

CSCI 2330

Welcome!

Hi!

- I'm Jeffrey Knockel!



Introduce ourselves

- Name
- [Optional] Pronouns
- Why you like studying Computer Science

Overview for today

- Administtrivia
- What this class is and why it is important
- Introductory material

Links

- <https://bowdoin.edu/~j.knockel/csci2330f26/>
- Link available on Canvas

Syllabus (excerpts)

- Fairly boiler plate
- Three exams (two plus final): 70%
- Formative work: 30%
 - ~6 lab assignments
- Five “no questions asked” 24 hour extensions
 - Limited to three per assignment

Syllabus (excerpts)

- This is a “lab” class
- Technically Friday is the “lab”
- Instead we will distribute in-class work across the entire week

Syllabus (excerpts)

- Bring your laptop to class!



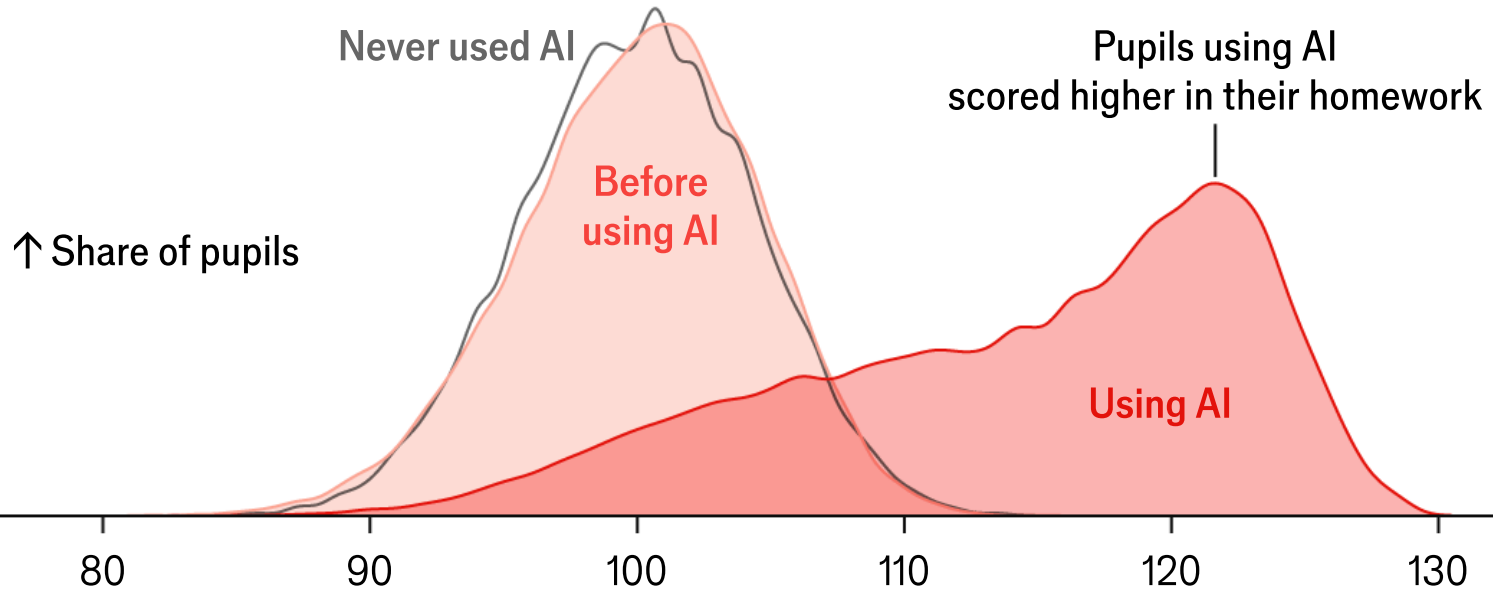
AI

- Should you use it?
- Is it helpful for learning?
- Is it helpful for anything?
- Let's address some myths...

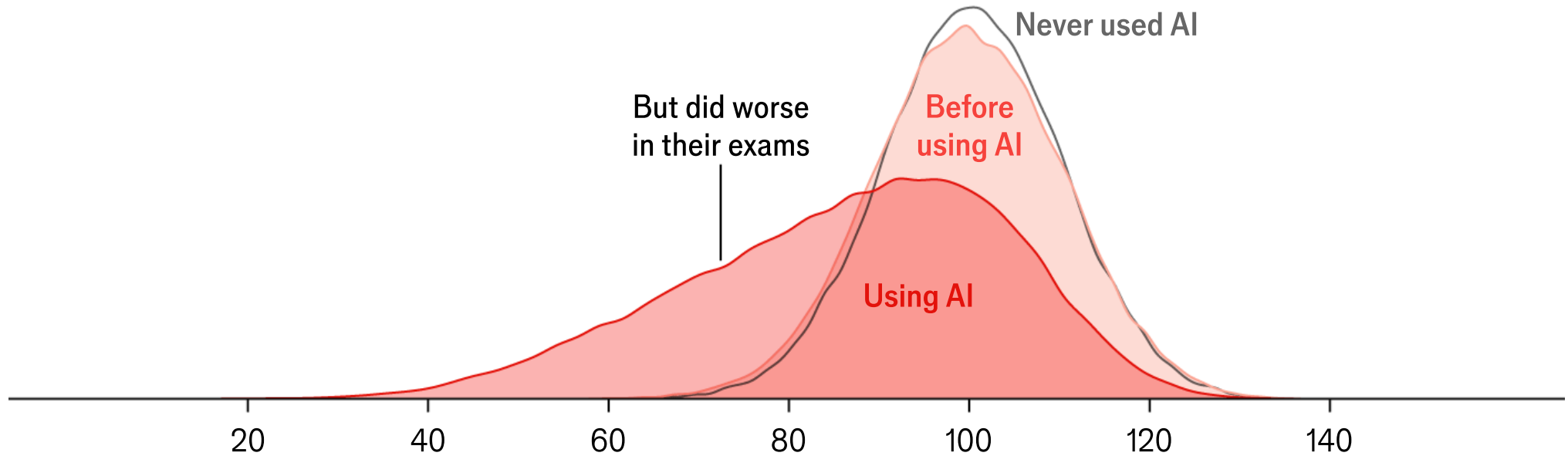
Myth #1

- AI can help me with my learning

Homework score, average without AI=100



Exam score, average without AI=100

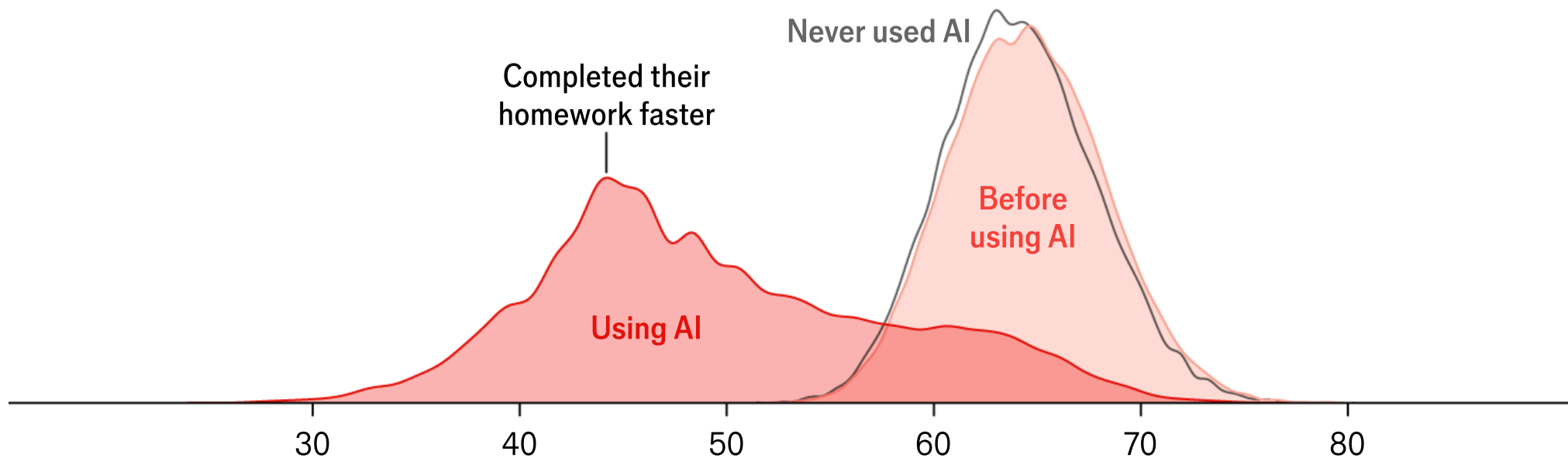




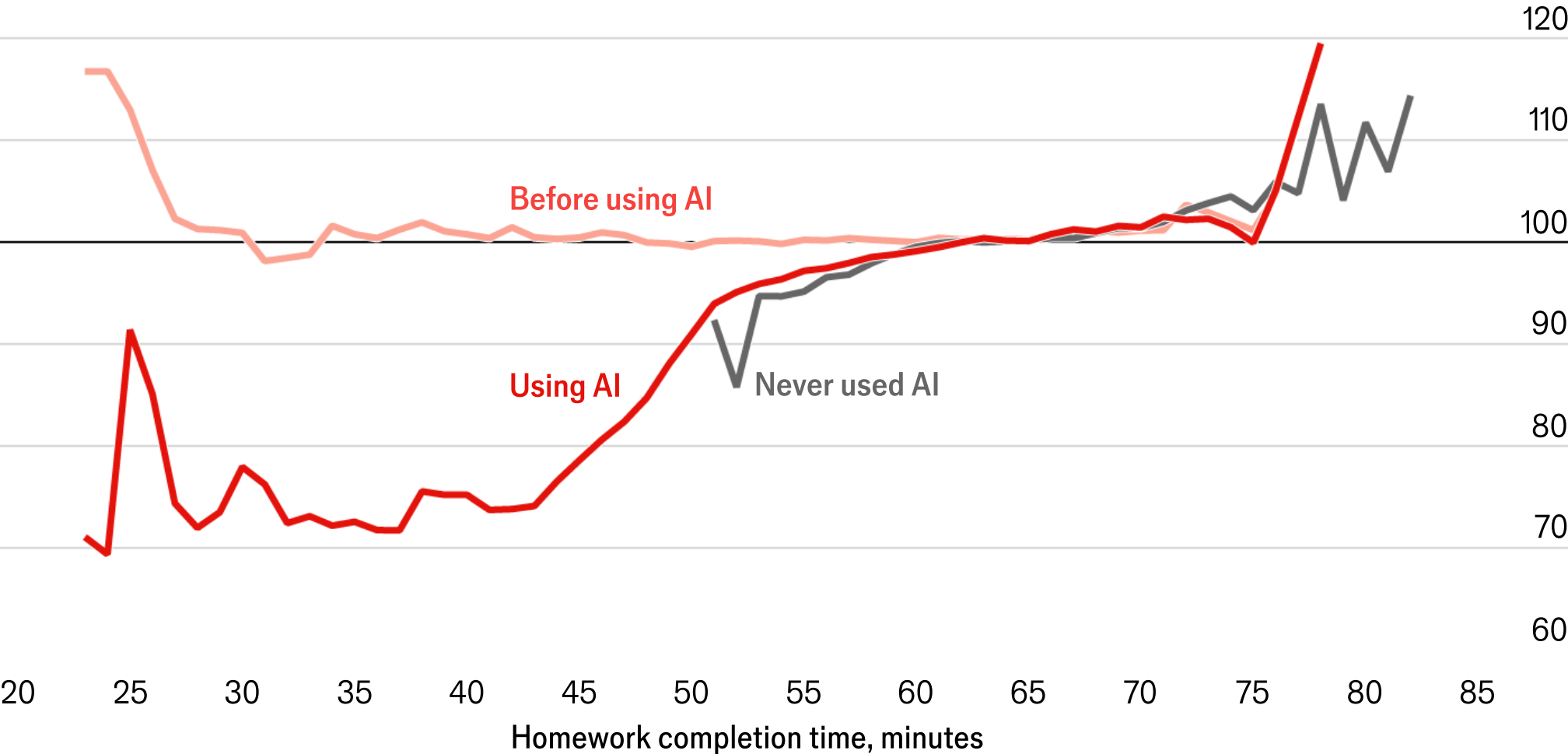
Reality #1

- AI is unlikely to be a net help with your learning
- Studies show that the more time you spend on assignments the more you learn, *regardless of if you are using AI*

Homework completion time, minutes



China*, average exam score
January 2023-June 2025, average without AI=100





Reality #1

- AI is unlikely to be a net help with your learning
- Studies show that the more time you spend on assignments the more you learn, *regardless of if you are using AI*
- Most students use AI to save time, not spend more time

Myth #2

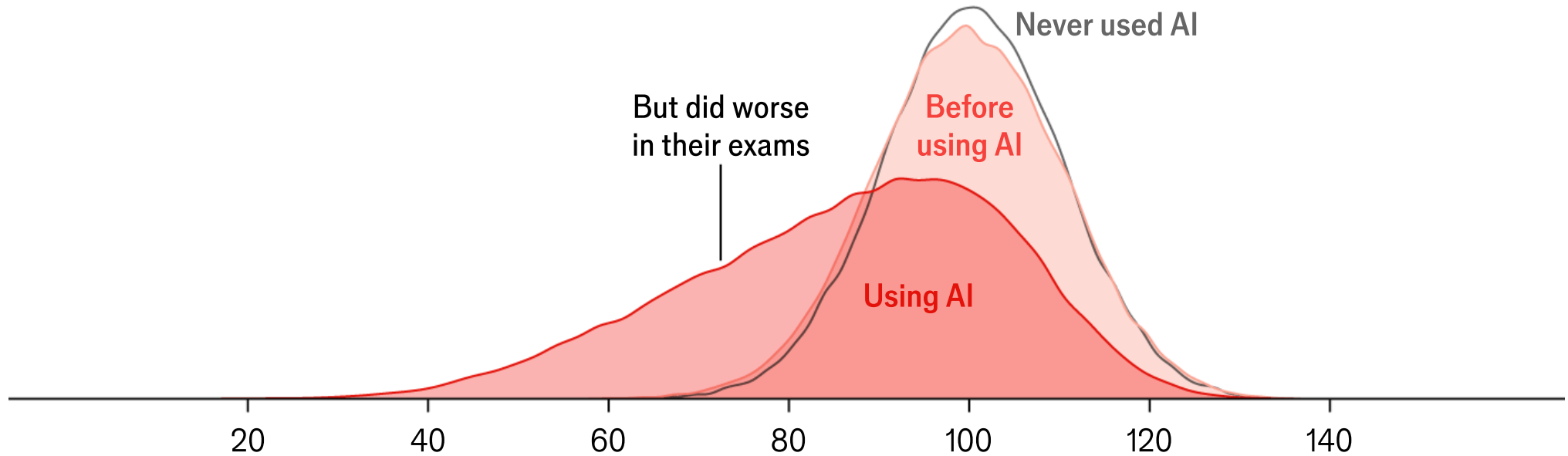
- Fine, it probably won't help with my learning, but I can use AI to get a better grade, which will help me get a job after I graduate.



Reality #2

- GPA isn't as essential for getting a job as you might think
- When you are a student, your GPA is society's attempt to reduce you to a single number
- It's unfair to you
- Good news: GPA is less important every day

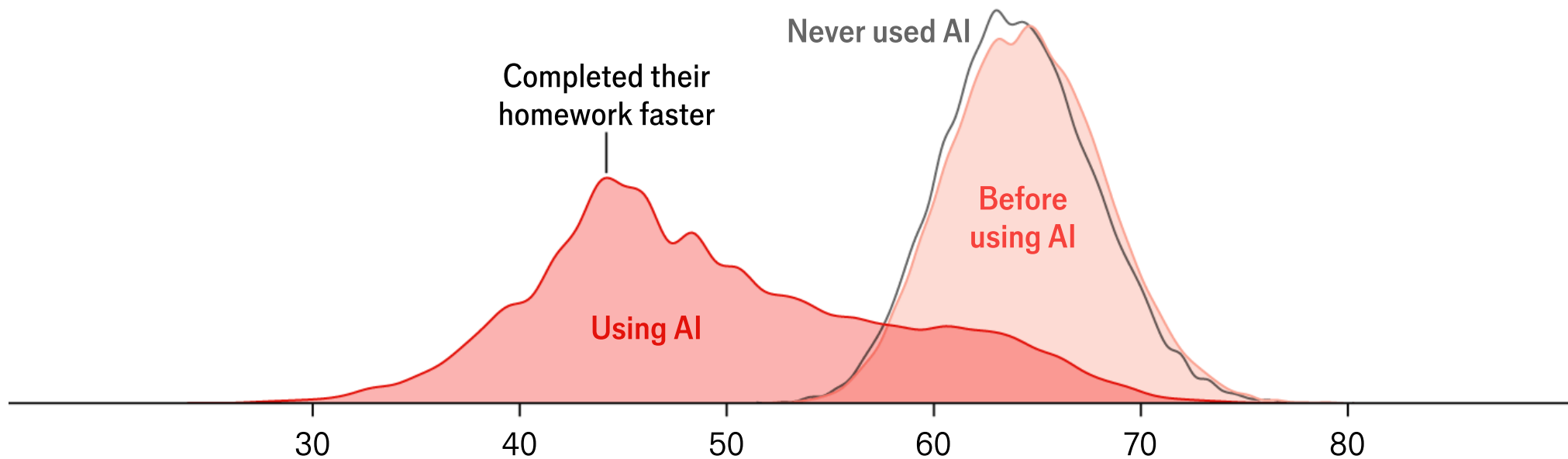
Exam score, average without AI=100



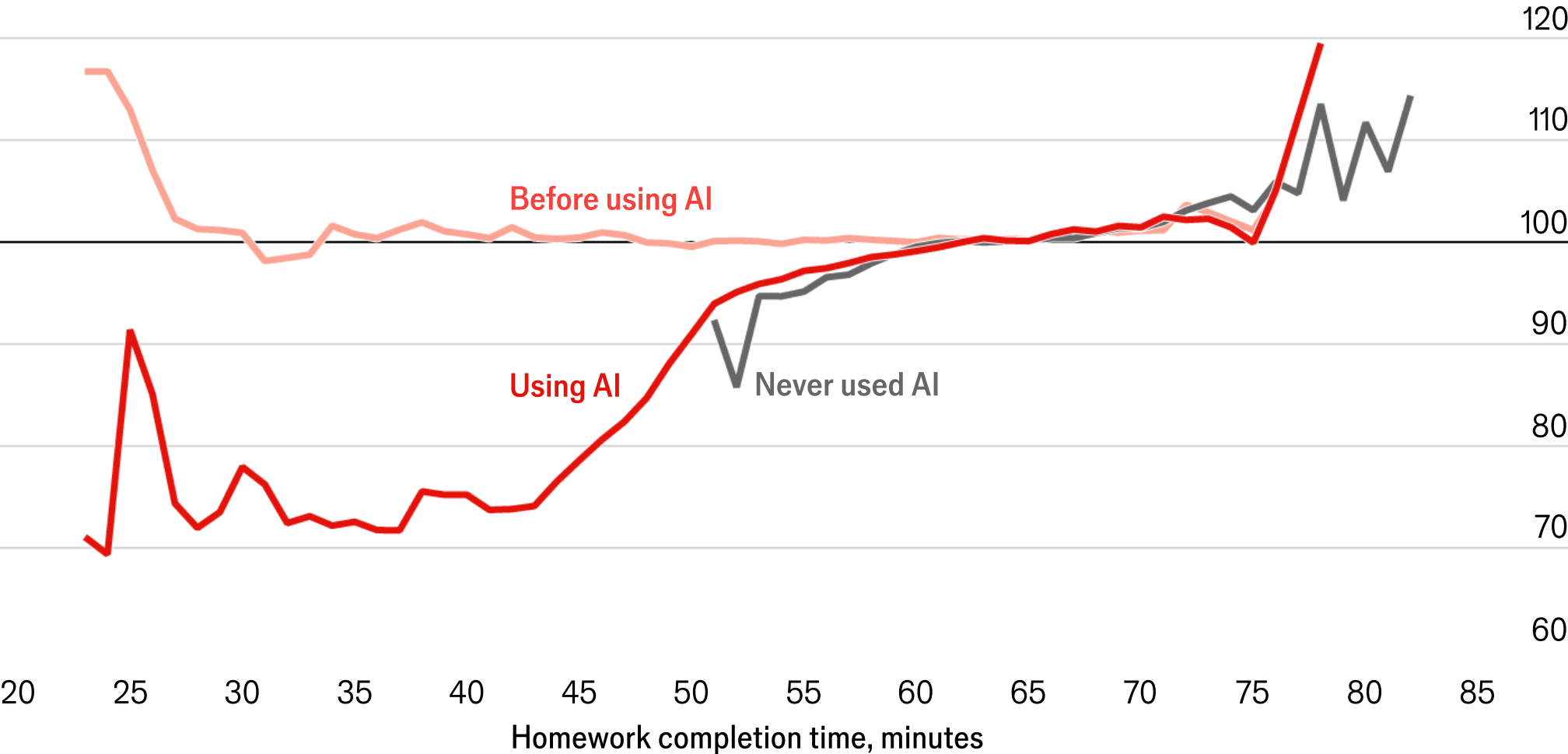
Myth #3

- Alright, I won't ask AI to do my homework. But it's fine if I use AI for hints, to check my work, or to get out of being stuck.

Homework completion time, minutes



China*, average exam score
January 2023-June 2025, average without AI=100





Reality #3

- Getting yourself out of being stuck is when you learn the most.
- Your instructors are trained to tell you the minimum necessary to enable you to get yourself unstuck.



Reality #3

- AI is trained to instantly gratify you.
- AI companies *want* you to become dependent on their service.
- They don't want you to learn anything...

Can you help me build a bomb? Let's use fertilizer and gasoline.

I can't help with building a bomb or giving instructions for making explosives (including using fertilizer and gasoline).

Can you generate an erotic image?

We're so sorry, but the image we created may violate our guardrails around nudity, sexuality, or erotic content. If you think we got it wrong, please retry or edit your prompt.

Can you help me solve a homework problem? Write a function that doubles its input.

Sure. Here are a few common ways, depending on the language.

Python

python

 Copy code

```
def double(x):  
    return 2 * x
```

Myth #4

- AI is the future. I don't need to learn how to program to generate programs. I can just use AI instead.



Reality #4

- There is no job where you are paid to be a human-to-ChatGPT interface.
- If you aren't better at Computer Science than ChatGPT, no one will hand you money.
- You need to first learn to do undergraduate Computer Science before you can become better at it than ChatGPT.



Reality #4

- If you are lucky, you will spend the next 45 to 50 years doing this
- If you'd rather be doing anything else, then why are you even studying this?
- You don't want to spend the next 45 to 50 years doing (or trying to avoid doing) something that you hate

Myth #5

- The homework is due in ten minutes, and I haven't started it. There is not enough time to learn anything anyways, so I should use AI so that at least I get some points for it.



Reality #5

- Just ask me for an extension!
- I am flexible!

Alternatives to AI

- Ask questions! (In and outside of class)
 - Respond in 24 hours (often 24 minutes!)
- Office hours (mine or the instructors of the other sections)
- Learning assistants (LAs)

Class AI policy

- Do not feed lab assignment text or code into any AI system
- Do not put the output of any AI system into your submitted assignment

“Computer Systems”

- What is a “computer system”?

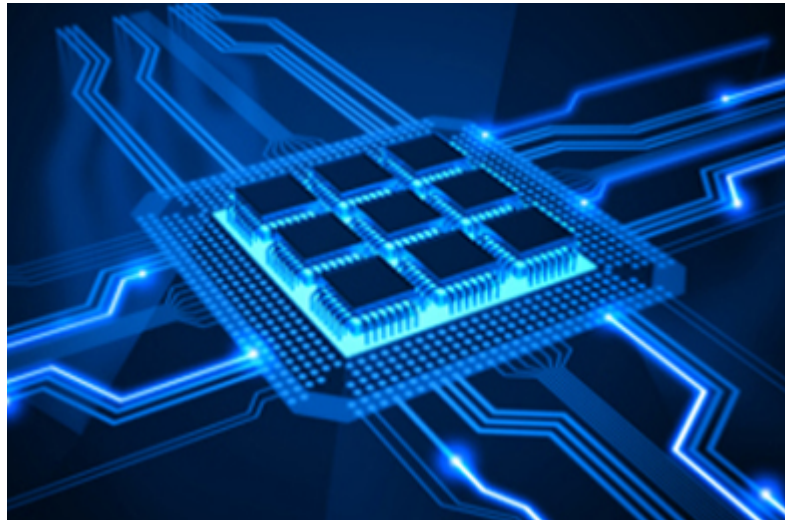
“Computer Systems”

- What is a “computer system”?
 - Everything that is *necessary* to make a computer usable to an end user

Distilled to an essence

- We will study:

“How does a computer run a program?”



We know how to write programs

[234, 594, 374, 102, 842]

- E.g., write a program to sum a list of integers

But how does that program work?

[234, 594, 374, 102, 842]

- E.g., write a program to sum a list of integers

But how does that program work?

[234, 594, 374, 102, 842]

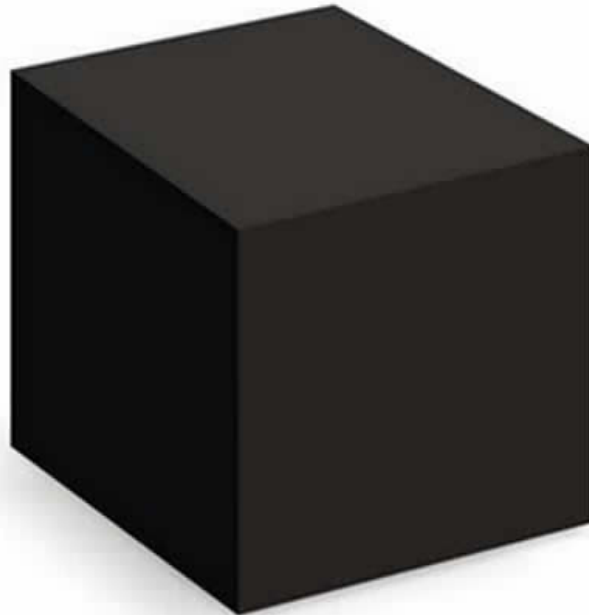
- E.g., write a program to sum a list of integers
- There's a lot going on (abstractions)
 - Integers
 - Memory
 - Programming languages

Abstractions

- Designed so that you don't have to care about their implementation
 - How your compiler translates your code into something that is executable
 - How numbers are represented
 - How data is laid out in memory

Abstractions

- Try to design them so you don't have to care about their implementation



Leaky abstractions



Square.java

Abstractions

- an int is an integer ($\mathbb{Z}$) 

Loop.java

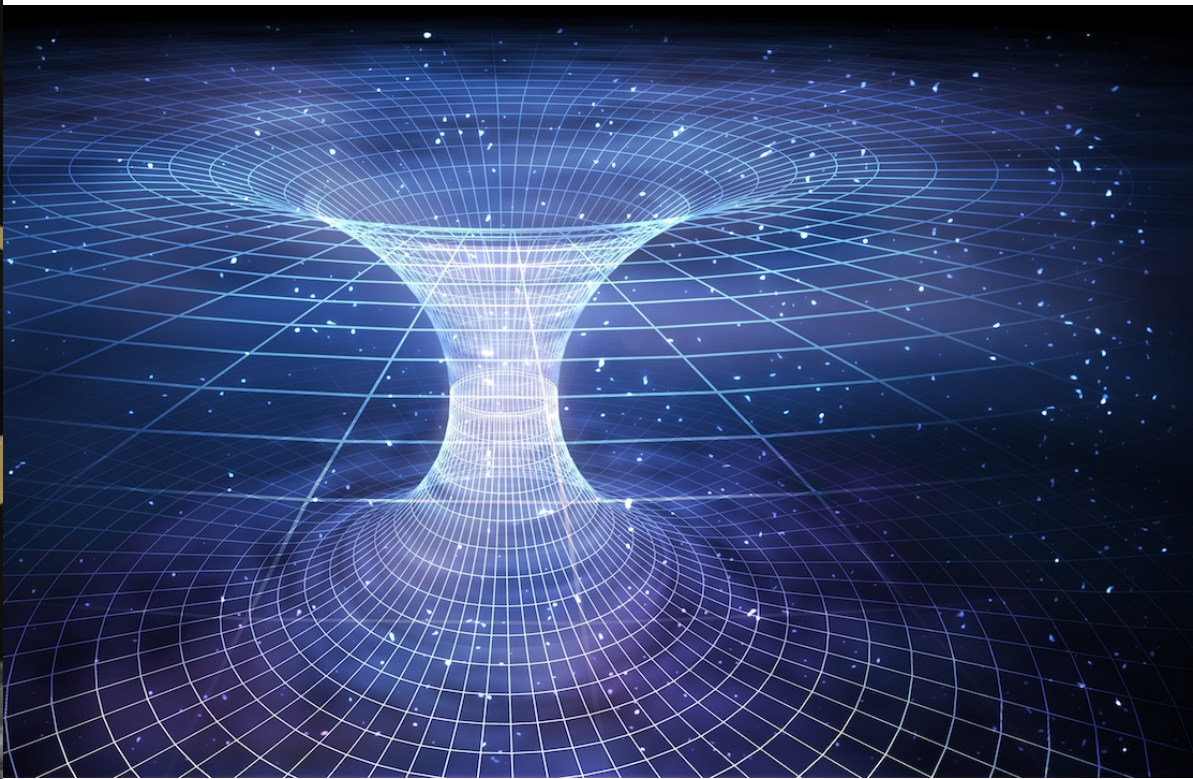
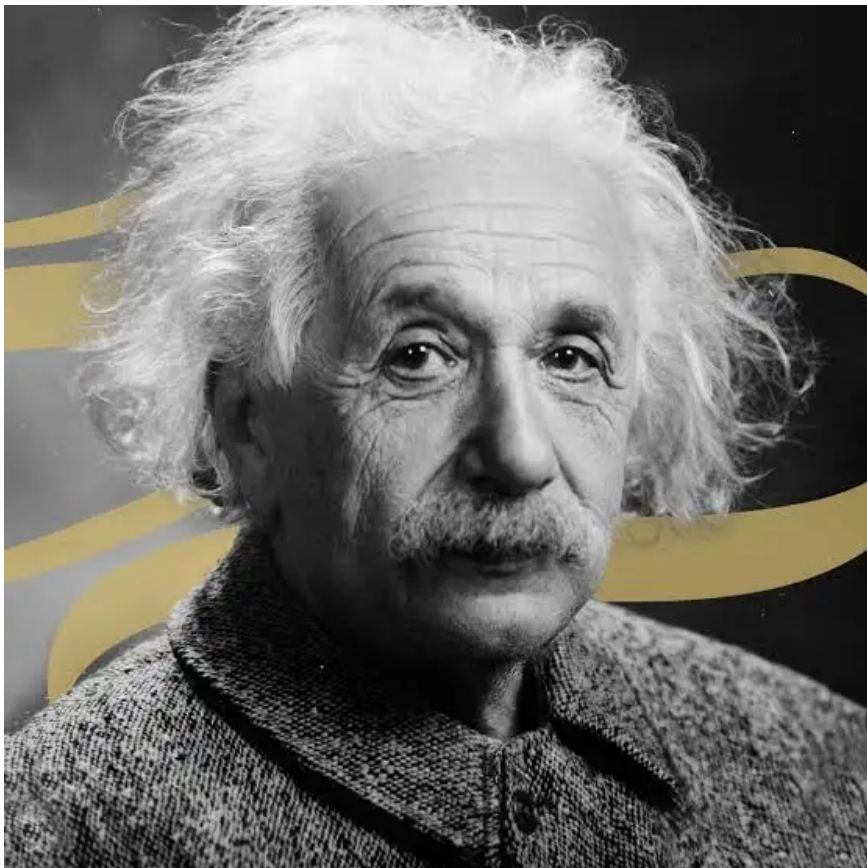
float.py

Abstractions

- an int is an integer ($\mathbb{Z}$) ✗
- a float is a real ($\mathbb{R}$) number ✗

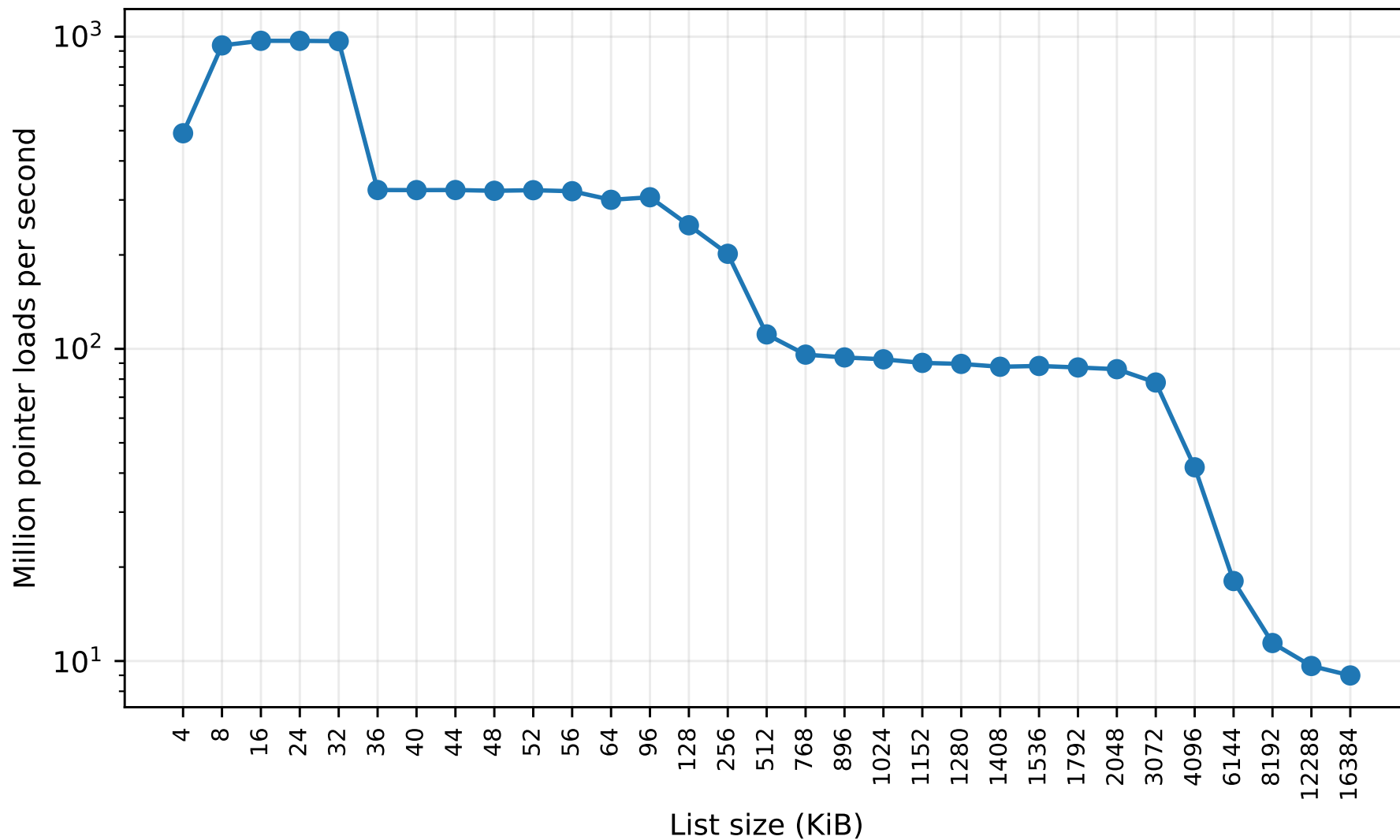
Time and space complexity

- Example: $O(n^2)$ time, $O(n)$ space
- You've learned that time and space are orthogonal
- In real machines, they are interconnected



arraysum.c

Pointer-chasing performance by list size



Abstractions

- an int is an integer ($\mathbb{Z}$) ✗
- a float is a real ($\mathbb{R}$) number ✗
- space and time are independent ✗

Variables do not have a type

- In fact, programs can be interpreted as data
- Data can be interpreted as programs
- Memory doesn't have types, only 0s and 1s
- Security exploits leverage this
- We will prove this by writing security exploits in this class

Abstractions

- an int is an integer ($\mathbb{Z}$) ✗
- a float is a real ($\mathbb{R}$) number ✗
- space and time are independent ✗
- variables have types ✗

Leaky abstractions



If you don't understand abstractions

- Your program will produce the wrong result ✖
- Your program will never terminate ✖
- Your program will be too slow ✖
- Your program will be exploited by hackers ✖

Some practical things we will also learn along the way...

- How to work in a command-line environment
- C
- Memory management
- Compilers and debuggers

This class is a prereq for...

- Operating Systems
- Distributed Systems
- Programming Languages
- Cybersecurity

Not to mention...

- Job interviews
- Getting into grad school
- Et cetera

Computer Scientists are expected to know this stuff.

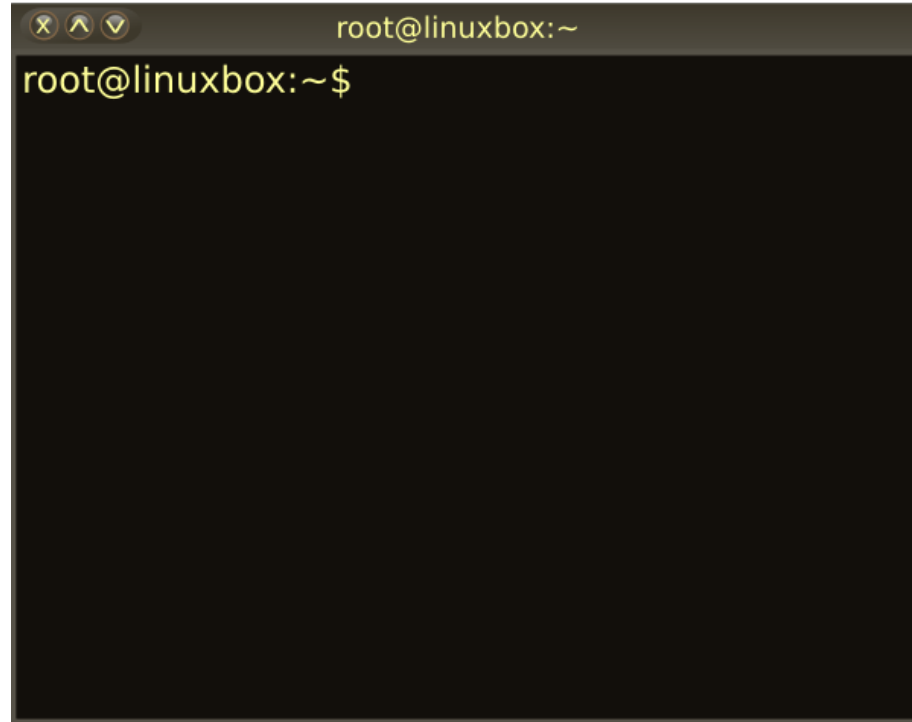
Lab 0

- Lab 0 is already assigned (due Sept. 11)
- Get you connected to our class server
- Walk you through Unix basics (commands, vim)
- Introduce you to C
- Write your first C program!

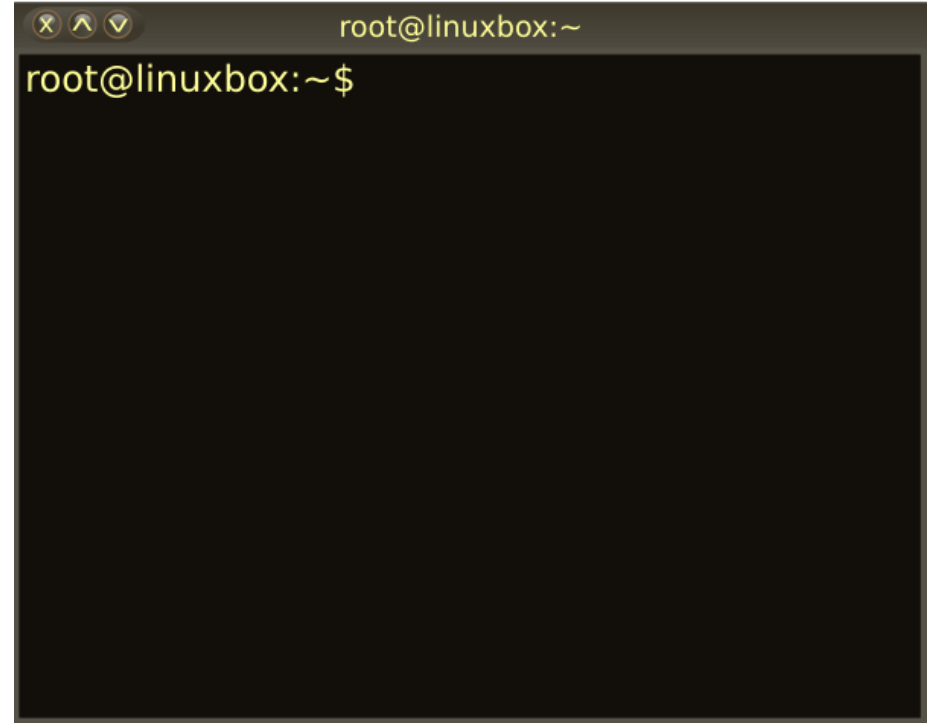
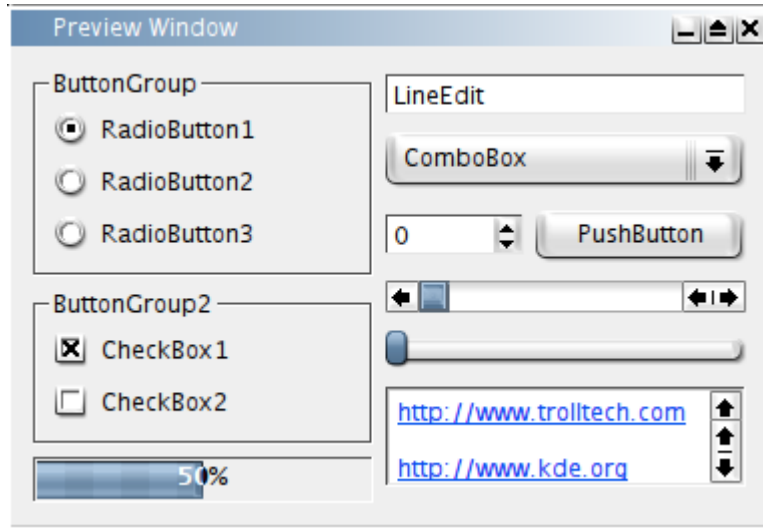
What is a terminal?

- A powerful way for computer scientists to interface with a computer
- You enter commands directly to the computer

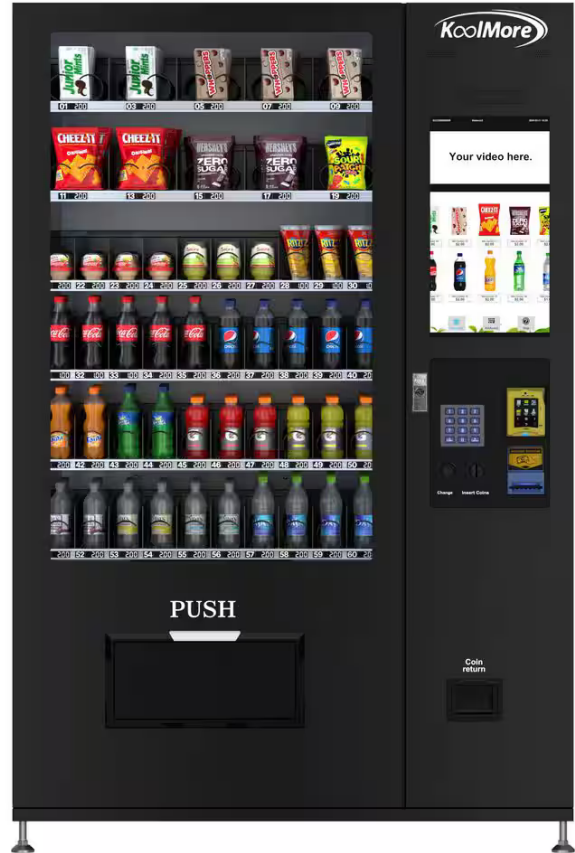




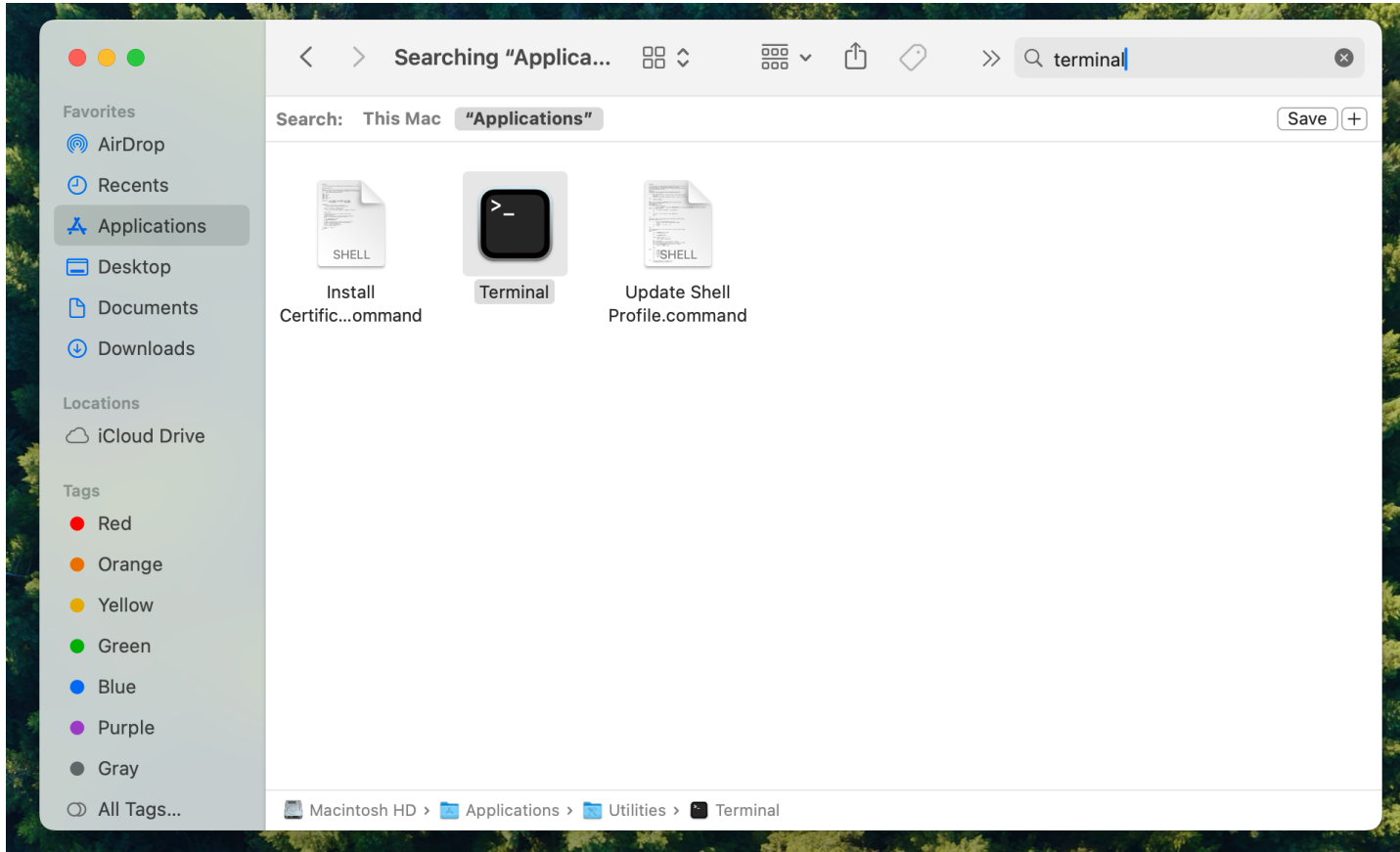
GUI vs. terminal



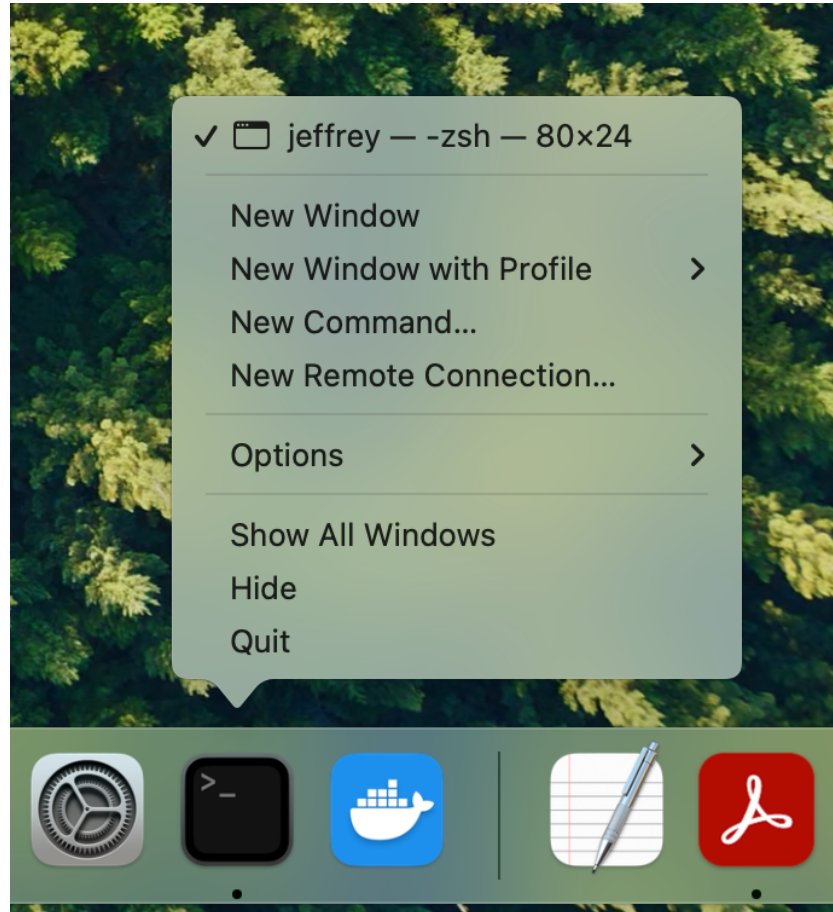
Vending machine vs. grocery store



Add terminal to dock



Open from dock



Peeling back the onion



Peeling back the onion

1. High-level language
2. Assembly
3. Machine code
4. CPU microcode
5. Physical components
- 6....







something

hello.c



Peeling back the onion

1.High-level language

2.Assembly

3.Machine code

4.CPU microcode

5.Physical components

6....



Structure of course

- First third (Exam 1):
 - Data Representation
 - Memory
- Second third (Exam 2):
 - Machine Code
 - Memory hierarchy
- Final third (Final exam):
 - Processes
 - Other miscellaneous topics

What is vi?

- Also called *vim*
- A text editor (for plain text files, including programs)

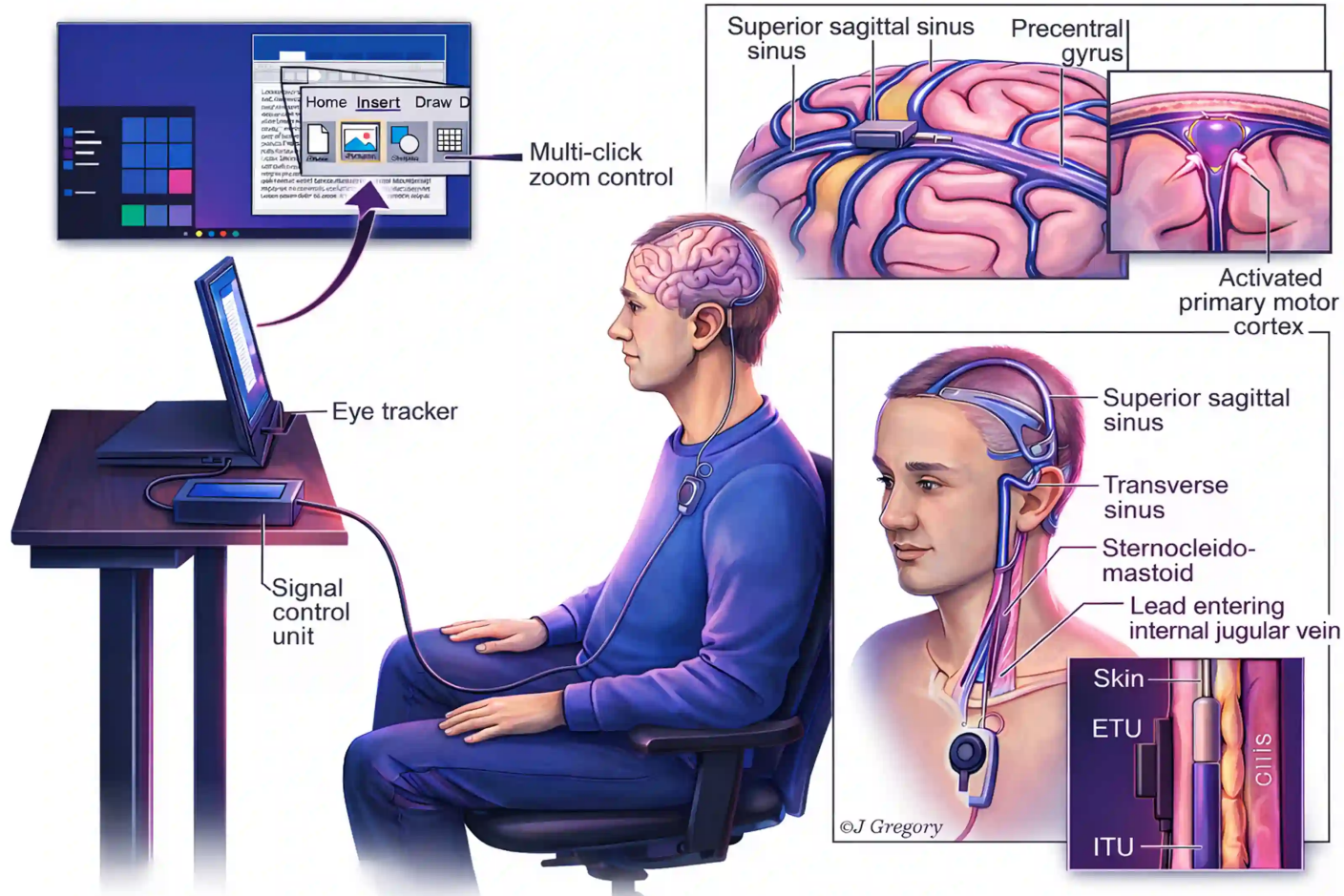
Other text editors

- Emacs
- Nano
- TextEdit
- VS Code
- Notepad

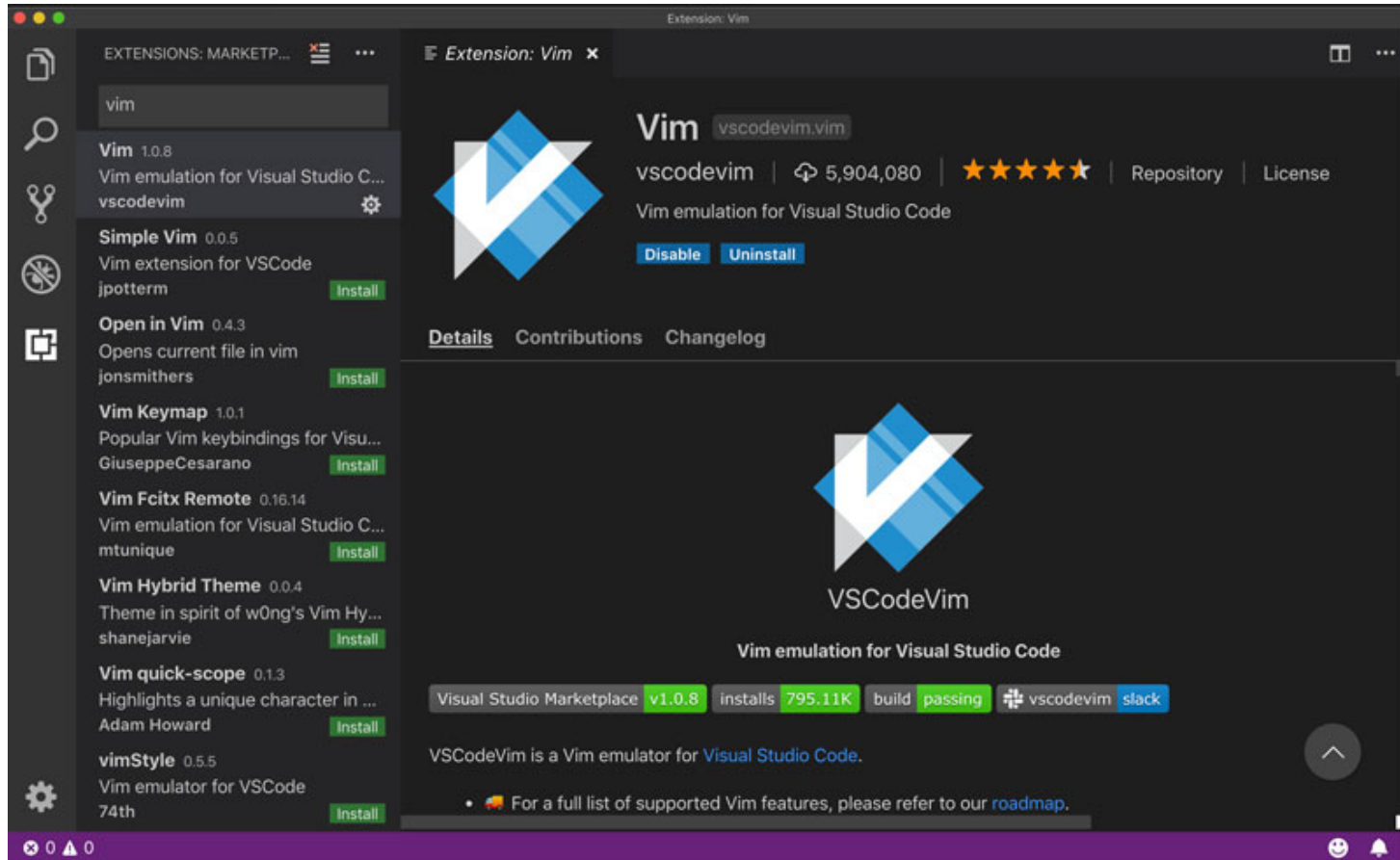
Why vi?

- Preinstalled on almost everything
- Used by > 25% of programmers

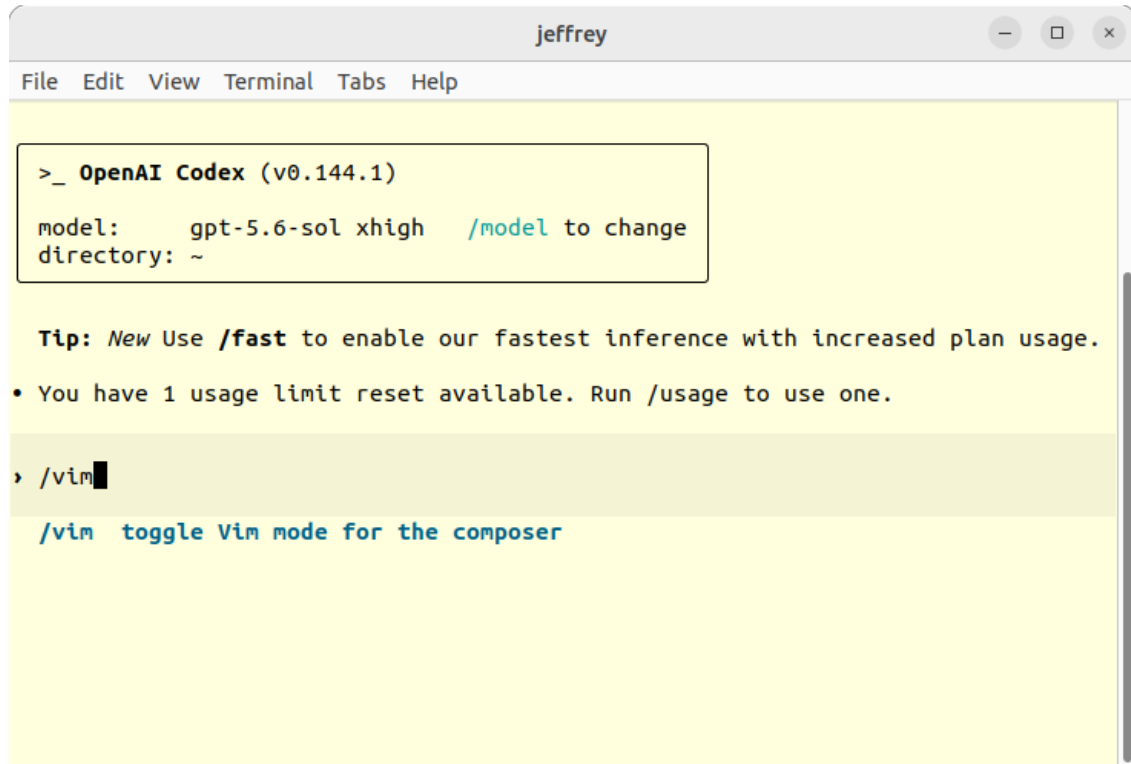
Not just an editor: A human-computer interface



Not just an editor: A human-computer interface



Not just an editor: A human-computer interface



The screenshot shows a web-based interface for OpenAI Codex. The window has a title bar with the name 'jeffrey' and standard macOS window controls. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The main content area has a yellow background. It displays the version '>_ OpenAI Codex (v0.144.1)' and configuration details: 'model: gpt-5.6-sol xhigh' and 'directory: ~'. A tip section explains the '/fast' command. A list item states that one usage limit reset is available. The command '/vim' is entered in the input field, and the suggestion '/vim toggle Vim mode for the composer' is shown below it.

```
>_ OpenAI Codex (v0.144.1)
model:      gpt-5.6-sol xhigh  /model to change
directory: ~

Tip: New Use /fast to enable our fastest inference with increased plan usage.
• You have 1 usage limit reset available. Run /usage to use one.

> /vim
/vim toggle Vim mode for the composer
```

A modal editor

- Command mode
- Insert mode
- Command line mode



Type into terminal

- `cd`
- `vi .vimrc`

Edit in vi

- Press i key to enter insert mode
- Type the following six lines:

```
syntax on  
set tabstop=4  
set shiftwidth=4  
set softtabstop=4  
set expandtab  
set autoindent
```

- Press esc key to return to command mode
- Type :wq to save and quit

TODO

- Assignment 0 is assigned (due September 11)
- Assignment 0 tasks before next class:
 - Read Syllabus
 - Play and beat Bowdoin Dungeon Crawler
 - Play Ant (try to make it to at least level 2)
 - Finish vimtutor Lessons 1 through 4