0;
}
```

```
1
0
```



Boolean Values...

From the example above, you can read that a **true** value returns **1**, and **false** returns **0**. However, it is more common to return boolean values from boolean expressions (see next page).



Boolean Expressions

A **Boolean expression** is a C++ expression that returns a boolean value: **1** (true) or **0** (false).

You can use a comparison operator, such as the **greater than** (**>**) operator to find out if an expression (or a variable) is true:

Example

```
int x = 10;
```

```
int y = 9;
```

```
cout << (x > y); // returns 1 (true), because 10 is higher than 9
```



Boolean Expressions...

```
#include<iostream>
using namespace std;
#include<cmath>
```

```
int main()
{
    int n1=10;
    int n2=7;
    cout<<(n1>n2);
    return 0;
}
```

Output: 1



Boolean Expressions...

Or even easier:

Example

`cout << (10 > 9);` // returns 1 (true), because 10 is higher than 9

In the examples below, we use the **equal to** (`==`) operator to evaluate an expression:

Example

`int x = 10;`

`cout << (x == 10);` // returns 1 (true), because the value of x is equal to 10

Example

`cout << (10 == 15);` // returns 0 (false), because 10 is not equal to 15



References

- <https://beginnersbook.com/2017/08/cpp-data-types/>
- <https://www.geeksforgeeks.org/c-data-types/>
- http://www.cplusplus.com/doc/tutorial/basic_io/
- <https://www.geeksforgeeks.org/basic-input-output-c/>
- <https://www.w3schools.com/cpp/default.asp>
- <https://www.javatpoint.com/cpp-tutorial>

THANK YOU

