Relational Operators

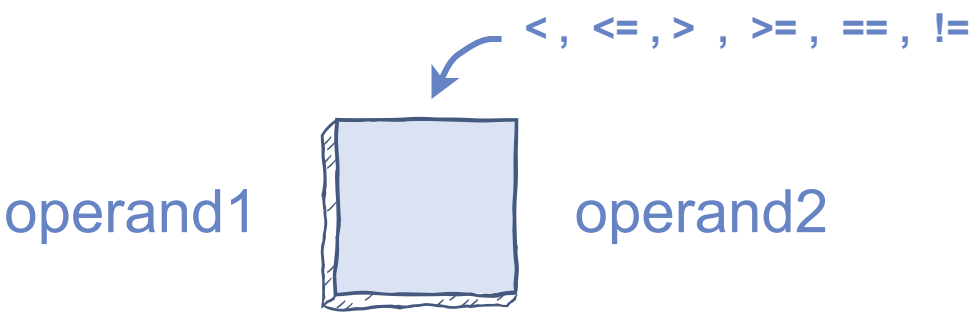
In this lesson, we will cover the relational operators from soup to nuts.

We'll cover the following

Introduction to relational operators

A relational operator compares the value of two operands.

> The output of a relational operator is a `bool` data type.



Here is the list of relational operators available in C++.

Operator	Operation	Use
>	Greater than	Returns 1 if operand1 is greater than operand2
>=	Greater than or equal to	Returns 1 if operand1 is greater than or equal to operand2
<	Less than	Returns 1 if operand1 is less than operand2
<=	Less than or equal to	Returns 1 if operand1 is less than or equal to operand2
==	Equal to	Returns 1 if operand1 is equal to operand2
!=	Not equal to	Returns 1 if operand1 is not equal to than operand2

Example program with `int`

Consider two operands of type `int`: The value of `operand1` is `50`, and the value of `operand2` is `26`. Let's apply each relational operator on them.

Run the code below and see the output!

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

int main() {

  int operand1 = 50;
  int operand2 = 26;
  cout << " operand1 = " << operand1 << " , operand2 = " << operand2 << endl;
  cout << " Is operand1 less than operand2? " << (operand1 < operand2) << endl;
  cout << " Is operand1 less than or equal to operand2? " << (operand1 <= operand2 )<< endl;
  cout << " Is operand1 greater than operand2? " << (operand1 > operand2) << endl;
  cout << " Is operand1 greater than or equal to operand2? " << (operand1 >= operand2) << endl;
  cout << " Is operand1 equal to operand2? " << (operand1 == operand2) << endl;
  cout << " Is operand1 not equal to operand2? " << (operand1 != operand2) << endl;

  return 0;
}
```

In the above code, you can see the output is `0` if the relation evaluates to `false` and the output is `1` if the relation evaluates to `true`.

> 📝 In C++, we can also compare the `float`, `string`, and `char` data types using relational operators.

> 📝 When we apply relational operators to the operands of type `char`, the compiler will compare the ASCII values of the character.

Example program with a `string`

Consider two operands of a string data type. Let's apply a relational operator to these operands and see the results.

Try running the code below!

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

int main() {

  string operand1 = "Microsoft";
  string operand2 = "Samsung";
  cout << " Is operand1 greater than operand2? " << (operand1 > operand2) << endl;

  return 0;
}
```

In the code above, the compiler continually compares the strings character by character until the ASCII value of characters in both strings is equal.

> 📝 If you try to write `<=`, `>`, `>=`, `==`, and `!=` with a space, a syntax error will occur.

Q

What is the output of the following code?

```
int main() {
  int operand1 = 67;
```

```
  int operand1 = 67;
  int operand2 = 70;

  cout << operand1 == operand2 << endl;
  cout << operand1 < operand2 << endl;
   return 0;
}
```

Retake Quiz

This sums up our discussion of relational operators. Let’s move on to the next lesson, where we will learn logical operators.