

## Increment and Decrement

---

- Other operators with side effects are the pre- and post-increment and decrement operators.
  - Increment: `++` `++x`, `x++`
    - `++x` is the same as : `(x = x + 1)`
      - Has value  $x_{old} + 1$
      - Has side-effect of incrementing `x`
    - `x++`
      - Has value  $x_{old}$
      - Has side-effect of incrementing `x`
  - Decrement `--` `--x`, `x--`
    - similar to `++`

## Relational Operators:-

- Relational operators allow you to compare variables.
  - They return a 1 value for true and a 0 for false.

Operator	Symbol	Example
----------	--------	---------

Equals	<code>==</code>	<code>x == y</code> NOT <code>x = y</code>
--------	-----------------	--

Greater than	<code>&gt;</code>	<code>x &gt; y</code>
--------------	-------------------	-----------------------

Less than	<code>&lt;</code>	<code>x &lt; y</code>
-----------	-------------------	-----------------------

Greater/equals	<code>&gt;=</code>	<code>x &gt;= y</code>
----------------	--------------------	------------------------

Less than/equals	<code>&lt;=</code>	<code>x &lt;= y</code>
------------------	--------------------	------------------------

Not equal	<code>!=</code>	<code>x != y</code>
-----------	-----------------	---------------------

- There is no `bool` type in C. Instead, C uses:
  - 0 as false
  - Non-zero integer as true



- $x \gg n$ 
  - Shifts the bits in  $x$   $n$  positions right.
    - shifts in the sign if it is a signed integer (arithmetic shift)
    - shifts in 0 if it is an unsigned integer
  - $x \gg 1$  is  $0111\ 1111\ 1111\ 1000_2$  (unsigned)
  - $x \gg 1$  is  $1111\ 1111\ 1111\ 1000_2$  (signed)

## Operating on Bits (3)

---

- Bitwise logical operations
  - Work on all integer types
    - & Bitwise AND
   
 $x = \quad 0xFFFF$ 
  
 $y = \quad 0x002F$ 
  
 $x \& y = \quad 0x0020$
    - | Bitwise Inclusive OR
   
 $x | y = \quad 0xFFFF$
    - ^ Bitwise Exclusive OR
   
 $x \wedge y = \quad 0xFFDF$
    - ~ The complement operator
   
 $\sim y = \quad 0xFFD0$ 
      - o Complements all of the bits of  $X$