BITWISE OPERATORS

BINARY

- Integers, characters, shorts, etc all have a binary representation underneath
- Type just determines how they are converted to binary
- Binary: 0s and 1s
- Like decimal, but instead of 100s, 10s, 1s place we have 4s, 2s, 1s place (aka powers of 2)

BITWISE OPERATORS

- Operations performed on individual bits of binary numbers (bits in same locations)
- Perform on all bits in binary numbers
- Be careful -- add extra parentheses
    - Bitwise operators typically have lower preference
    - Safer just to add parentheses

BITWISE OPERATORS (CONT.)

- & = AND:
  - 1 if both bits are 1
  - 0 otherwise
- | = OR:
  - 1 if either bit is 1
  - 0 otherwise
- ^ = XOR:
  - 1 if exactly one of the two bits is 1
  - 0 otherwise
- ~ = bitwise complement

BIT SHIFTING

- `<< n`
  - shift left by n bits
  - adds 0s at the end
- `>> n`
  - shift right by n bits
  - adds 0s at the beginning (for unsigned)
- Be careful -- only use unsigned if you plan to do bit shifting (behavior with right shift varies for signed values)

WHO CARES?

- Can come up in programming interviews
- Sometimes it's far easier to do something using bit manipulation
- Example:
  - Check if a number is power of 2

MINILAB 14