

bit - binary digit

- Most basic unit of information in a computer
- Has one of two values - 0 or 1
- Interpretation depends on context
 - On or off
 - True or false
 - Literal 1 or 0
 - One state or the other
 - Part of a more complex piece of information (number, string, etc.)

byte

- Smallest addressable unit of memory
- These days almost always 8 bits

- Can store

- a single extended ASCII character

- unsigned values from 0 to 255 ($2^8 - 1$)

- signed values from -2^7 to $2^7 - 1$ (-128 to 127)

- Multi-byte values

- int in C and C++ is typically 4 bytes

Integer representation

- Unsigned - just use the binary representation

- Signed

- Historically, there are several options for representing negative numbers

- Two's complement has become the standard

Two's complement representation

- Positive numbers are represented as themselves
- A negative number is the two's complement of its absolute value
- Calculating two's complement is done in 2 steps
 1. invert the bits (0's become 1's and vice versa. stopping here is one's complement)
 2. Add one, discard overflow

4-bit signed integers

5 \rightarrow 0101

-5 \rightarrow two's complement of 0101

flip bits: 1010

add one: 1011

- Addition is preserved

$$\begin{array}{r} -5 \\ + 6 \\ \hline 1 \end{array} \quad \begin{array}{r} 1011 \\ 0110 \\ \hline *0001 \end{array}$$

$$\begin{array}{r} -5 \\ + 5 \\ \hline 0 \end{array} \quad \begin{array}{r} 1011 \\ 0101 \\ \hline *0000 \end{array}$$

Carry bit is thrown away

- To subtract numbers, convert the second number to its two's complement and add

- Two's complement of 0 is 0

- Only 1 representation of 0

- Non-negative numbers start with 0, negative numbers start with 1

	0000
flip	1111
add	0000