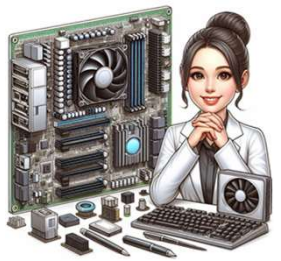
CSC252: Computer Organization

Lecture 0 - Introduction

Prof. Diana Diazh
(she/her/hers)
University of Arizona
Spring 2025



WELCOME!

We are very excited to introduce you to the amazing world of computer organization

Lecture 0: Overview

Today we'll discuss:

1. Syllabus
2. How to store data in a computer?

So we can understand:

1. Classroom rules and responsibilities
2. Prerequisites for this course

Introductions

About me

Associate Professor of Practice
Diana Diazh, MS, PhD
(she/her/hers)

ddiana@cs.arizona.edu
https://www.linkedin.com/in/dianadiazh/
My Office: Gould-Simpson 835

Call me Diana or Prof. Diazh (/ “DEE”+ “az”/)

Teaching experience:

- Computer Design
- Machine Organization
- Data Science
- Database systems
- Python
- OOP
- Bioinformatics I
- …

Research interests:

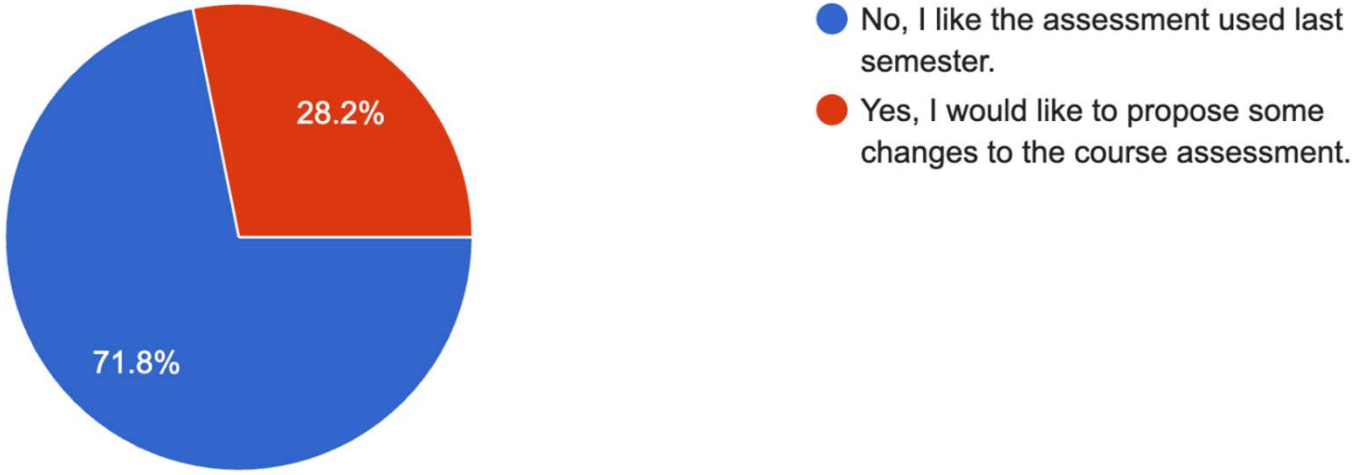
- Data Science
- CS education
- Bioinformatics
- Finance
- Sustainability

About your classmates (127 students enrolled)

Thanks for filling out the survey about expectations!

Please read the "Grading Scale and Policies" from last year's syllabus https://drive.google.com/file/d/18bPVK-ys...hanges to the course Grading Scale and Policies?

71 responses



Teaching Assistants

- What do they do?
  - Help answer questions in the classroom and the class question/answer forum (Discord)
  - Hold office hours (Gould Simpson Building - room 813)
  - Grade your assignments
  - Help grade tests
  - Act as a “point of contact” for course questions, grading, and help in understanding concepts covered

About our TAs

Valerie Militeeva (Course Coordinator)

Chitrangada Juneja (CCIT)

Ellie Laton (Sr. TA)

Jesse Oved

Asadkhon Rasulov

Akbarali Sodikov

Bryan Wang

Ameya Madhugiri

- TA Introductions!
  - Advice for students, Interesting Programming Project, Unpopular opinion, or Favorite CS topic

Our Values

- Probably you will not understand everything right away - it will take a few tries
- The future of CS is in this room
- This is an inclusive space - diverse voices will make us stronger, let’s be respectful
- We embrace (and encourage) failure.

You belong here.

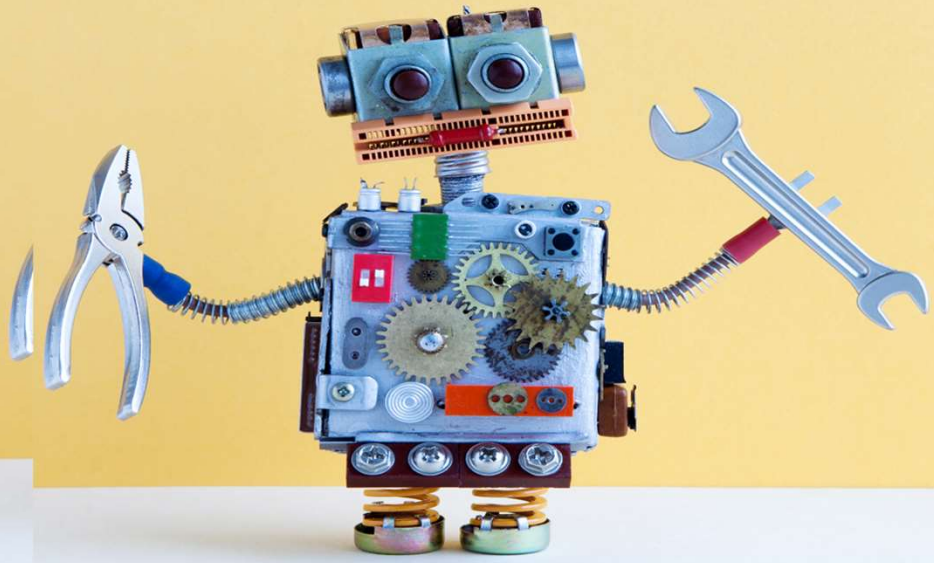
Part of first assignment

- Attend office hours next week
- I'll send the details over Discord

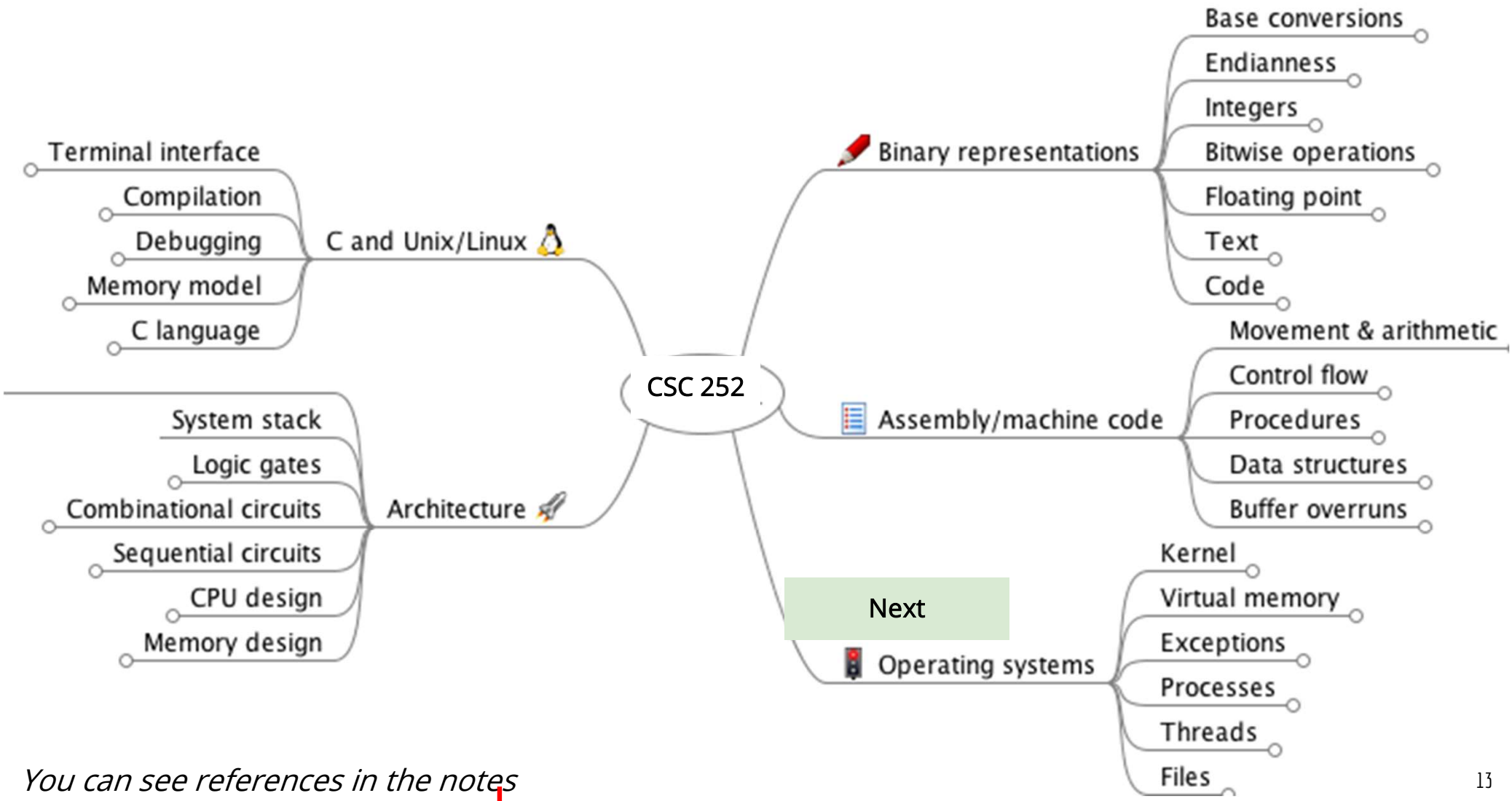
What is CSC252?

- Prerequisites
  - CSC 210
  - Familiarity with Java
- Key Topics
  - Assembly language (MIPS)
  - CPU design (logic, ALUs, pipelining, cache)
  - Some C

We will look under the computer “hood”



Curriculum



You can see references in the notes

Key Websites

- Class homepage https://profdiazh.github.io/csc252/
  - syllabus, lectures, assignments posted here
- Questions: Discord server managed by Tas
  - sign up and use your name or netid
- Grades: http://d2l.arizona.edu
- Reading: Zybooks
- Submit assignments: Gradescope

Grading Breakdown

5%	zyBooks Readings	drop 1 week	50 points
24%	Assembly Projects	drop 1 asm	240 points
21%	Simulation Projects	drop 1 sim	210 points
5%	Homework assignments - exam prep	drop 1 homework	50 points
30%	Tests	worst replace with final	300 points
15%	Final		150 points

Online Textbook

- **Required** Readings
  - Open from d2l
  - You can access for free for 30 days

- Activities are embedded in the textbook
  - See the zyBooks website for the due dates
  - Reading a head of time will help you prepare for lecture

Exams

- Tests
  - Every 2 weeks (6 total)
  - 30 minutes long
  - Stay for lecture afterwards
- Final Exam
  - Wednesday, May 14,
  - From 6:00 to 8:00 pm
- All exams:
  - Closed book and notes
  - No calculators
  - No make-up

Homework

- Homework assignments help you study for the test
  - Due at 5pm, 2 days **before** the test
  - Turn in through Gradescope
  - Will be checked for completion, not graded for correctness
- Group work **encouraged**
- But everybody must turn in their own copy

Projects

- Due on Wednesday at 7 pm, most weeks
- TAs will stop helping at 5pm
- Turn in to Gradescope
- Some projects focus on writing assembly
- Others focus on hardware simulation
- Work **individually**

- **Use the grading scripts!**

Late Days? Makeup?

- **No late days**

  Sorry, the paperwork is too much!

- Emergencies
  - Contact the dean of students to get the excuse
  - Email me, I'll be happy to work with you
  - But we drop some assignments

Regrade Requests

- Regrade requests due **7** days after grades are announced to you
  - Later than that, we generally don't even consider them
  - Use Gradescope's "Regrade Request" feature
  - If you ask for a regrade and got a replied, but you still want me to check your request, email me or attend my office hours

Grading Scale

- Standard 90/80/70/60 scale
  - Might lower the bar, never raise it
  - Extra credit
    - In-class participation

Contacting Us

- For general questions use Discord
  - Other students probably want to know the same thing!
- For questions about your code
  - Email us or DM TAs
- For grading questions
  - Email me
- If you have a private matter
  - Email me

Academic Integrity

YOUR WORK MUST BE YOUR OWN

Academic Integrity

- OK to discuss problems *"in broad terms"*
  - "What is Diana asking us to do?"

- **NOT OK** to discuss solutions
  - "How did you solve the problem?"
- **NOT OK** to look at others' code or share your code

Academic Integrity

- Always OK to talk to TAs & Instructor
  - Ask anything!
  - We might say “can't tell you”
  - We'll never penalize you for a question
- Discord a good place for general questions
  - Clarification of assignments

Academic Integrity

Penalties:

- Minimum:
  - **negative** 100% on the assignment
    - Better to skip the assignment than cheat!
  - Report to the Dean of Students
- Can be more!

Academic Integrity

Hard Coding:

- Stuck on a problem? Tempted to just "solve the testcases?" **Don't!**

- Secret testcases
- Academic Code of Integrity

Classroom behavior

- Treat other respectfully
- Do not be noisy or disruptive during lecture
- Use your laptop but only for note-taking and following along with lecture, coding, In-class Activities
- You may not chat, game, facebook, netflix, etc

ICA

Introduce yourselves!

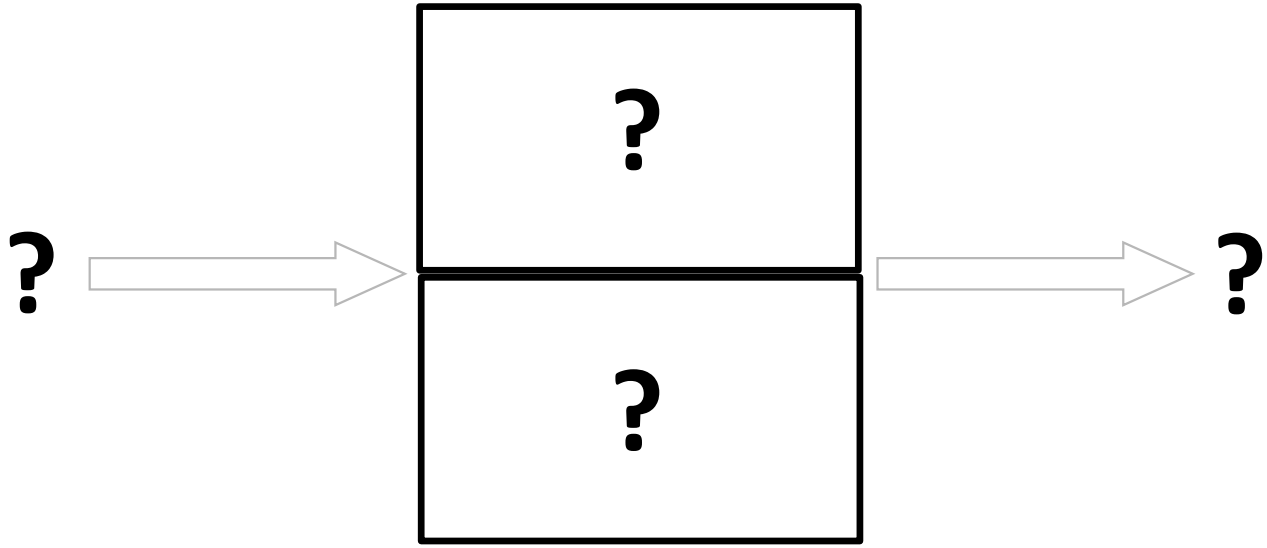
1. Go around your table and introduce yourselves to everyone
    a. Say your name/major/year
2. Come up with a name for your group
    1. Find one thing you all have in common

Question 1

1. **Submit** the answers to the following questions to Gradescope as a group (1 person of the group submits)
    1. Group name
    2. What assignments are due on Monday?
    3. What assignments are due on Wednesday?

Course Overview

What is a computer?



Which of these is (or have) a computer?

A

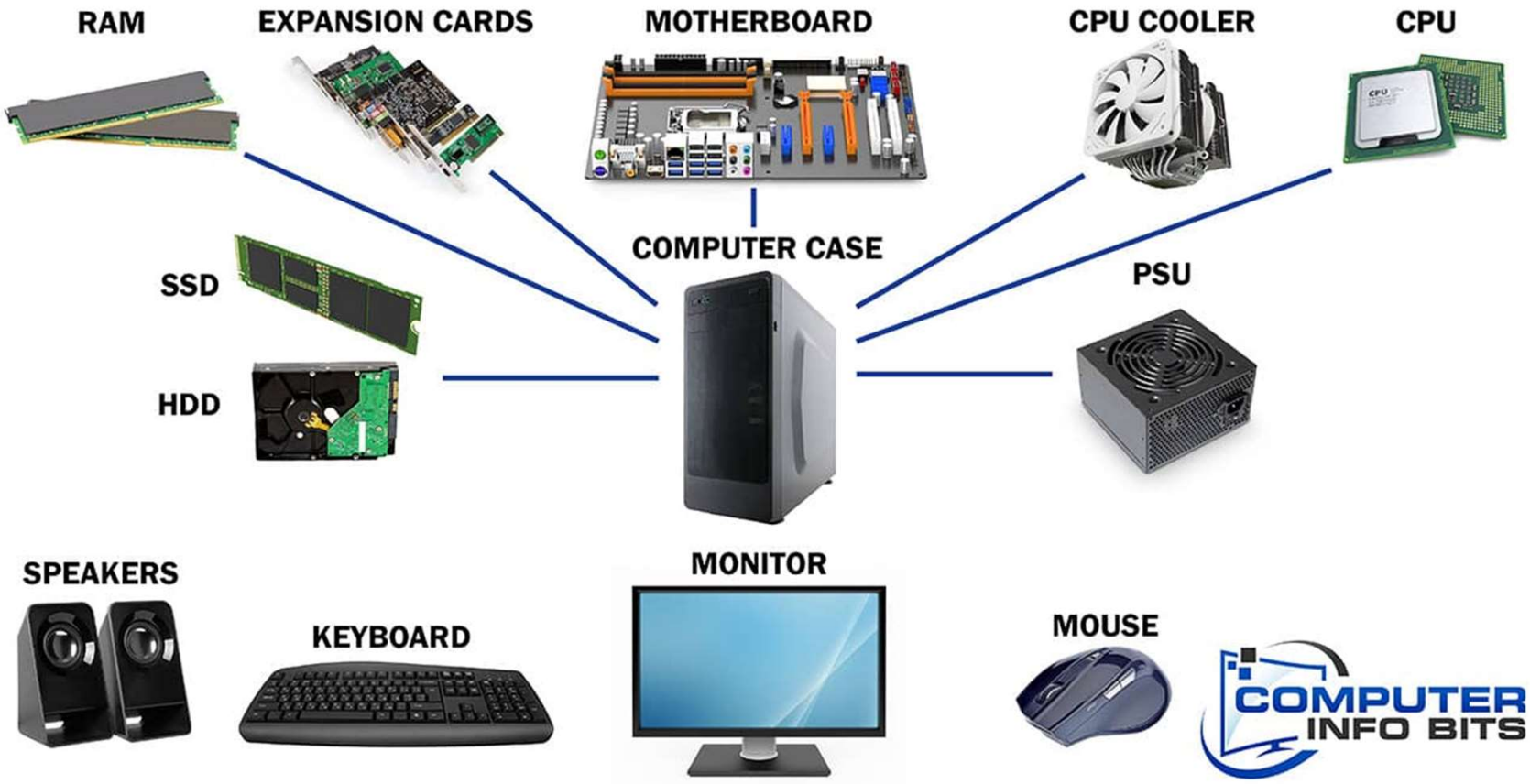
B

C

D

E ALL OF THE ABOVE

PARTS OF A COMPUTER

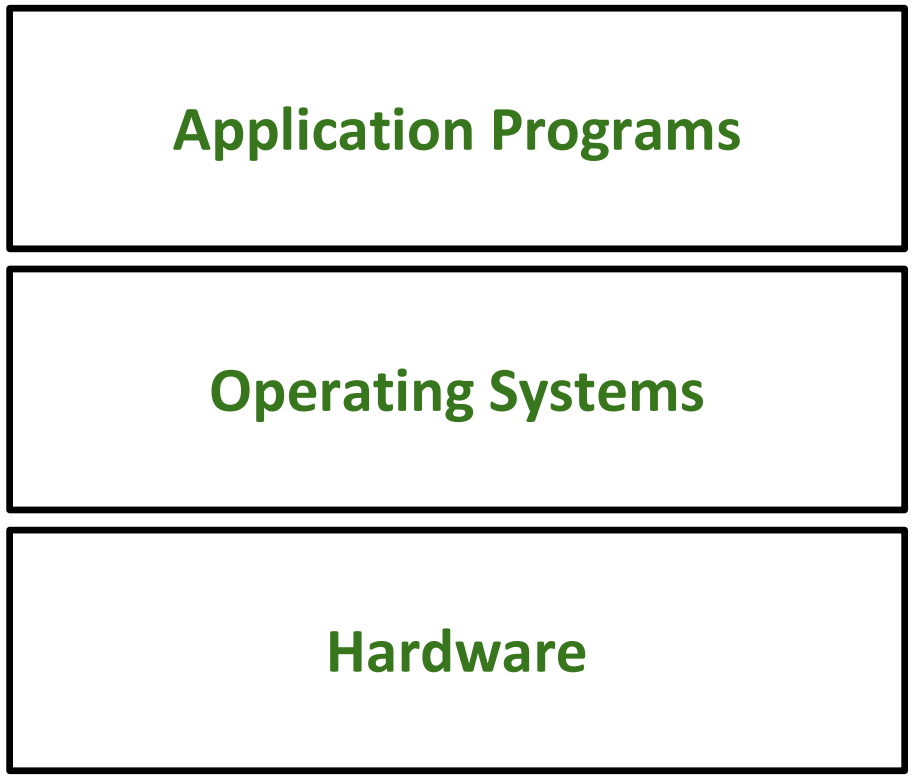


Question 2

1. Explain using a drawing on your whiteboards
   How do computers work?

1. Take a picture of your drawing and upload it to Gradescope

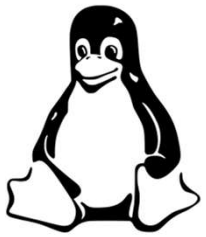
How does a program run?



Example

Application Programs

Google Chrome



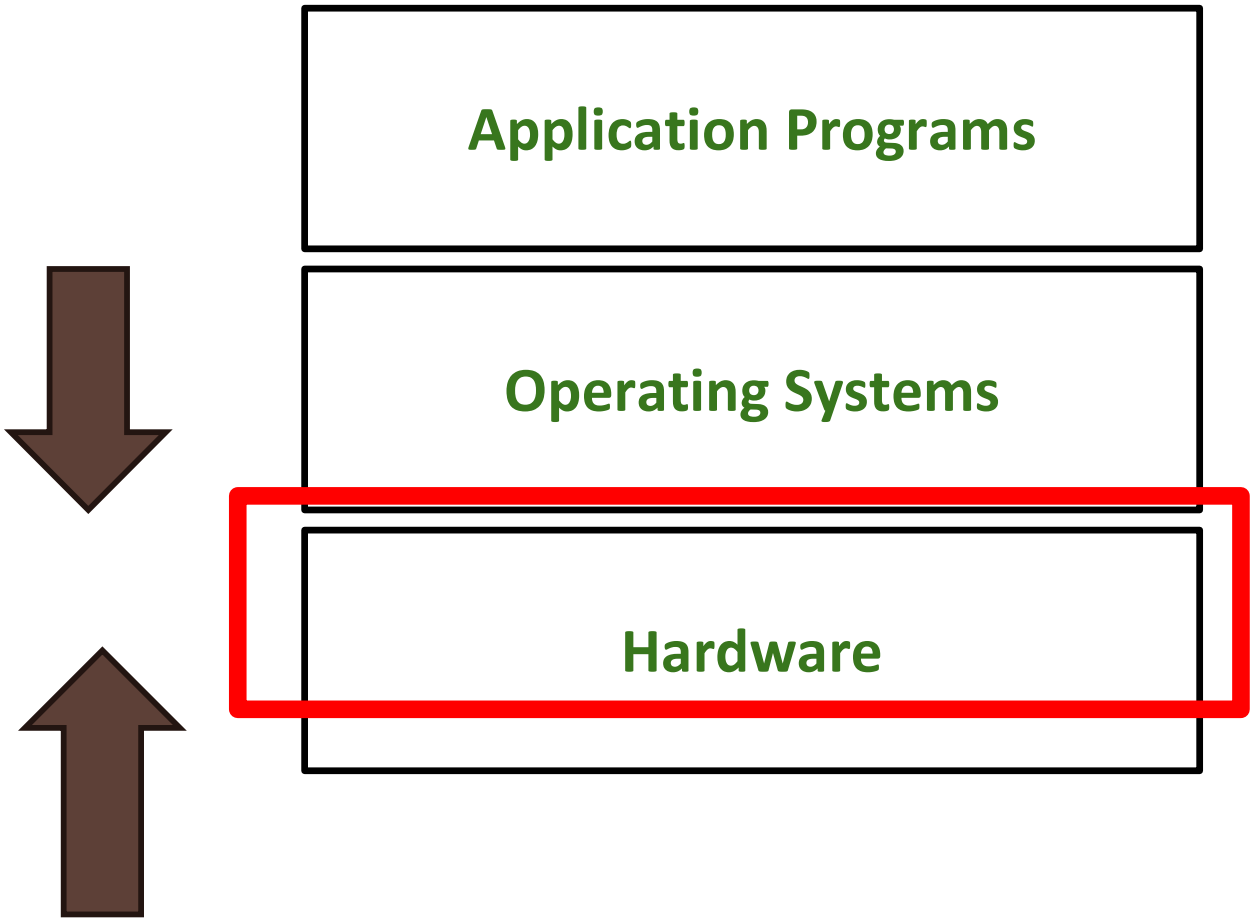
Operating Systems

Ubuntu

Hardware

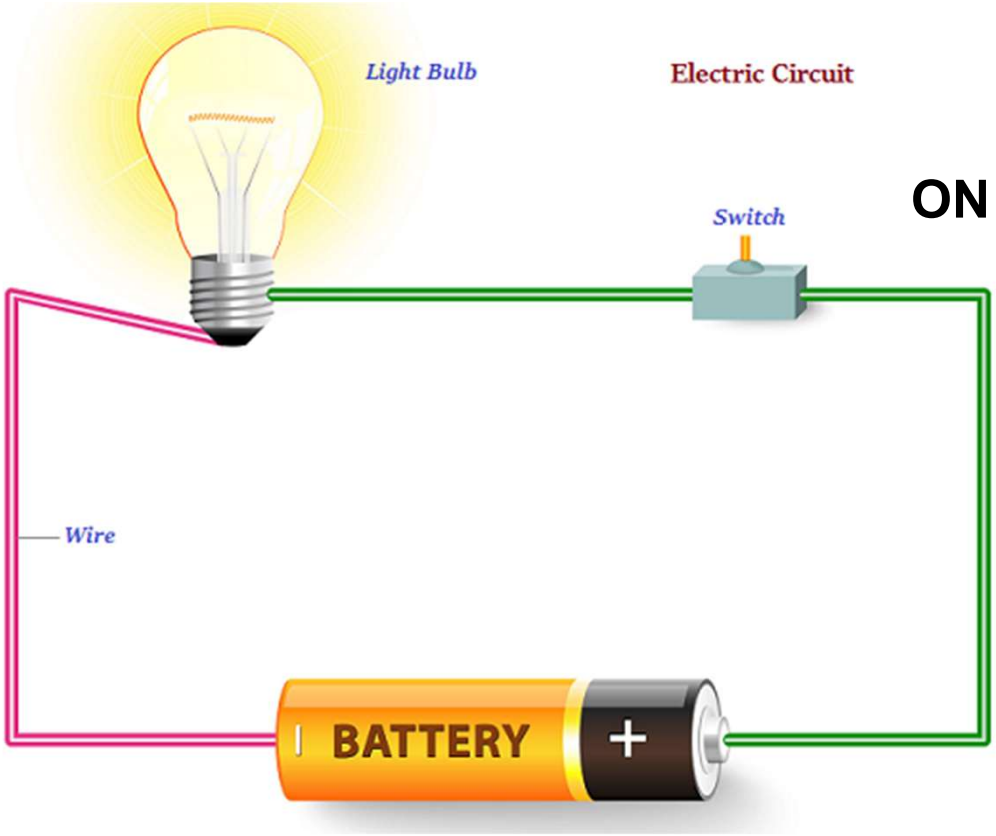
AMD x86-64

In this class



How computers work?

Electricity

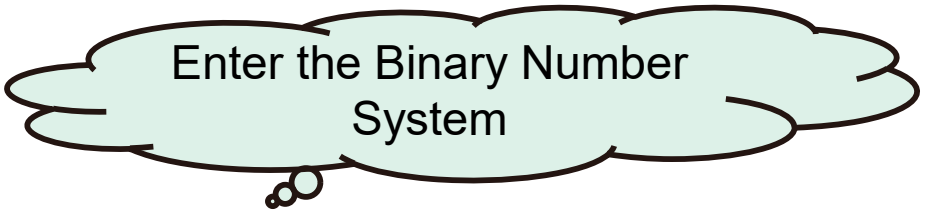


ON / OFF = bit

Everything in the computer is Voltage or Not Voltage (on/off) (1/0) (true/false)

In the computer everything is ones or zeros (binary)

- Numbers
- Strings
- Images
- Videos
- VR games
- Files
- Programs
- Operative Systems
- Compilers
- everything….



Let’s store a number in a computer

How can humans represent numbers?

Eastern Arabic numerals	10	٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩	8th Century
Vietnamese numerals (Chữ Nôm)	10	𠬠 𠄩 𠀧 𦊚 𠄼 𦒹 𦉱 𠔭 𠃩	<9th Century
Western Arabic numerals	10	0 1 2 3 4 5 6 7 8 9	9th Century
Glagolitic numerals	10	Ⰰ Ⰱ Ⰲ Ⰳ Ⰴ Ⰵ Ⰶ Ⰷ Ⰸ ...	9th Century
Cyrillic numerals	10	а в г д е ѕ з и ѳ і ...	10th Century
Rumi numerals	10	𐹠 𐹡 𐹢 𐹣 𐹤 𐹥 𐹦 𐹧 𐹨	10th Century
Burmese numerals	10	၀ ၁ ၂ ၃ ၄ ၅ ၆ ၇ ၈ ၉	11th Century[10]
Tangut numerals	10	□□□□□□□□□□	11th Century (1036)
Cistercian numerals	10		13th Century
Maya numerals	5&20		<15th Century
Muisca numerals	20		<15th Century
Korean numerals (Hangul)	10	영 일 이 삼 사 오 육 칠 팔 구	15th Century (1443)
Aztec numerals	20		16th Century
Sinhala numerals	10	□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□	<18th Century
Pentadic runes	10		19th Century

We use western Arabic numerals

- decimal system (base 10)
- Use digits 0 – 9
- Each position changes the value by a multiple of 10.
- 132 = 1 × 100 + 3 × 10 + 2 × 1

How many dogs do you see in this picture?

write your answer in **decimal** system using 3 decimal places

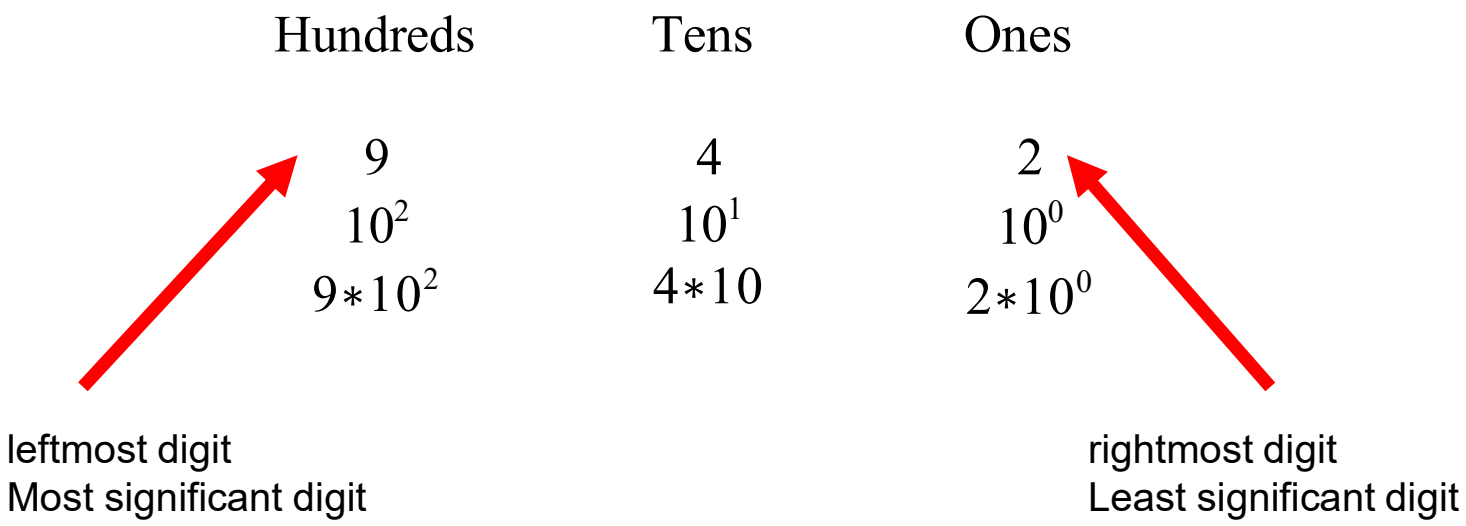
0	0	6

3 decimal places

Number Systems

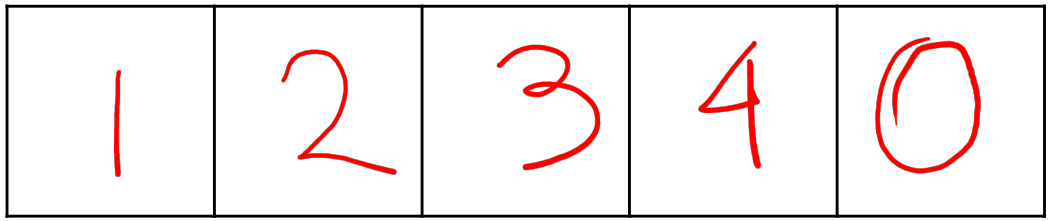
- Base-10 uses powers of 10 for each column

  942 is:



Answer in your whiteboard

Let's say you are receiving a price of $12345, but you have to change one of these five digits to zero.
Which digit would you change to maximize your price?



5 decimal places (aka 5 digits)

Range of Values

- Three digit number:

Hundreds	Tens	Ones
9	9	9
10^2	10^1	10^0
$9*10^2$	$9*10$	$9*10^0$

- Max value for base 10: $999 = 10^3 - 1$
- Max value for any base: $b^k - 1$
  - Base *b*, with *k* digits

Addition

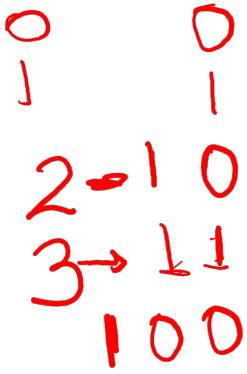
- To add:
  - Add each column, from right to left
  - If answer is too large (2 digits), carry to next column

$$\begin{array}{cccc} 1 & & 1 & \\ & 9 & 4 & 2 \\ + & 2 & 4 & 9 \\ \hline 1 & 1 & 9 & 1 \end{array}$$

Question 3 on Gradescope

- How much do you know about the binary system?

Question 4: Binary Addition



	0	0	1	0	1	0	1	0
+	0	0	0	1	1	1	1	1
	0	1	0	0	1	0	0	1

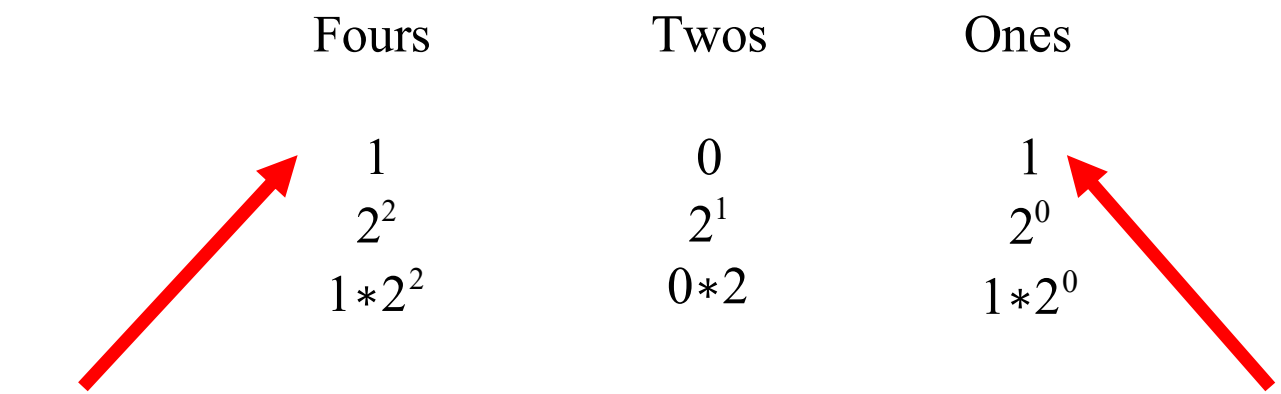
Group Exercise:

- Add using binary (watch for carries)
- Convert both numbers to decimal
- Convert answer to decimal
- Check

Number Systems

- Base-2 uses powers of 2 for each column

  101 is:



leftmost **bit**

Most significant **bit (MSB)**

rightmost **bit**

Least significant **bit (LSB)**

Storing numbers in a computer

Storing binary numbers

- Why Binary?
  - Simple electrical implementation

    Max Voltage = 1 Min Voltage = 0

- *bit* A binary digit (one wire)
- *nibble* 4 bits
- *byte* 8 bits
- *half-word* *16 bits (*defined by the system)
- *word* *32 bits (*defined by the system)

A Byte

- Binary: Each column is worth a power-of-2

				$8's$		$4's$		$2's$		$1's$	
				2^3		2^2		2^1		2^0	
1	0	1	0	1		0		1		1	
				8	+	0	+	2	+	1	=
											1_{te}

$0 \quad 2$

- Max value in a byte: $2^8 - 1 = 255_{ten}$

A Byte

- Binary: Each column is worth a power-of-2

$128's$	$64's$	$32's$	$16's$	$8's$	$4's$	$2's$	$1's$	
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
1	0	1	0	1	0	1	1	
128 +		+	+ 0 +	8 +	0 +	2 +	1	= 17_{ten}
		3						1_{te}
	0	2						

- Max value in a byte: $2^8 - 1 = 255_{ten}$

How many dogs do you see in this picture?
write your answer in a byte

Reading and writing binary is difficult for humans

Hex and Octal

- Hexadecimal (base 16) and octal (base 8) are **shorthand for binary**
- 1 octal digit = 3 bits
- 1 hex digit = 4 bits

Binary	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Octal

- Counting (and adding) works like any other base:

Octal: 65, 66, 67, 70, 71, 72 …

75, 76, **????**

Converting to/from Octal

- Group bits into packs of 3 (**starting at the right**)
- Convert each group to octal

Convert to Octal:

110111_2

1000_0000_2 (byte)

$1010\_1010\_1110\_0111_2$ (half-word)

Converting to/from Octal

- Group bits into packs of 3 (**starting at the right**)
- Convert each group to octal

Convert to Octal:

110111_2	67_8
1000_0000_2	200_8
$1010\_1010\_1110\_0111_2$	125347_8

- Unfortunately, octal digits don't line up with bytes!

Hexadecimal

- Base 16: digits 0-9, then a-f (or A-F)

Binary	Hex	Binary	Hex
0000	0	1000	8
0001	1	1001	9
0010	2	1010	a
0011	3	1011	b
0100	4	1100	c
0101	5	1101	d
0110	6	1110	e
0111	7	1111	f

Why Hexadecimal?

- Dense, easy to type & read
- 2 hex digits = 1 byte

- In C (and thus, in many languages), hex constants start with `0x`

Why Hexadecimal?

- Dense, easy to type & read
- 2 hex digits = 1 byte

- Great constants

```
0xdeadbeef
0xbaadf00d
0x0000000ff1ce      (MS Office)
0xc00010ff          (iOS error - overheat)
face:b00c           (part of an IPv6 addr)
```

Decimal to/from Octal, Hex

- Possible to go direct, but annoying
- Easier: go to binary, then to octal/hex

Class Exercise:

Convert the following decimal numbers to octal and hex:

```
17
127
1041
65534
```

Decimal to/from Octal, Hex

- Possible to go direct, but annoying
- Easier: go to binary, then to octal/hex

Class Exercise:

Convert the following numbers to octal and hex:

17	1_0001_2	21_8	11_{16}
127	111_1111_2	177_8	$7f_{16}$
1041	$100\_0001\_0001_2$	2021_8	411_{16}
65534	$1111\_1111\_1111\_1110_2$	177776_8	$fffe_{16}$