UC Berkeley EECS
Sr Lecturer SOE
Dan Garcia

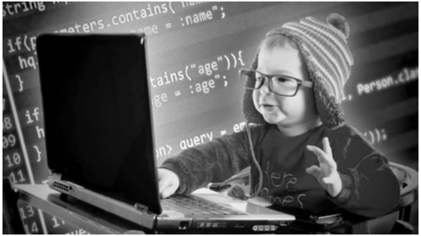
The Beauty and Joy of Computing

Lecture #3
Abstraction II

What I Wish I Knew When I Started My SW Career

Among the advice given:

- Don't be afraid to learn on job
- Never ask for permission unless it would be reckless not
- Exercise
- Long hours: sometimes ok, usually harmful
- Learn as much as you can. It's hard, and it takes work



lifehacker.com/what-i-wish-i-knew-when-i-started-my-career-as-a-softwa-1681002791

Abstraction (revisited): Numbers

- Number bases, including binary and decimal, are used for reasoning about digital data.
- Bits represent binary data using base two digits: zero and one.
- Hexadecimal, or base-16, is often used in reasoning about data e.g., colors in images.
- Different bases help in reasoning about digital data; digital data is stored in bits.

```
000000100000
000111011100
001000000100
011011011011
011000000011
001000100010
001010001010
000101110010
000010000100
000001111000
```

Base 10 #s, Decimals

Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Example:

$3271 = 3271_{10} =$

$(3 \times 10^3) + (2 \times 10^2) + (7 \times 10^1) + (1 \times 10^0)$

Base 2 #s, Binary (to Decimal)

Digits: 0, 1 (<u>bi</u>nary <u>di</u>gits → bits)

Example: "1101" in binary? ("0b1101")

$$1101_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

$$= 8 + 4 + 0 + 1$$

$$= 13$$

Base 16 #s, Hexadecimal (to Decimal)

Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
10,11,12,13,14,15

Example: "A5" in Hexadecimal?

$$0xA5 = A5_{16} = (10 \times 16^1) + (5 \times 16^0)$$

$$= 160 + 5$$

$$= 165$$

Decimal vs Hexadecimal vs Binary

- **N bits = 2^N things**
- **4 Bits**
  - 1 "Nibble"
  - 1 Hex Digit = 16 things
- **8 Bits**
  - 1 "Byte"
  - 2 Hex Digits = 256 things
  - Full color is often 256 Red, 256 Blue, 256 Green (#4A00FF)

D	H	B
00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

☑

(Cal) Smallest to Largest?

a) 0xC < 0b1010 < 11

b) 0xC < 11 < 0b1010

c) 11 < 0b1010 < 0xC

d) 0b1010 < 11 < 0xC

e) 0b1010 < 0xC < 11

Convert FROM decimal TO binary

- **E.g., 13 to binary?**
- **Start with the columns**

~~13~~
~~5~~
~~1~~
0

$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
1	1	0	1

- **Left to right, is (column) ≤ number n?**
  - If yes, put how many of that column fit in n, subtract col * that many from n, keep going.
  - If not, put 0 and keep going. (and Stop at 0)

Convert FROM decimal TO hexadecimal

- **E.g., 165 to hexadecimal?**
- **Start with the columns**

~~**165**~~
~~**5**~~
0

$16^3=4096$	$16^2=256$	$16^1=16$	$16^0=1$
0	0	10 (A)	5

- **Left to right, is (column) ≤ number n?**
  - If yes, put how many of that column fit in n, subtract col * that many from n, keep going.
  - If not, put 0 and keep going. (and Stop at 0)

Convert Binary ←→ Hexadecimal

- **Binary → Hex? Easy!**
  - Always left-pad with 0s to make full nibbles, then look up!
  - E.g., **`0b11110`** to Hex?
    - **`0b11110 → 0b00011110`**
    - Then look up: **`0x1E`**
- **Hex → Binary? Easy!**
  - Just look up, drop leading 0s
    - **`0x1E→0b00011110→0b11110`**

D	H	B
00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

☑

(Cal) Why do we use different bases?

a) Binary is used by computers, since transistors are bistable (at two values)

b) Hex is used by humans for encoding binary information because it's 4 times more efficient (number of chars)

c) Decimal because we have 10 fingers

d) The fact that computers use binary is below our level of abstraction

e) All of the above

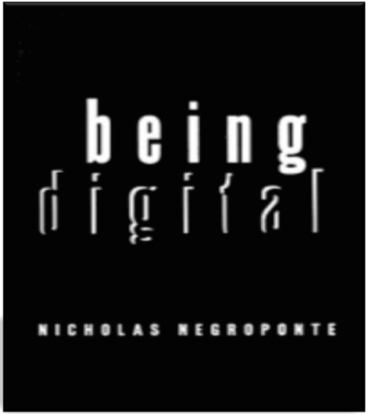
Abstraction (revisited): Digital Data

- A combination of abstractions is used to represent digital data.
- At the lowest level all digital data are represented by bits.
    - Bits can represent anything!
- Bits are grouped to represent higher-level abstractions including numbers and characters.
    - Logical values? 0 → False, 1 → True
    - Colors? 00 → Red, 01 → Green, 10 → Blue
    - Characters? 00000 → 'a', 00001 → 'b', …
- Higher-level abstractions such as Internet protocol (IP) packets, images, and audio files are comprised of groups of bits that represent different parts of the abstractions.

Interpretation of a Binary Sequence…

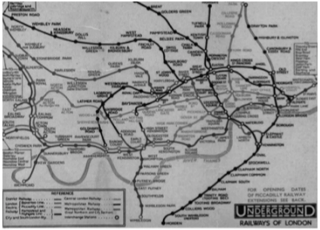
- **…depends on how it is used (e.g., as instruction, number, text, sound, or image).**
- **The sequence of bits that represents…**
  - …an instruction may also represent data processed by that instruction.
  - …a character/letter may also represent a number.
  - …a color in an image may also represent a sound in an audio file.

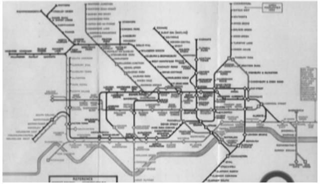
(Wikipedia)



Detail Removal Often Comes At A Cost

- **Removing detail isn't universally a positive thing.**
  - The simplification often takes out the subtlete aspects of the original
  - Cliff notes not always better than novel!





The London Underground 1928 Map & Harry Beck 1933 map.

en.wikipedia.org/wiki/Round-off_error

Overflow and Roundoff

- **Overflow**
  - When the number of represented things exceeds digits allocated for it.
  - E.g., Odometer rollover
    - 99999→00000
  - E.g., Adding 15 + 2 using 4 bits:
    - 0b1111 + 0b10 = 0b1
- **Roundoff error**
  - When the true real number can't be stored exactly given the encoding due to the fixed number of bits
    - E.g., $\pi = 3.14$
  - Sometimes this error accumulates causing problems!

Summary: Abstractions everywhere!

- **Applications and systems are designed, developed, and analyzed using levels of hardware, software, and conceptual abstractions.**
  - E.g., Mobile apps and systems
  - E.g,. Web services (both an application and a system)
- **This course will include examples of abstractions used in modeling the world, managing complexity, and communicating with people as well as with machines.**