



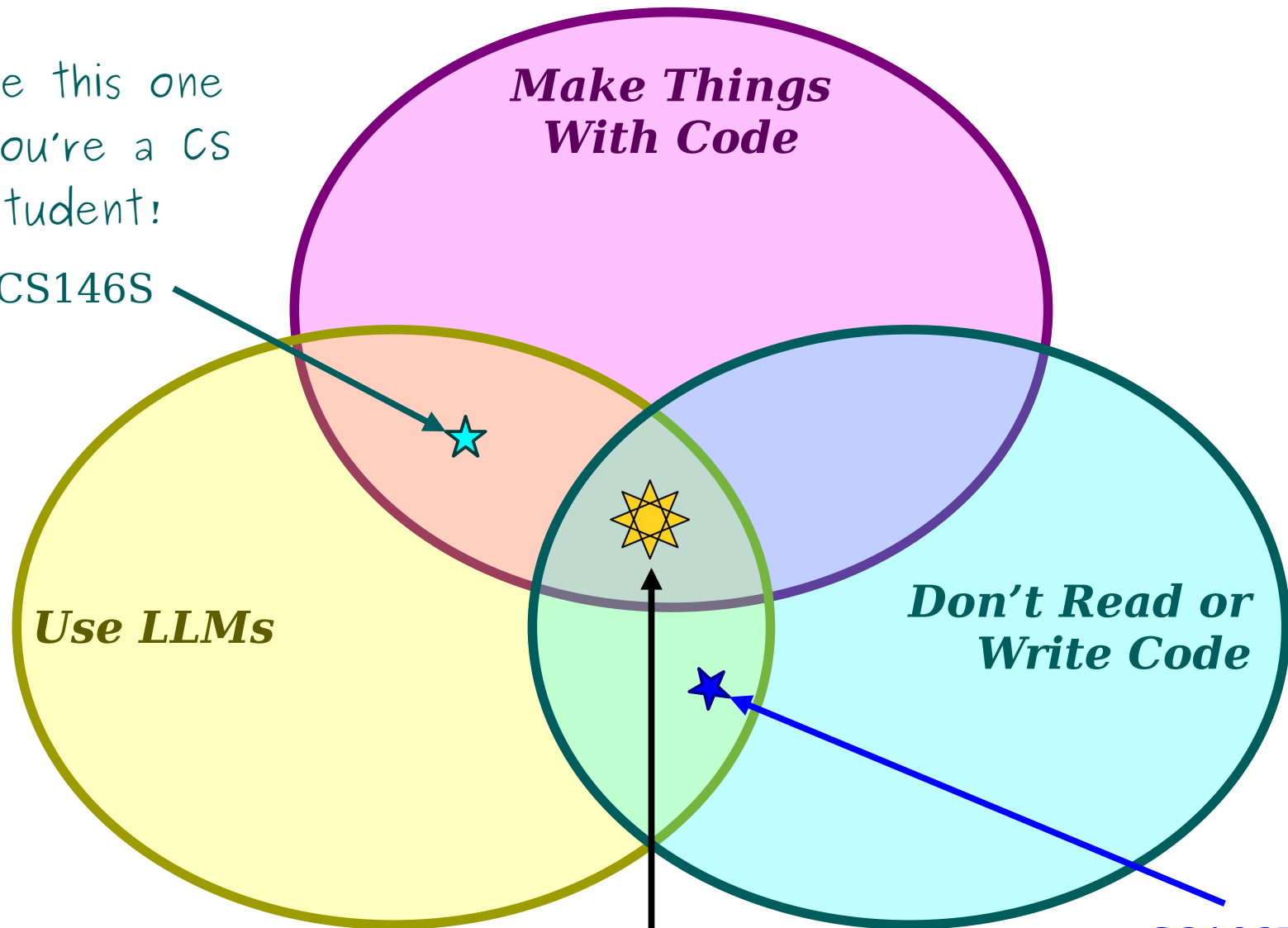
CS193V: Effective Vibecoding

Welcome!

What is this course?

Take this one
if you're a CS
student!

CS146S



**CS193V
(This Course!)**

CS193T

Take this one
for AI outside
of coding!

Goal: ***Learn to make things out of code,
without writing that code yourself.***

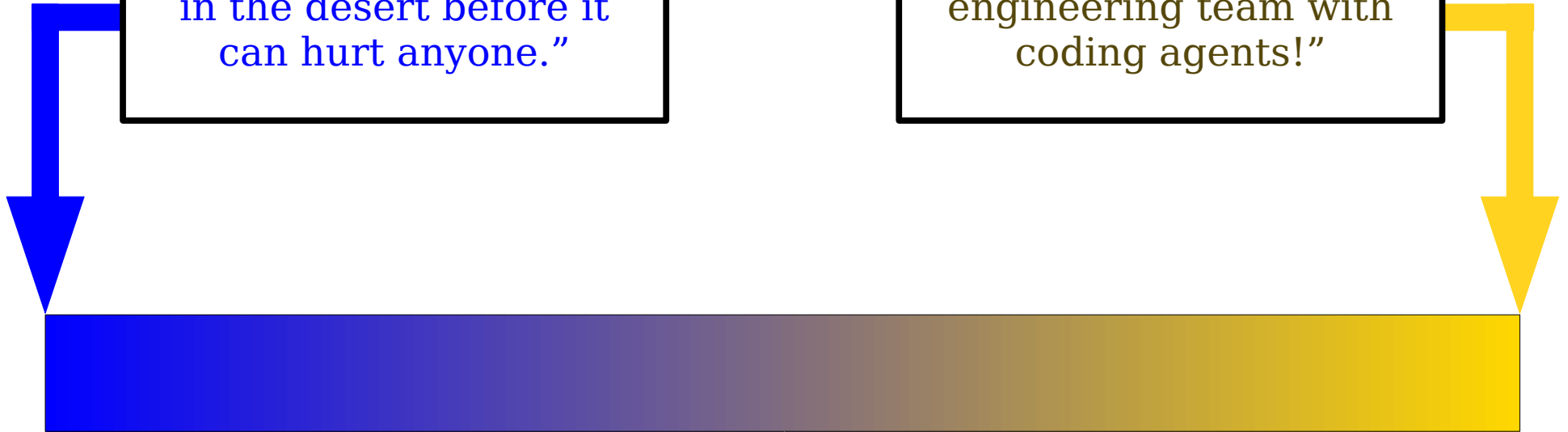
Let's refine this.

What software can you make with
AI without reading or writing
any code?

“AI code is so unreliable
that we should bury it
in the desert before it
can hurt anyone.”

“AI is so good at coding,
I just replaced the entire
engineering team with
coding agents!”

Where we actually
are right now.



Where We Are Now

- AI tools are great at vibecoding small- to medium-sized projects.
 - Modern coding agents can let you do in a week what would have previously taken years of technical training.
- AI is not (yet) ready to build large-scale software on its own, and even if it could it wouldn't be safe to deploy.
 - Companies laying off engineers are starting to have real "buyer's remorse."
 - Deploying software you haven't audited – especially if you don't understand the risks – can lead to harm.
- ***This Course:*** Learn to build small- to medium-sized projects.

You'll Learn How To...

- ... write data analysis scripts for your research lab.
- ... build working prototypes of larger software projects.
- ... create automated tools to audit government / corporate data in real-time.
- ... make fun personal apps designed to work well for you or your friends.
- ... publish a custom website for yourself, a group you're in.
- ... *(etc.)*

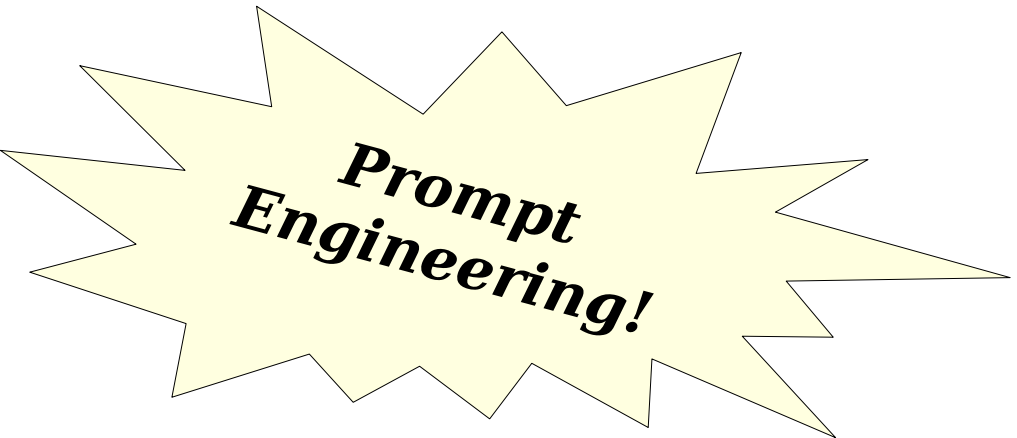
Out of Scope

- Build production websites with 100+ users.
 - But a *working prototype* is in scope.
- Start a software company without engineers.
 - The technology isn't there yet.
 - The risks are *real* and *externalized*!
- Augment existing coding skills using AI.
 - If you have well-developed coding skills, ***take CS146S instead of this course***. You will learn *way* more.

What do you need to learn to
vibecode effectively?



**Multiagent
Workflows!**



**Prompt
Engineering!**



MCP!



RAG!



**Context
Management!**

Our Core Thesis

The best way to get AI to write good code for you is to *have a clear sense of what you want to build* and *use established software engineering practices.*

Elements of Design

How do I turn ideas into concrete plans?

How will people interact with my software?

How do I make it welcoming and intuitive?

What workflows should I optimize for?

Software Engineering

How do we ensure the code we write works?

How do we collaborate with others when writing code?

When things don't work, how do we fix them?

How do we decide what we should make?

How do we assess risks when deploying software systems?

To Summarize

Course Goals

- *Learn to make things out of code, without writing that code yourself.*
- To that end:
 - Explore foundational concepts from software engineering.
 - Learn to introspect on your ideas and flesh them out before the AI takes over.
 - Understand the limits of what AI tools can make and the risks of deploying AI-written code.

Week 1	Technical Foundations <i>terminal usage, version control, software testing, ...</i>
Week 2	
Week 3	Knowing What to Build <i>plan mode, user studies, ...</i>
Week 4	Data Analysis and Visualization <i>data file formats, API keys, data security, ...</i>
Week 5	Web Applications <i>client/server, deploying websites, testing, prototyping, ...</i>
Week 6	
Week 7	Working in Teams <i>issue tracking, continuous integration, ...</i>
Week 8	Privacy and Security <i>zero-days, prompt injection, the lethal trifecta...</i>
Week 9	Mobile Development <i>remote debugging, emulators, ...</i>
Week 10	Wrap-Up / Special Topics <i>tbd</i>

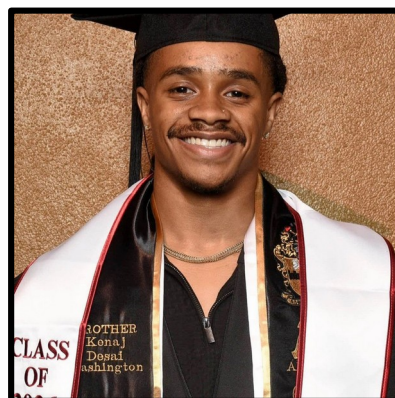
Course Logistics



Keith Schwarz
htiek@cs.stanford.edu
(Instructor)



Mara Bravo-Santos
(TA)



Kenaj Washington
(TA)



Shirley Yu
(TA)

The Course Website

<https://cs193v.stanford.edu>

Everything for the course
will be on this website! We
won't use our course Canvas
for anything.

This Course Is New!

- This is a brand-new course! Your feedback is essential for making it run smoothly.
 - Point out typos, grammatical errors, formatting issues, unclear instructions, technical problems, calibration issues, topics you'd like to learn more about, etc.
 - Feel free to email me (htiek@cs.stanford.edu) with suggestions and feedback.
 - Over the course of the quarter, you'll get a few emails from **High-Resolution Course Feedback (HRCF)** asking for your input.
- Treat this as a collaborative endeavor where we're all going to learn a ton together.

Course Components

- **20% Homework**
 - Lots of little assignments, one per lecture.
 - Fairly open-ended; all exercise specific technical concepts.
 - Graded on completion, not correctness.
 - You'll show off your creations to your peers at the start of each lecture. Expect lots of *ooh's* and *whoa!'s*. 😊
- **40% Tutorial Sessions**
 - Meet with a TA in a small group and do a task together.
 - Graded based on participation and demonstrated knowledge.
 - Come prepared and have fun doing cool things.
 - Details later this week.
- **40% Final Project**
 - Work in a small group to build something of your own choosing.
 - Check in a few times with the course staff, then present what you created.
 - Details later this quarter.

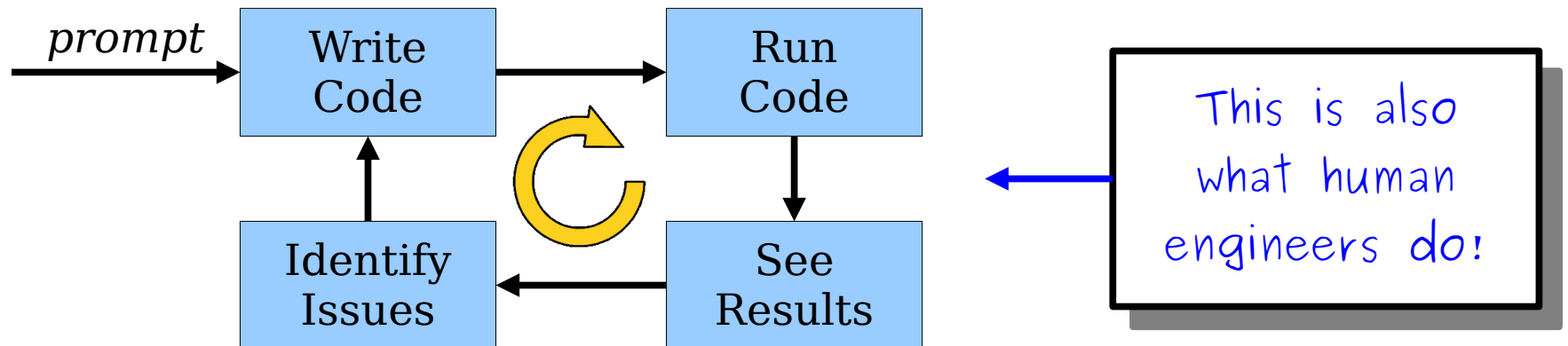
Assignment 0

- Assignment 0 goes out today. It's due at ***1PM Thursday***.
 - Sign up for Stanford's licenses for Anthropic and OpenAI.
 - Register for GitHub, which you'll use to submit work.
 - Tell us more about yourself so we can tailor the course to you!
- Full assignment instructions are on the course website under the "Assignments" tab.

Getting Started:
How do we vibecode something?

Coding Agents

- You're probably used to interacting with AI by going to a website and typing something in.
 - You can *technically* vibecode this way, but I don't recommend it. You'll see why in a minute.
- Instead, we use a tool called a ***coding agent***.
 - The term “agent” means too many different things. If a term in CS only has one meaning, someone will come up with a new meaning for it. 😊
- A coding agent lets an LLM run commands on your computer.



Coding Agents

- We'll use two coding agents this quarter.
 - **Claude Code** (Anthropic)
 - **OpenAI Codex** (OpenAI)
- We're using these because Stanford has site licenses with these companies.
- **My Claim:** Once you learn how to use any coding agent, you can very easily pick up how to use another coding agent.
 - Antigravity, Cursor, OpenCode, ...
- There are other tools for vibecoding specific types of programs (e.g. Replit), but we won't cover them in this course.

**“Claude Code can build *anything*”
- Sierra Wang**

Demo: ***Using Claude Code***

Create a website that simulates a rectangular puddle of water with raindrops falling onto it.

Generate music in real-time whenever a raindrop hits the water and whenever two ripples touch one another.

Next Time

- ***Using the Terminal***
 - Controlling your computer with plain text, the old-fashioned way. 😊
- ***Running Coding Agents Safely***
 - LLMs are frequently wrong, and that's a bad thing when they control your machine!
- ***VMs and Containers***
 - Isolating programs on your computer.