Coding Fundamentals ASPIRE

(8/11 - 12/12)

Helix Charter High School
Isabelle Viraldo

Week 14

Welcome!

- Mondays: Discussion + Activity
- Fridays: Review + Programming Exercise

What do you want to learn?

What do you care about?

What do you want to accomplish?

feedback!

Helix Charter High School
Isabelle Viraldo

Week 14

Topics I hope to cover:

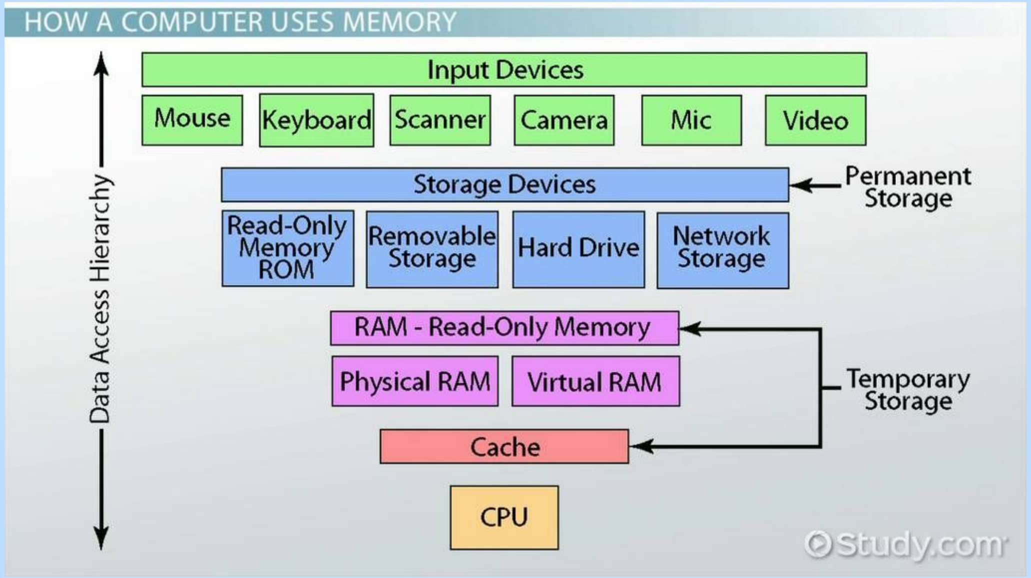
- GitHub (How to use and let’s set one up!)
- AI (Machine Learning vs Generative AI vs Image Detection, let’s break it down (and make one of our own))
- How to code! (Some practical skills, and also best practices)
- Binary (What is it? Why is it important? Who cares?)
- Robotics (What do you need to get a robot working?)
- How does your computer work? (What do computers do when you’re not looking?
- What do you want to learn?

Week 14

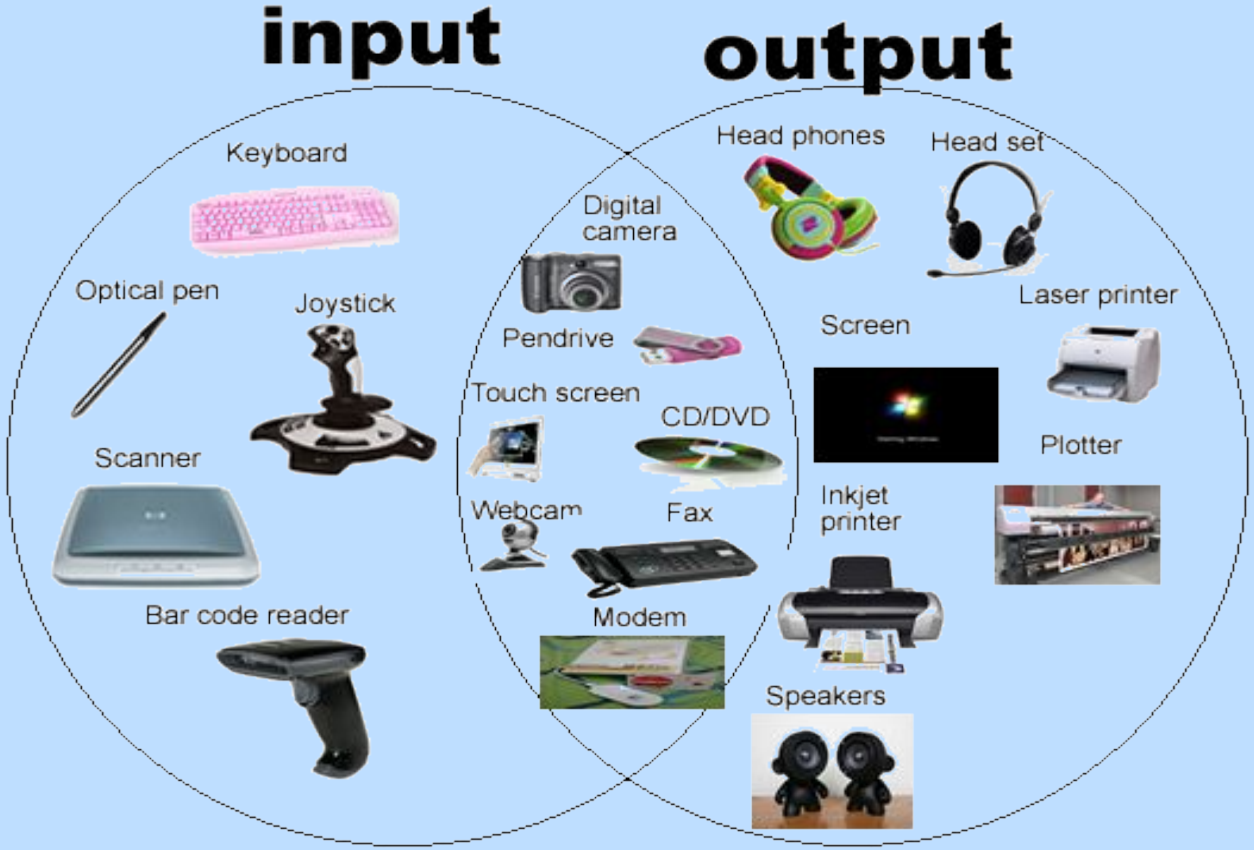
How does a computer work?

4 main parts:

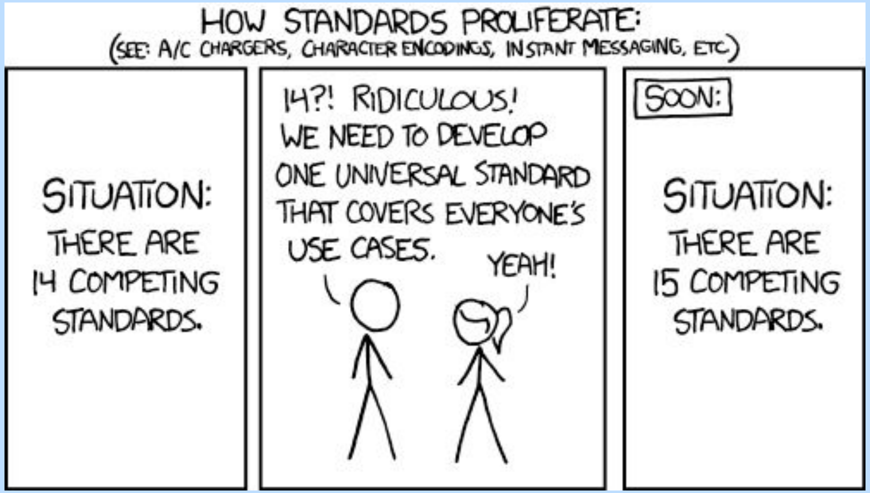
- CPU (Central Processing Unit)
- Memory (cache or RAM)
- Storage (Disks or SSD)
- I/O (Input/Output)



Helix Charter High School
Isabelle Viraldo


input
output
Keyboard
Optical pen
Joystick
Scanner
Bar code reader
Digital camera
Pendrive
Touch screen
CD/DVD
Webcam
Fax
Modem
Head phones
Head set
Laser printer
Screen
Plotter
Inkjet printer
Speakers

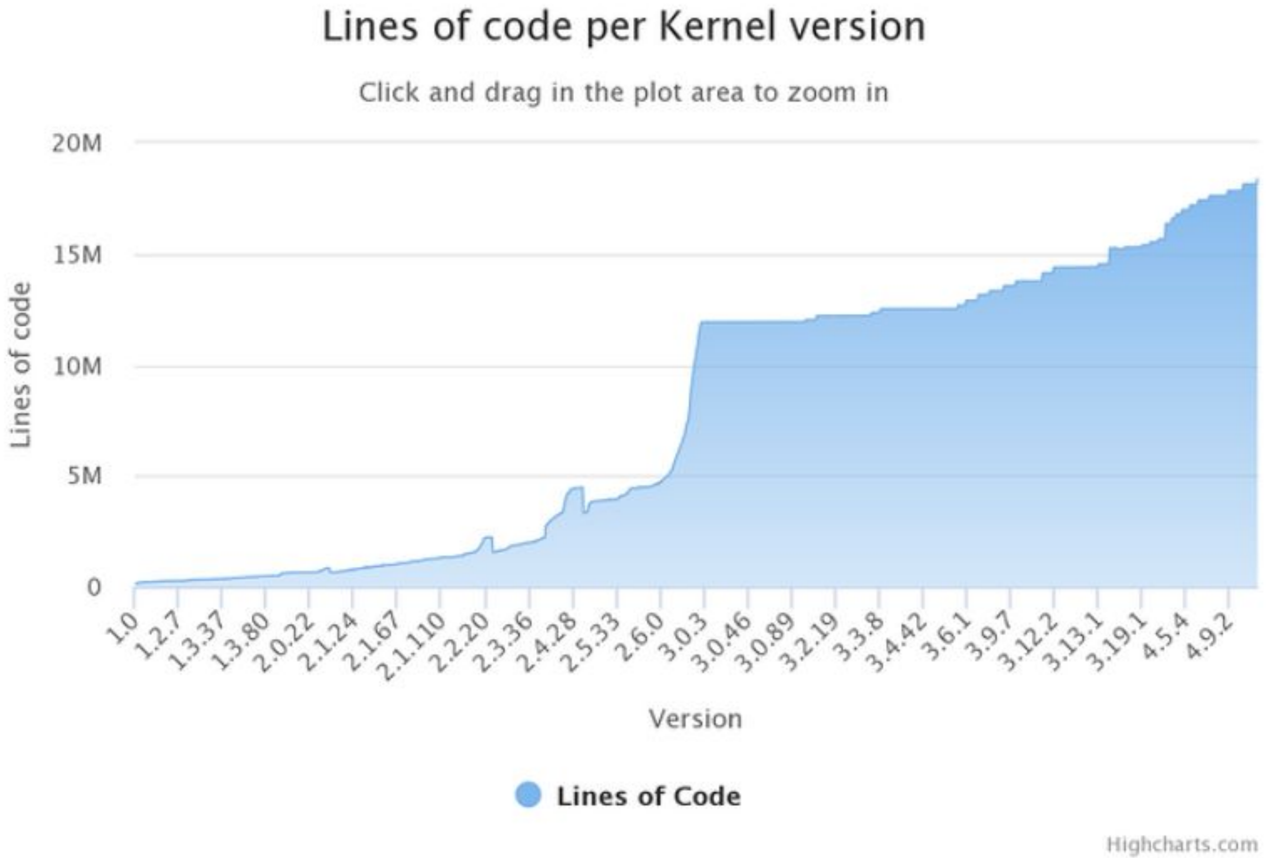
Week 14


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Linux Code Size

As of kernel version 4.6 (in lines)

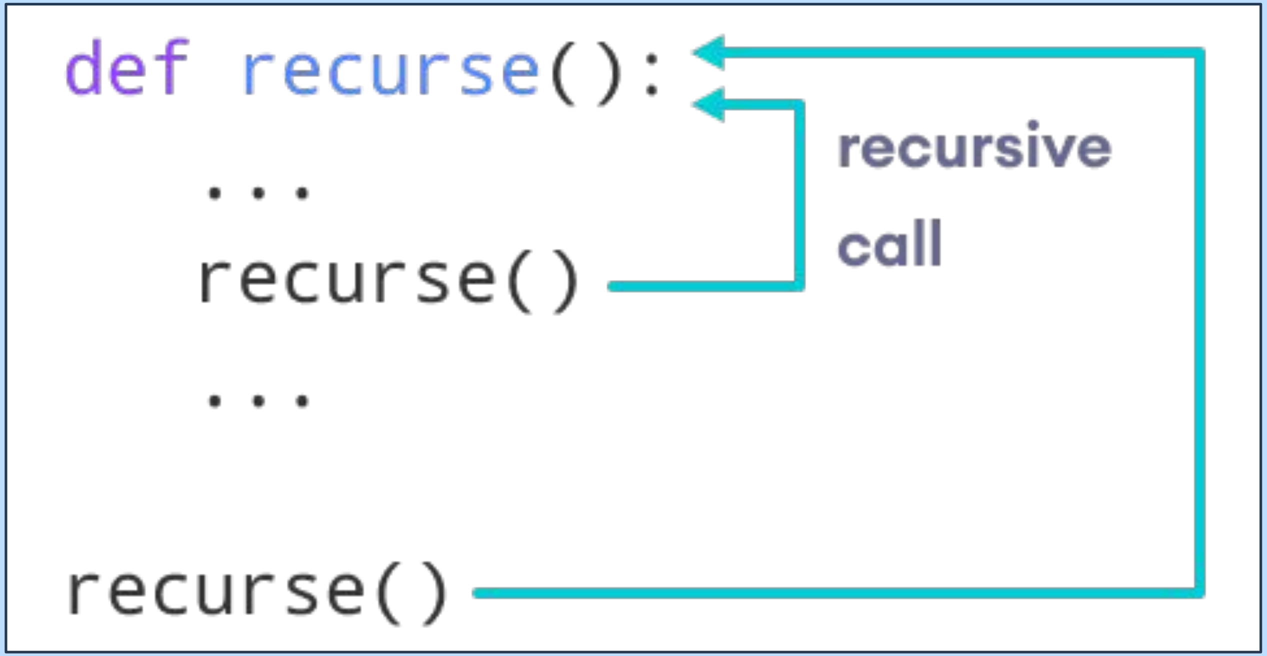
- **drivers/: 57.0%**
- **arch/: 16.3%**
- fs/: 5.5%
- sound/: 4.4%
- net/: 4.3%
- include/: 3.5%
- Documentation/: 2.8%
- tools/: 1.3%
- kernel/: 1.2%
- firmware/: 0.6%
- lib/: 0.5%
- mm/: 0.5%
- scripts/: 0.4%
- crypto/: 0.4%
- security/: 0.3%
- block/: 0.1%



Week 14

Recursion

- A function that calls itself



```
def recurse():
    ...
    recurse()
    ...

recurse()
```

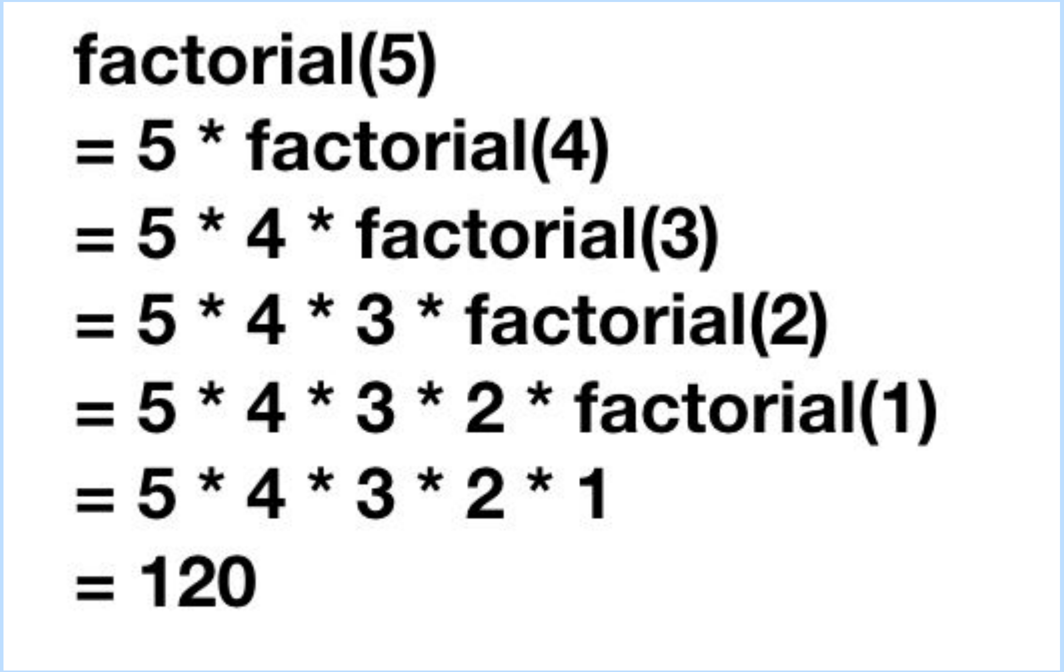
Week 14

Recursion
- A function that calls itself

5! = 5 * 4 * 3 * 2 * 1

5 * 4! = 5 * (4 * 3 * 2 * 1)
...

What if we return the multiple, if the current number is 1



factorial(5)
= 5 * factorial(4)
= 5 * 4 * factorial(3)
= 5 * 4 * 3 * factorial(2)
= 5 * 4 * 3 * 2 * factorial(1)
= 5 * 4 * 3 * 2 * 1
= 120

Week 13

Coding activity! Get out your Chromebooks!

Everyone look up:

python online compiler

Or

Go to: https://tinyurl.com/yc4w9mdh

feedback!

Helix Charter High School
Isabelle Viraldo