CS 31: Introduction to Computer Systems

01: Introduction & Data Representation

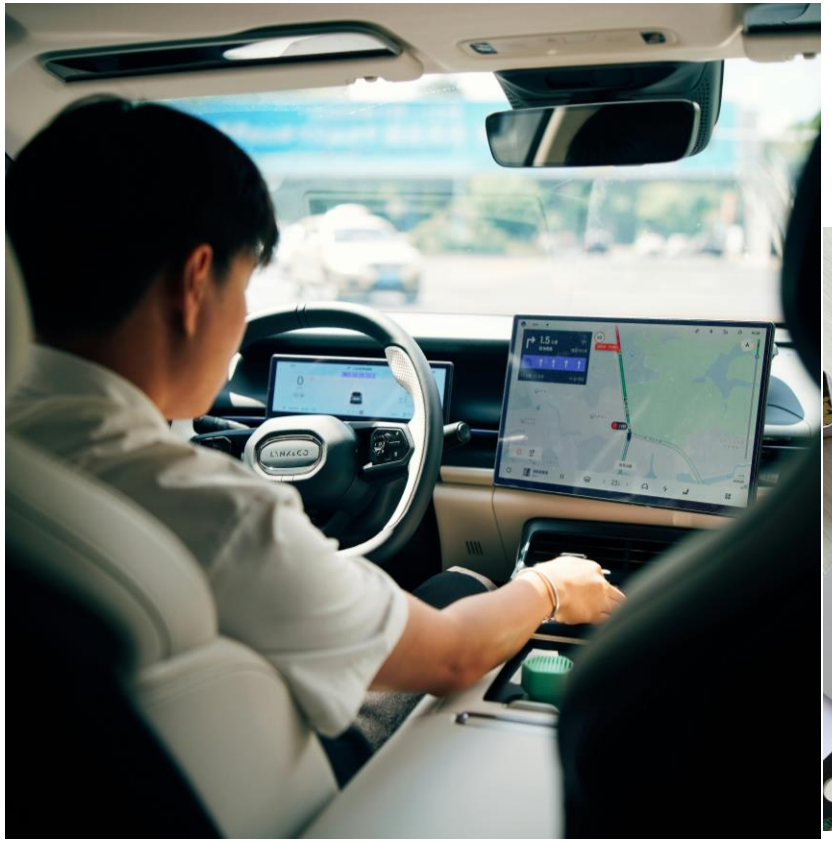
01-19-2025

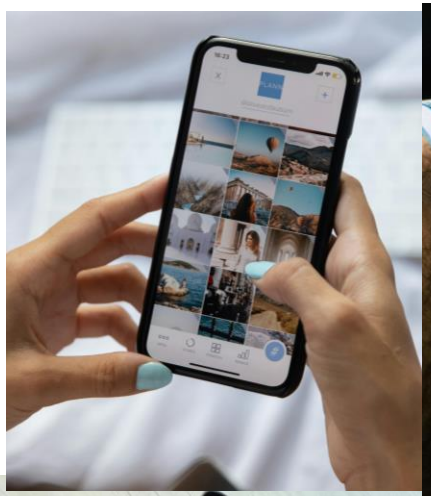
Welcome to Computer Systems!

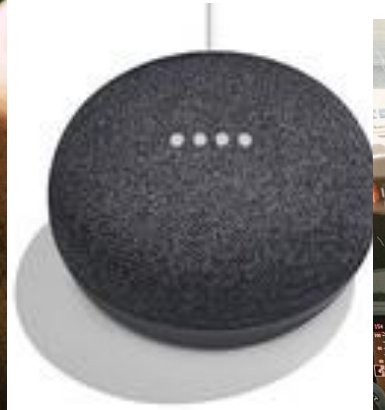
What is this class about?

.. but first what are some computer systems that you've seen or heard about?

Computer Systems







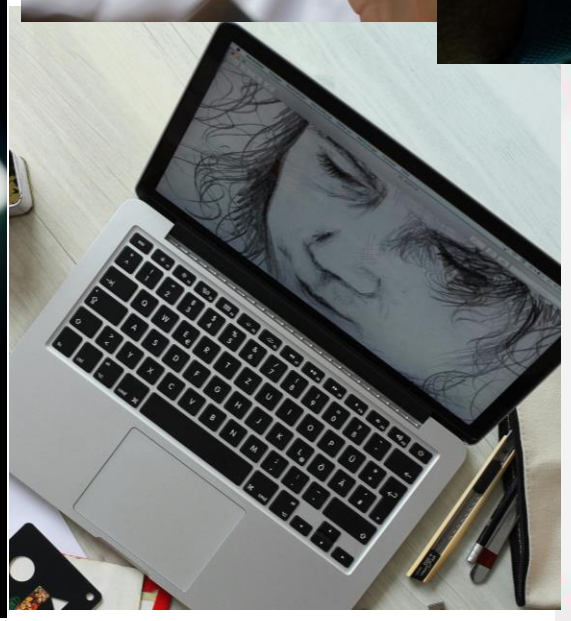


Photo by Plann: smartphone
Brett Sayle: fitness tracker
Artist Linbei: smart car
Mariya B.: Airplane Cockpit
Pierre Lecourt: Google Home Mini

Are all of these computer systems the same? What are some of the design trade-offs?

Computer Systems

Are all these computer systems the same? What are some of the design trade-offs?

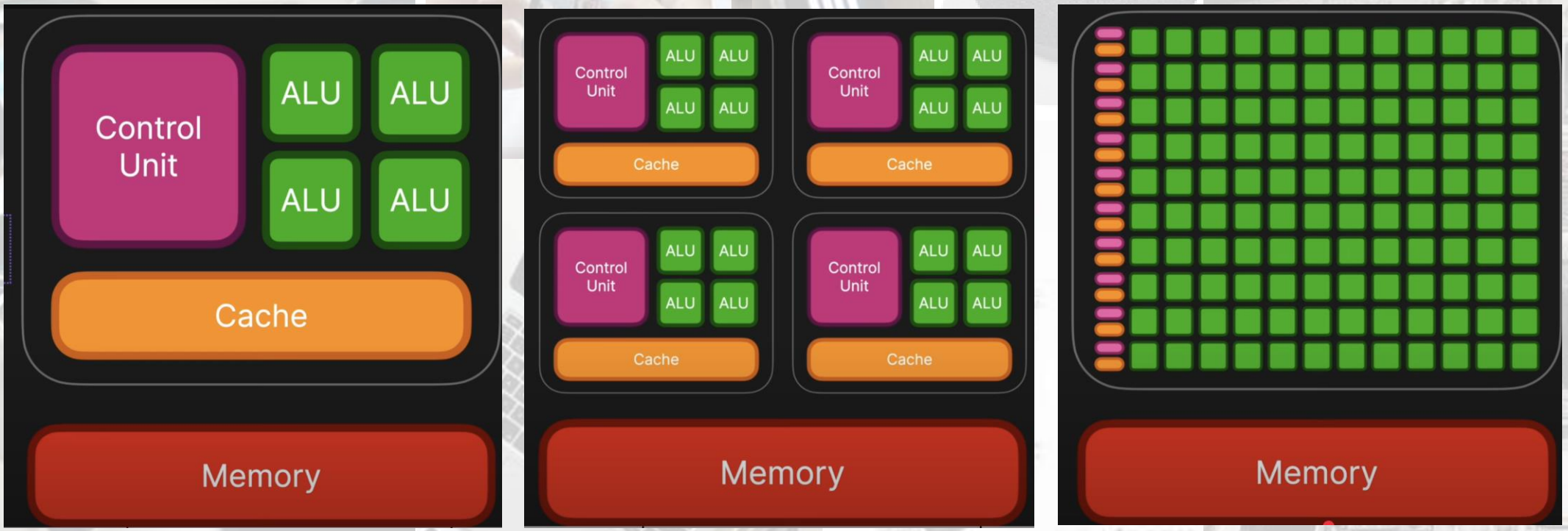
Hardware Design:
Chip Design
Number of CPUs (processing power)
Power Supply
Memory (RAM)
Storage
Cost $$

Operating System Design
File storage abstraction
Virtual Memory abstraction
Process (running program) abstractions.

Programs that can run
General purpose
Resource constrained (smart devices): run minimal set

the computer architect's role is one of making trade-offs. Working with a set of criteria (which often conflict), such as cost, speed, and reliability, the architect determines which functions, or parts of functions belong inside the boundary and which belong outside the boundary.

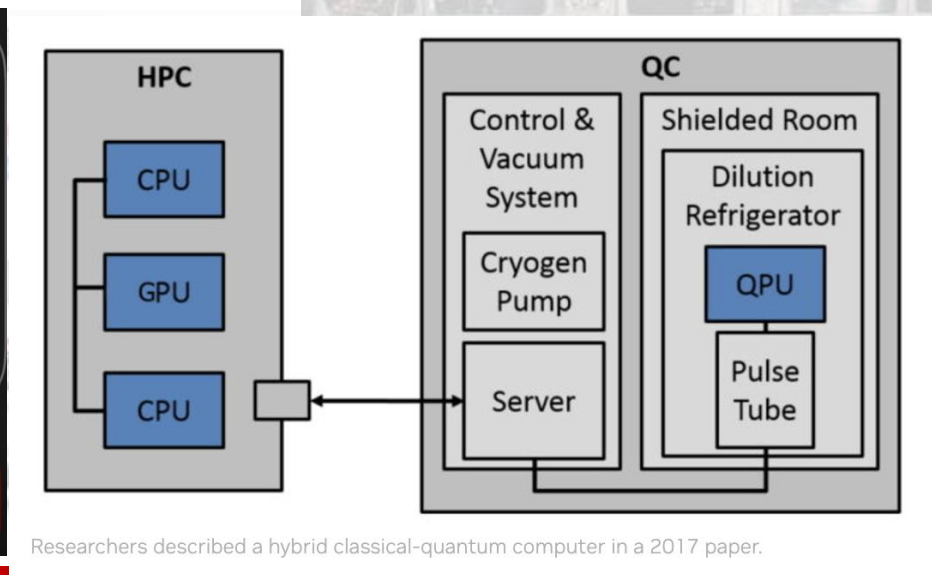
Real-world Design Trade-offs: General Purpose CPUs Vs. Specialized GPU (for Machine Learning and AI) ....



A Single CPU (Central Processing Unit) vs. a Multi-Core CPU (CPU) vs. Graphics Processing Unit (GPU)

the computer architect's role is one of making trade-offs. Working with a set of criteria (which often conflict), such as cost, speed, and reliability, the architect determines which functions, or parts of functions belong inside the boundary and which belong outside the boundary.

Not too distant future..



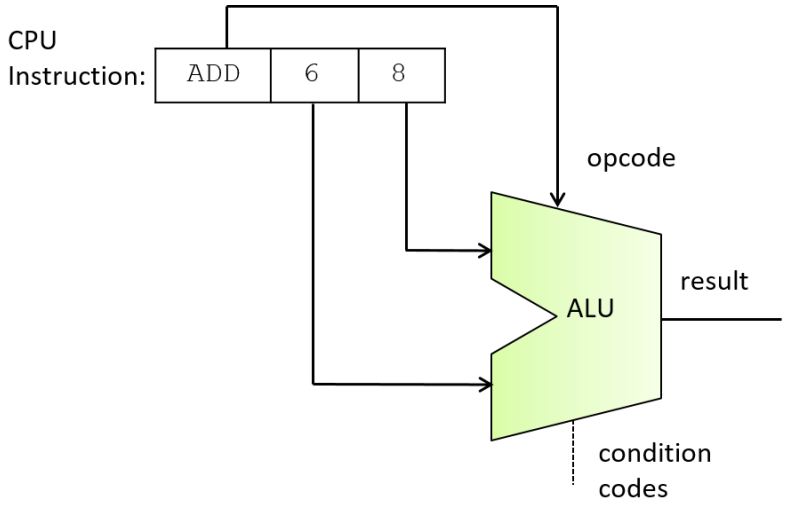
Central Processing Unit (CPUs) and multi-core CPUs vs. Graphics Processing Unit (GPU) vs QPU (Quantum Processing Unit)

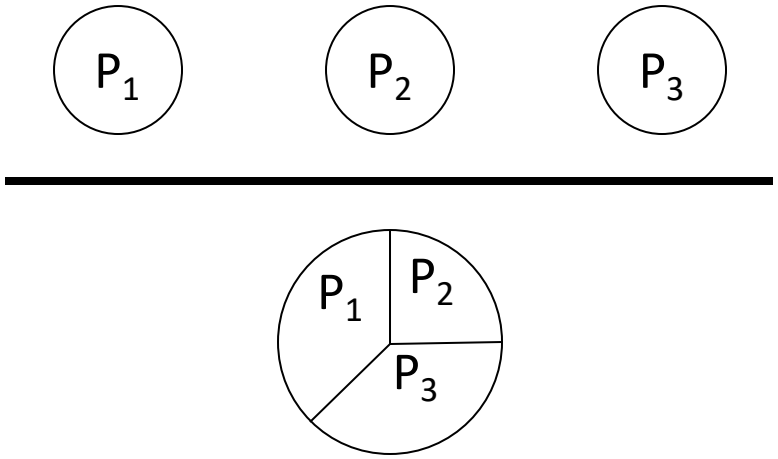
the computer architect's role is one of making trade-offs. Working with a set of criteria (which often conflict), such as cost, speed, and reliability, the architect determines which functions, or parts of functions belong inside the boundary and which belong outside the boundary.

What is a computer system?

Hardware (HW) & Special Systems Software (OS) that work together to run application programs

- HW executes program instructions
- OS that manages the computer HW
- OS also provides abstractions to the programs/users





C program:

```
// example C program
void main() {
  int authenticate;
  scanf("Enter your
  username and pwd:");
}
```

Computer Hardware

Operating System
CPU | Memory | Storage

Program

What is a computer system?

Hardware (HW) & Special Systems Software (OS) that work together to run application programs

What are the goals of our system? Correctness

- Is $x^2 >= 0$?
  - Floating point values: Yes!
  - Integers
    - 40000 * 40000 = 1600000000
    - 50000 * 50000 = ??

example: courtesy Eleanor Birrell

What is a computer system?

Hardware (HW) & Special Systems Software (OS) that work together to run application programs

What are the goals of our system? Performance

What's the big-O time complexity of the two loops below? are they identical?

```
void copyij(int src[2048][2048],
            int dst[2048][2048]){
  int i,j;
  for (i = 0; i < 2048; i++){
    for (j = 0; j < 2048; j++){
      dst[i][j] = src[i][j];
    }
  }
}
```

4.3ms

```
void copyji(int src[2048][2048],
            int dst[2048][2048]){
  int i,j;
  for (j = 0; j < 2048; j++){
    for (i = 0; i < 2048; i++){
      dst[i][j] = src[i][j];
    }
  }
}
```

81.8ms

picture: courtesy Eleanor Birrell

What is a computer system?

Hardware (HW) & Special Systems Software (OS) that work together to run application programs

What are the goals of our system? Security

```
void admin_stuff(int authenticated){
        if (authenticated){//do admin stuff
        }
}

void vulnerable(char * input){
        char data[100];
        memcpy(*input, data);
}
```

what's wrong with this code?

code: inspired by Eleanor Birrell

Also:

- C programming
- Parallel Pthread Programming
- Debugging: gdb, valgrind
- X86 Assembly Code programming
- Working Effectively with Partners & Groups
  - Lab Partner: on all lab assignments (except Lab 1)
  - HW & Study Group: for all HW assignments, and class discussion groups

About Us

- Professor Vasanta Chaganti: Lecture & Lab A
  - Please call me Vasanta (she/her)
  - www.cs.swarthmore.edu/~chaganti

- Lecturer Jocelyn Corey: Labs B, C
  - Please call her Jocelyn (she/her)
  - www.cs.swarthmore.edu/~jcorey

Office Hours:

Tuesday 2.00 - 3.30PM	SCI 253	Vasanta Chaganti
Wedneday 2:00 - 3:30PM	SCI 252-B	Jocelyn Corey
Thursday 1:00 - 2:00PM	SCI 253-B	Jocelyn Corey
Thursday 2:00 - 3:00PM	SCI 253	Vasanta Chaganti

https://www.swarthmore.edu/computer-science/ninjas

CS31 Ninjas!

What: Student Mentors

- assist in class and lab sessions
- run study sessions: help with lab assignments

Who: Ben, Erin, Jenna, Kristen, Morgan and Lona

When and Where:

- Mondays 8:00-10:00pm, Sci Cntr. 240
- Wednesdays 7:00-9:00pm, Sci Cntr. 256

Ninjas

CS Ninjas | Becoming a CS Ninja | Our Academic Support Coordinator

Resources

- EdSTEM Q&A Forum
  - https://edstem.org/us/courses/72711/discussion

- Slides & recordings on course website

- Lab sections:

Section A	Friday 8:50 AM - 10:20 AM (Chaganti)	SCI 240
Section B	Friday 2:00 PM - 3:30 PM (Corey)	SCI 240
Section C	Friday 3:45 PM - 5:15 PM (Corey)	SCI 240

Email Policy

- For *general* or *lab* questions, please use EdSTEM rather than email.
  - Your classmates benefit from your questions
  - Your classmates can answer your questions
  - I will check the forum frequently

- For *personal* questions, feel free to directly email me.

- I will attempt to respond to within 24 hours (often sooner)

Please be mindful…

Diversity, inclusion, and a mutual sense of belonging are all core values of this course. All participants in this course must be treated with respect by other members of the Swarthmore CS community. We must all strive, students and faculty both, to never make anyone feel unwelcome or unsafe in any way. Violations of these principles are viewed as unacceptable, and we take them very seriously. If you ever feel discriminated against or otherwise excluded, no matter how minor the offense, we encourage you to reach out to Vasanta, Kevin, Jocelyn or one of the college deans.

- Differing background / experience
  - Class year
  - Having taken CS 35
  - Pre-college experience

Clickers!

- Lets you vote on questions in real time.
- Like pub trivia, except the subject is always computer systems ☺

Clicker Registration

https://forms.gle/YBFvNWPTXgiySMHx5

If you don't register your clicker, I can't give you credit for quizzes / participation!

Participation scores count from week

Grading

Grades will be weighted as follows:

Percentage	Course Activity
30%	Labs
20%	Final Exam
15%	Midterm Exam 1
15%	Midterm Exam 2
5%	Written Homework
5%	Class Attendance and Participation
5%	Reading Quizzes
5%	Lab Participation

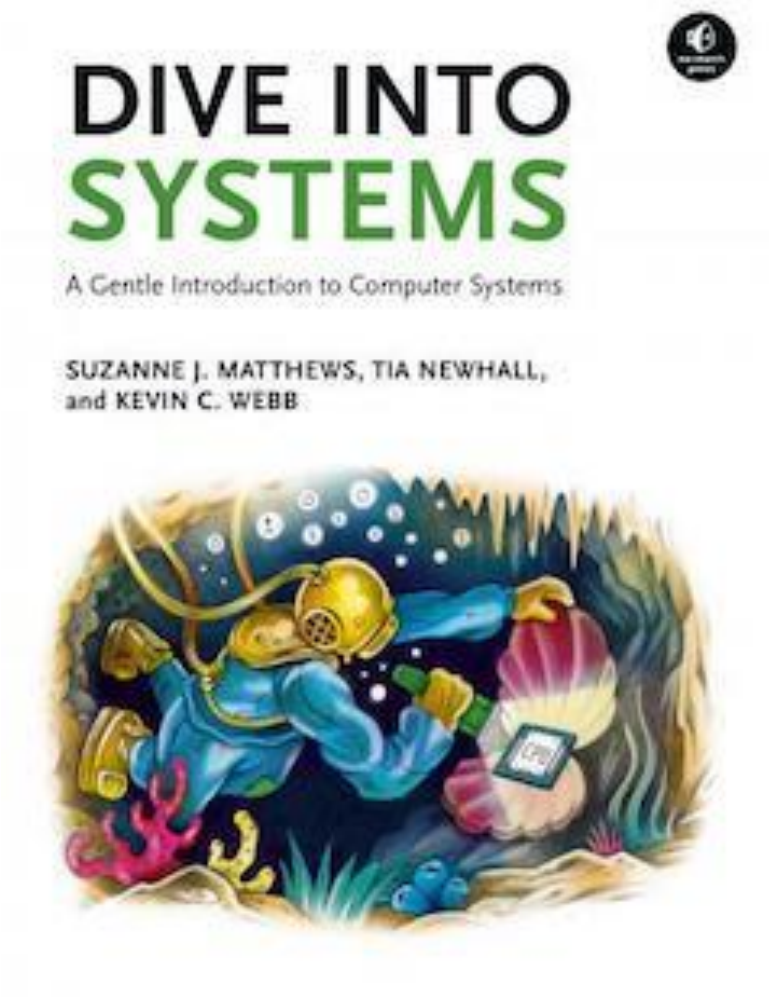
I will drop your three lowest quizzes/no-shows for class.

Reading Quizzes

- Readings from online textbook
  https://diveintosystems.org

- Target difficulty: did you read?

- Goal: incentivize / reward preparation
  - Can be an easy 5%!



Policies: Late Submissions

Genie (as William F. Buckley Jr)“ There are a few,..provisos, a, a couple of quid pro quos.” - in Aladdin

- Lab Lateness
  - 2 days of extra time for the semester (granularity of days)
  - Email AFTER you are done!
  - No Email: Grade whatever is present at the deadline.

Policies: Academic Dishonesty

- Collaboration
  - **You may discuss approaches, not solutions**
  - You must submit your own work
  - Exams may include questions on programming
- Cheating
  - We take this very seriously. It can have a negative impact on your course grade, your GPA and your record at Swarthmore and beyond.
  - **Don’t do it!**

Policies: Academic Dishonesty

- Few examples of cheating on labs
  - Screen sharing with folks not in your lab partnership
  - “Let me read my code out to you, or share the exact API for a particular function”
  - Share in words the content in your code: “I first used strncpy to copy the string up to n bytes, and then appended a null character at the end”
  - I’m applying a “security mindset” to “think like an attacker” on course assessment infrastructure
  - I just used ChatGPT to help with “some” of my assignment.

Policies: Academic Dishonesty

- Examples of how not to cheat:
  - Behave as though you are a CS ninja
  - “What approaches did you try so far?”, “Looks like you have gotten more of the string than you need to, use man pages to look at other string functions”
  - Don’t know how to help your friend? Ask them to post to EdSTEM to the class or send a post privately to me.
  - *Don’t shortcut your learning.*

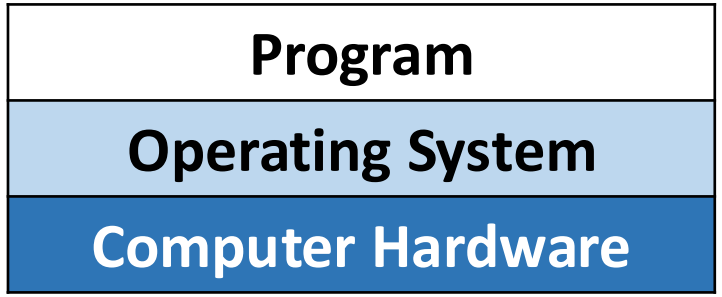
Why CS31? Or the Seven Whys

- [1 minute] Ask yourself:
  Q: Why am I taking CS31?
  A: Because I want to major in CS
  Q: Why?
  A: Because...
- [3 minutes] Turn to the person next to you and ask "Why" five more times
  Q: Why?
  A: <answer>
  ...

What will you learn?

- You are here.

How a Computer Runs a Program



How instructions & data are encoded

OS Abstractions, Resource management

How underlying HW organized & works

This week:

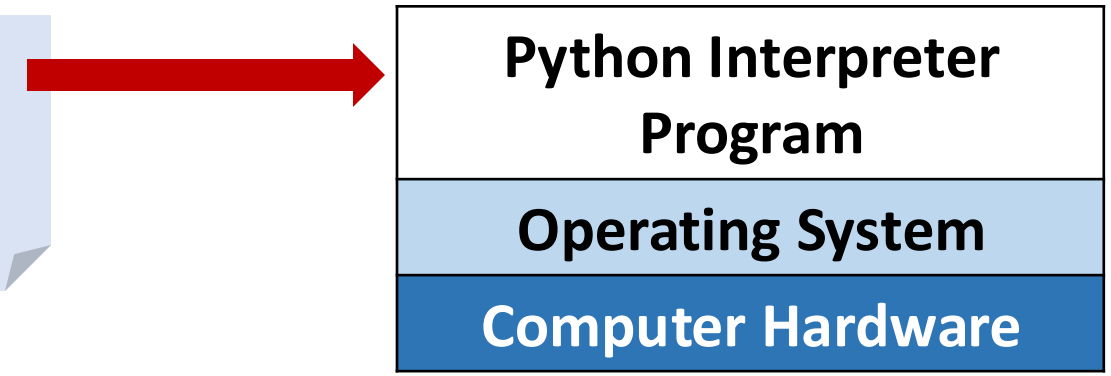
- Binary encoding of program data: 6, -3, ‘a’, 4.8
- Operations on binary data: 6 + 12, -3 + 4

Quick Look at Running Programs

CS21: Python Programs are Interpreted

- Python program is input to the Python interpreter
- The Python Interpreter is a program that runs on the Computer System (OS/HW)

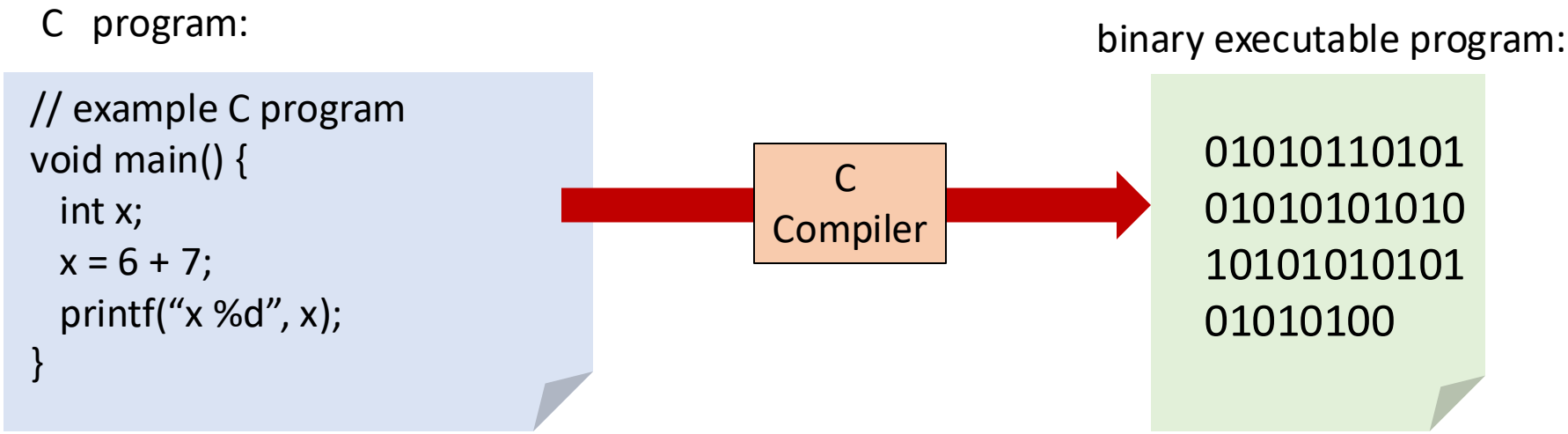
```
def main:
  x = 6 + 7
  print("x: %d" %(x))
main()
```



We will consider programs that run directly on the OS/HW (e.g. python interpreter, vim, firefox, …)

Programs that run on OS/Arch

First, program written in a high-level language is compiled (translated) into a binary (0's and 1's) executable:

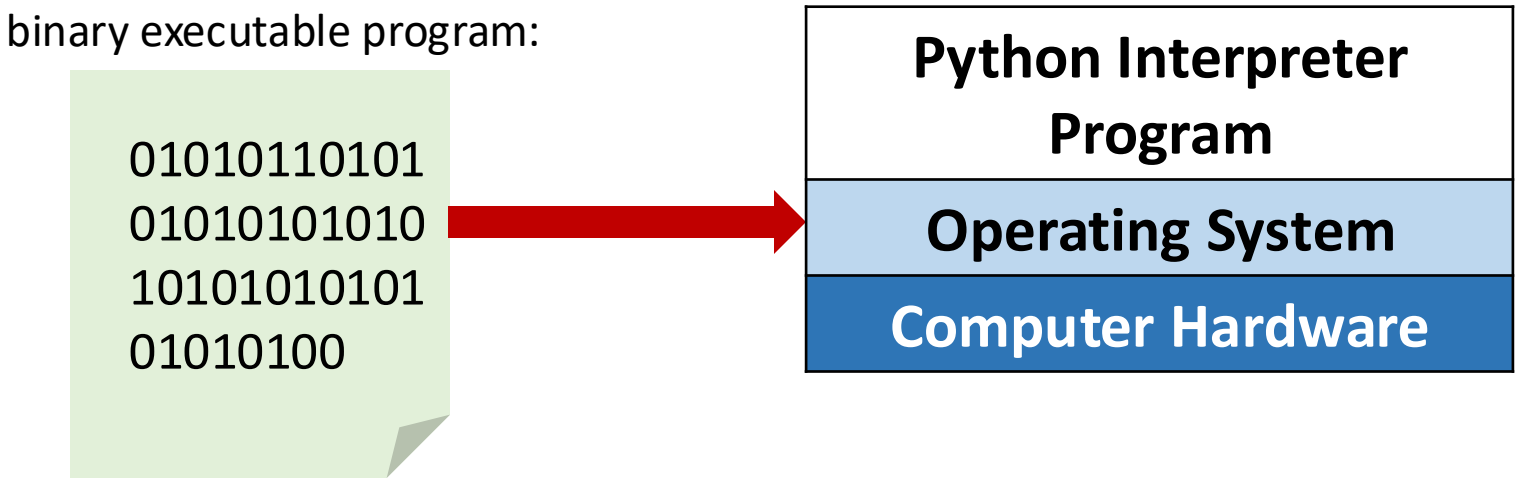


both instructions (`adding 6+7`) and the data (`6`) are translated to binary!

Programs that run on OS/Arch

W/OS help: HW circuitry runs the binary executable version of the program

- low and high voltages, representing binary values (0, 1),
  drive the HW circuitry to run the program



What we will learn this week

1. Binary Representation of program data types ex. 6, -4, ‘a’
   - C data types and sizes, bit, byte, word
   - signed and unsigned representation
2. Operations on binary data
   - Addition and Subtraction on integer types. (e.g.: 6 + 12 15 – 5 -9 + 12)
   - Some other operations on bits
   - Bit shifting, bit-wise OR, AND and NOT

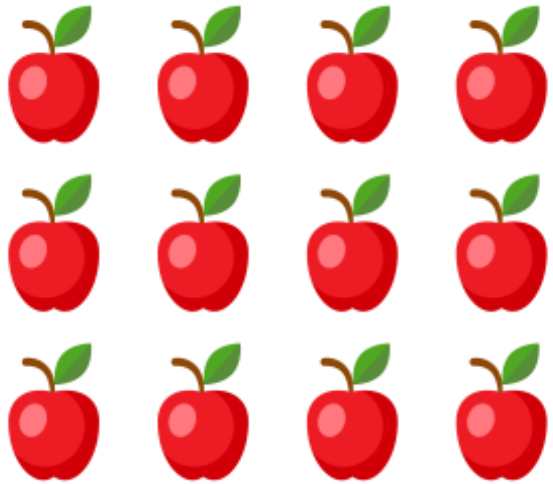
Number Representation

How many apples are there?

A. 12

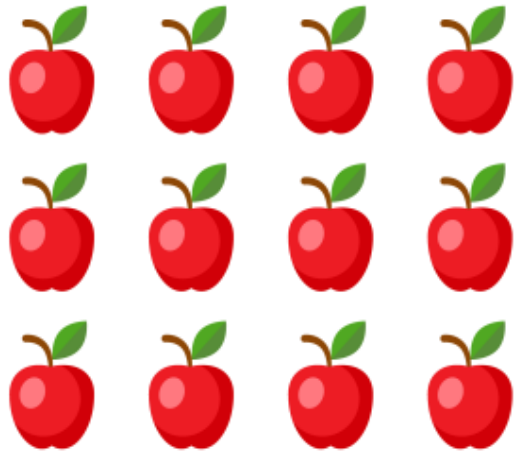
B. 1100

C. c



Number Representation

How many apples are there?



A. 12 (decimal, base 10)
B. 0b1100 (binary, base 2)
C. 0xc (hexadecimal, base 16)
D. all of these

We are using different number systems to represent the concept of twelve

- to be clear about which representation:
  - prefix binary with 0b
  - prefix hex with 0x

E.g.: Without a prefix what does “10” refer to?

decimal: 10, binary: 0b10 = 2 hex: 0x10 = 16!

Data Representation

- Decimal Number System
- Binary Number System
- Hexadecimal Number System
- … it’s all ones and zeros.. just different data representations

Bits and Bytes

- Bit: a 0 or 1 value (binary)
  - HW represents as two different voltages
  - 1: the presence of voltage (high voltage)
  - 0: the absence of voltage (low voltage)

- Byte: 8 bits, the smallest addressable unit

Memory: 01010101 10101010 00001111 …

Let's start with what we know

- Digits 0-9
- Positional numbering
- Digits are composed to make larger numbers
- Known as the Base 10 representation

Binary Digits: (BITs)

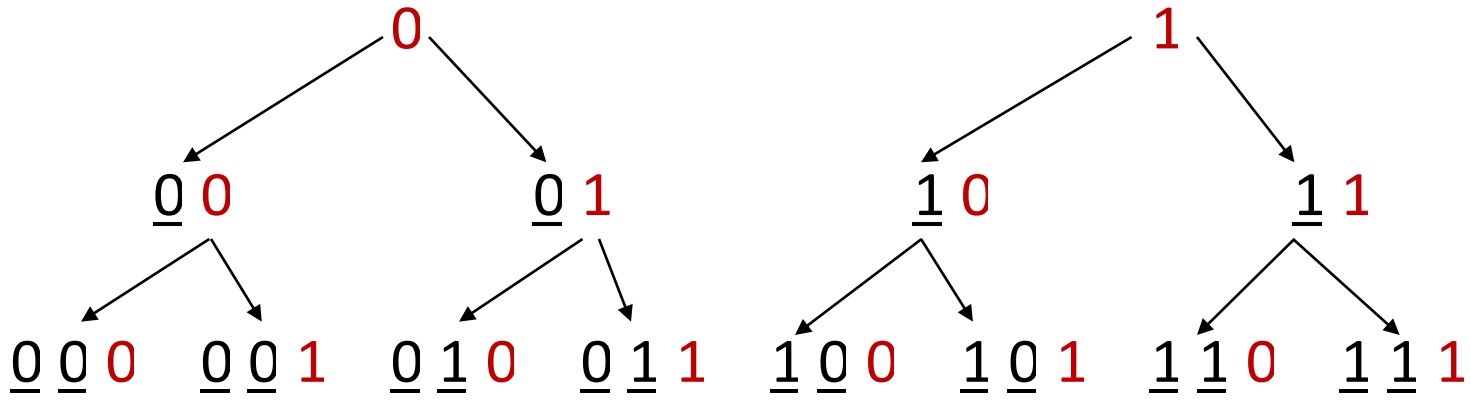
- One bit: two values (0 or 1)
- Two bits: four values (00, 01, 10, or 11)
- Three bits: eight values (000, 001, …, 110, 111)

How many values?

1 bit:

2 bits:

3 bits:



4 bits: 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 16 values
0 1 0 0 0 1 0 1 0 1 1 0 0 1 1 1
1 0 0 0 1 0 0 1 1 0 1 0 1 0 1 1
1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1

N bits: 2^N values

C Types and Sizes

In C, all variables must be declared as a specific type before they can be used:

first declare:

```
char ch;   // ch is a char type
           // (char is not a string!!)
int x;     // x is an int type
float y;   // y is a float type
```

then use:

```
ch ='a';   // ch stores 97(ascii value of a)
ch = 99;   // can also assign ch a  numeric value
x = 34;
y = -6.34;
```

(we will learn a little C this week and a lot more next week)

C Types and Sizes

- Every value is stored in some number of bytes:

  34 -6.34 'a' (stored as ascii value 97)

- Different Types have different number of bytes:

  1 byte: char, unsigned char

  2 bytes: short, unsigned short

  4 bytes: int, unsigned int, float

  8 bytes: long long, unsigned long long, double

  4 or 8 bytes: long, unsigned long

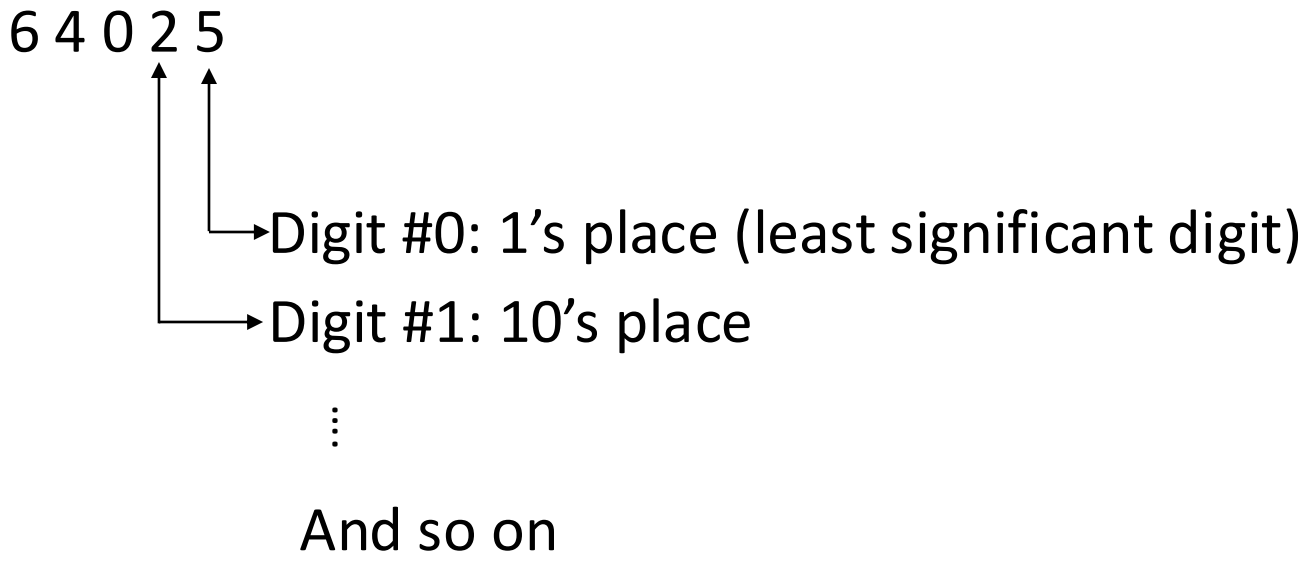
- Type specifies how to interpret a byte(s)'s value:

  11110001: -15 or 241

How do we use this storage space (bits) to represent a value?

Let's start with encodings we know:

- Decimal number system (Base 10)
- Sequence of digits in range [0, 9]

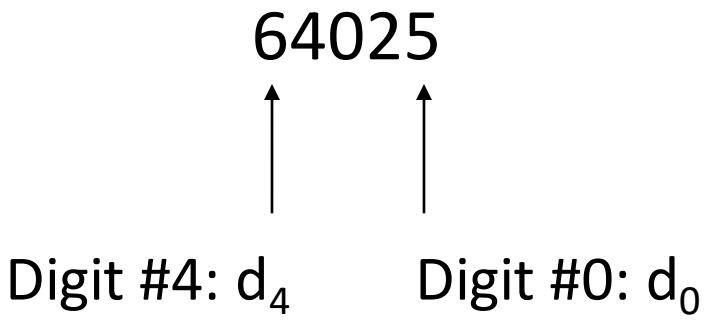


The sequence of digits represents some value

64025 in decimal represents sixty four thousand twenty five

What is the significance of the N^{th} digit number in this number system? What does it contribute to the overall value?

A. $d_N * 1$
B. $d_N * 10$
C. $d_N * 10^N$
D. $d_N * N^{10}$
E. $d_N * 10^{d_N}$



Consider the meaning of d_3 (the value 4) above. What is it contributing to the total value?

Positional Notation: Decimal Base 10

The meaning of a digit depends on its position in a number.

A number, written as the sequence of digits

$d_n d_{n-1} \ldots d_2 d_1 d_0$

where d is in {0,1,2,3,4,5,6,7,8,9},

represents the value:

$$[d_n * 10^n] + [d_{n-1} * 10^{n-1}] + \ldots + [d_2 * 10^2] + [d_1 * 10^1] + [d_0 * 10^0]$$

Positional Notation: Decimal Base 10

A number, written as the sequence of digits

$d_n d_{n-1} \ldots d_2 d_1 d_0$

where d is in {0,1,2,3,4,5,6,7,8,9},

represents the value:

$$[d_n * 10^n] + [d_{n-1} * 10^{n-1}] + \ldots + [d_2 * 10^2] + [d_1 * 10^1] + [d_0 * 10^0]$$

64025 =

$6 * 10^4 + 4 * 10^3 + 0 * 10^2 + 2 * 10^1 + 5 * 10^0$

60000 + 4000 + 0 + 20 + 5

Positional Notation: Decimal Base 10

A number, written as the sequence of digits

$d_n d_{n-1} \ldots d_2 d_1 d_0$

where d is in {0,1,2,3,4,5,6,7,8,9},

represents the value:

$$[d_n * 10^n] + [d_{n-1} * 10^{n-1}] + \ldots + [d_2 * 10^2] + [d_1 * 10^1] + [d_0 * 10^0]$$

64025 =

$6 * 10^4 + 4 * 10^3 + 0 * 10^2 + 2 * 10^1 + 5 * 10^0$

60000 + 4000 + 0 + 20 + 5

any Base number system follows this same pattern…

Positional Notation: Decimal Base 10

The value of a digit depends on its position in the number.

Pattern: $\text{Digit}_{\text{position}} * \text{Base}^{\text{position}}$ $(d_N * b^N)$

$$[d_n * 10^n] + [d_{n-1} * 10^{n-1}] + \ldots + [d_2 * 10^2] + [d_1 * 10^1] + [d_0 * 10^0]$$

$$[d_n * b^n] + [d_{n-1} * b^{n-1}] + \ldots + [d_2 * b^2] + [d_1 * b^1] + [d_0 * b^0]$$

Binary: Base 2

Used by computers: Indicated by prefixing number with **0b**

A number, written as the sequence of digits in {0,1}

$$[d_n * 2^n] + [d_{n-1} * 2^{n-1}] + \ldots + [d_2 * 2^2] + [d_1 * 2^1] + [d_0 * 2^0]$$

- 10101: $1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0$

  $= 16 + 0 + 4 + 0 + 1 = 21$

Binary Digits: (BITs)

7 6 5 4 3 2 1 0

Most significant bit ⟶ 10001111 ⟵ Least significant bit

Representation: $1 \times 2^7 + 0 \times 2^6 \ldots\ldots + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

10001111 = 143

What is the value of 0b110101 in decimal?

A number, written as the sequence of digits in {0, 1}

$$[d_n * 2^n] + [d_{n-1} * 2^{n-1}] + \ldots + [d_2 * 2^2] + [d_1 * 2^1] + [d_0 * 2^0]$$

A. 26
B. 53
C. 61
D. 106
E. 128

What is the value of 0b110101 in decimal?

A number, written as the sequence of digits

$d_n d_{n-1} \ldots d_2 d_1 d_0$ where d is in {0,1}, represents the value:

$$[d_n * 2^n] + [d_{n-1} * 2^{n-1}] + \ldots + [d_2 * 2^2] + [d_1 * 2^1] + [d_0 * 2^0]$$

A. 26
B. 53
C. 61
D. 106
E. 128

5 4 3 2 1 0 ← powers of 2

110101

$1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$32 + 16 + 4 + 1 = 53$

Different Representations

- Binary: base 2 digits [0,1]
- Decimal: base 10 digits [0, 1, …, 9]
- Hexadecimal: base 16 digits [0, …,9,a,b,c,d,e,f]

Relationship between Binary and Hexadecimal: 16 is 2^4

- each hex digit is unique permutation of 4 binary digits

  0000: 0 0001:1 0010:2 0011:3 0100:4 0101:5 0110:6 0111:7

  1000: 8 1001:9 1010:a 1011:b 1100:c 1101:d 1110:e 1111:f

Why hex? Shorthand for binary that is easier for humans to read

0011111011111010 -> 0011 1110 1111 1010 -> 0x 3 e f a

Hexadecimal: Base 16

Indicated by prefixing number with **0x**

A number, written as the sequence of digits

$d_n d_{n-1} \ldots d_2 d_1 d_0$

where d is in {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F},

represents the value:

$[d_n * 16^n] + [d_{n-1} * 16^{n-1}] + \ldots +$

$[d_2 * 16^2] + [d_1 * 16^1] + [d_0 * 16^0]$

What is the value of 0x1B7 in decimal?

A. 397
B. 409
C. 419
D. 437
E. 439

$[d_n * 16^n] + [d_{n-1} * 16^{n-1}] + \ldots +$

$[d_2 * 16^2] + [d_1 * 16^1] + [d_0 * 16^0]$

$16^2 = 256$

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

What is the value of 0x1B7 in decimal?

A. 397
B. 409
C. 419
D. 437
E. 439

$[d_n * 16^n] + [d_{n-1} * 16^{n-1}] + \ldots +$

$[d_2 * 16^2] + [d_1 * 16^1] + [d_0 * 16^0]$

$16^2 = 256$

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

Converting between Hex and Binary

Hex to binary:

expand each hex digit into its 4 binary digits:

0xa12f: a 1 2 f

1010 0001 0010 1111

0b1010000100101111

Binary to hex:

group into sets of 4 digits

convert each set of 4 to a single hex digit:

0b1001010100001111: 1001 0101 0000 1111

9 5 0 f

0x950f

High-level Takeaway

- You can represent the same value in a variety of number systems / bases.

- It's **all** stored as binary in the computer.
  - Presence/absence of voltage.