

CS208

W, 7 Jan 2026

Bitwise operators

"AND"

0101 & 1101

$$\begin{array}{r} 0101 \\ \& 1101 \\ \hline 0101 \end{array}$$

$$\begin{array}{r} 0101 \\ 1101 \\ \hline 1101 \end{array}$$

OR

Bitwise ops cont'd

$$\begin{array}{r} 0101 \\ \wedge 1101 \\ \hline 1000 \end{array}$$

XOR
"exclusive OR" — this OR that
but not both

$$\sim 0101 \rightsquigarrow 1010$$

$$\sim 1101 \rightsquigarrow 0010$$

bitwise
NOT

Example

int x = 0xABC;

int y = 0x543;

printf("0x%x", x | y);

32
bits

OR
|

x

0000 0000 0000 0000 0000 1010 1011 1100

y

0000 0000 0000 0000 0101 0100 0011

x | y

0000 0000 0111 1111 1111 1111

printf("0x%x\n", x | y) → 0xfff

Masking w/ &

Given X , is the fourth bit from the right (ie. the 2^3 's place) == 1 or == 0?

$$X \& 0x00000008 == 0 \text{ or } / ?$$

X	-----	110110110110	11
mask	0	000000000000	1000

$$X \& \text{mask} = 0$$

count_one_bits

```
int mask = 0x00000001;
```

loop 32 times

if $n \& \text{mask}$ is 1

increase your count

Shift mask left by 1

mask starts	00 ————— 01	shift again
after shifting	0 ————— 010	
		0 — 0100

$\ll$ - shift left

$\text{mask} = \text{mask} \ll 1;$

shifts mask left by 1 bit.

$\gg$ - shifts right

(might be useful in print-integers
- in-binary)