

Contents

- C++ Comparison Operators
- C++ Logical Operators
- String

C++ Comparison Operators

Operator	Name	Example
==	Equal to	<code>x==y</code>
!=	Not equal	<code>x !=y</code>
>	Greater than	<code>x>y</code>
<	Less than	<code>x<y</code>
>=	Greater than or equal to	<code>x>=y</code>
<=	Less than or equal to	<code>x<=y</code>

Ex-1: Comparison Operators

```
#include <iostream>
using namespace std;
int main(){
    int x = 25;
    int y= 23;
    cout << (x==y);
    return 0;
}
```

0

Ex-2: Comparison Operators

```
#include <iostream>
using namespace std;
int main(){
    int x = 15;
    int y= 13;
    cout << (x>=y);
    return 0;
}
```

1

Logical Operators

- Logical operators are used to determine the logic between variables or value.

Logical Operators

Operator	Name	Ex
&&	Logical AND	$x < 5 \ \&\& \ x < 10$
	Logical OR	$x < 5 \ \ x < 4$
!	Logical NOT	$!(x < 5 \ \&\& \ x < 10)$

Ex-3: Logical Operators (AND)

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

int main() {
    int x = 15;
    cout << (x > 13 && x < 20);
    return 0;
}
```

Ex-4: Logical Operators (OR)

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

int main() {
    int x = 15;
    cout << (x > 13 || x < 14);
    return 0;
}
```


Ex-5: Logical Operators (NOT)

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

int main() {
    int x = 15;
    cout << (!(x > 13 && x <
20)); // return 0;
}
```

0

C++ String

- Strings are used for storing text.
- A string variable contains a collection of characters surrounded by double quotes.
- To use strings, you must include an additional header file in the source code, the `<string>` library.

Ex-6: String

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

int main() {
    string greeting = "hi";
    cout <<greeting;
    return 0;
}
```



hi

Length of string

- This is just an alias of `length()`. It is completely up to you if you want to use `length()` or `size()`:

Ex:

```
|string txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
|cout << "The length of the txt string is: " << txt.length();
```

Ex-7: String

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

int main() {
    string myString = "hi";
    cout << myString[0];
    return 0;
}
```



hi

Ex-8: String

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

int main() {
    string myString = "hello";
    cout << myString[1];
    return 0;
}
```



e

Ex-9: String index number

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

int main() {
    string myString = "hi";
    myString[0] = 'P';
    cout << myString;
    return 0;
}
```

pi