

Early observations

15-113 – Monday 8/31

Reminders!

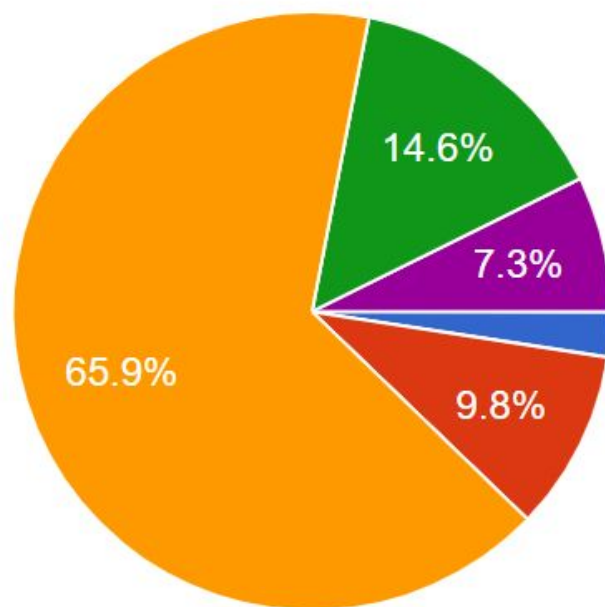
- If you haven't done the HW1 survey, please do that soon. If you joined the class within the last 3 days, send me an email and we'll make a catch-up plan.
- Go fill out the availability form right away:
<https://edstem.org/us/courses/102937/discussion/8209404>
 - This will be a checkin where you show your project/hw prototype to a TA for feedback
- Project 1 (your website) is due Sat 9/5 at 8pm!
- More information to come about pre-deadline TA meetings

HW1 survey results

How would you describe your ability to create working applications with that language when you *do not* have access to generative AI?

 [Copy chart](#)

41 responses



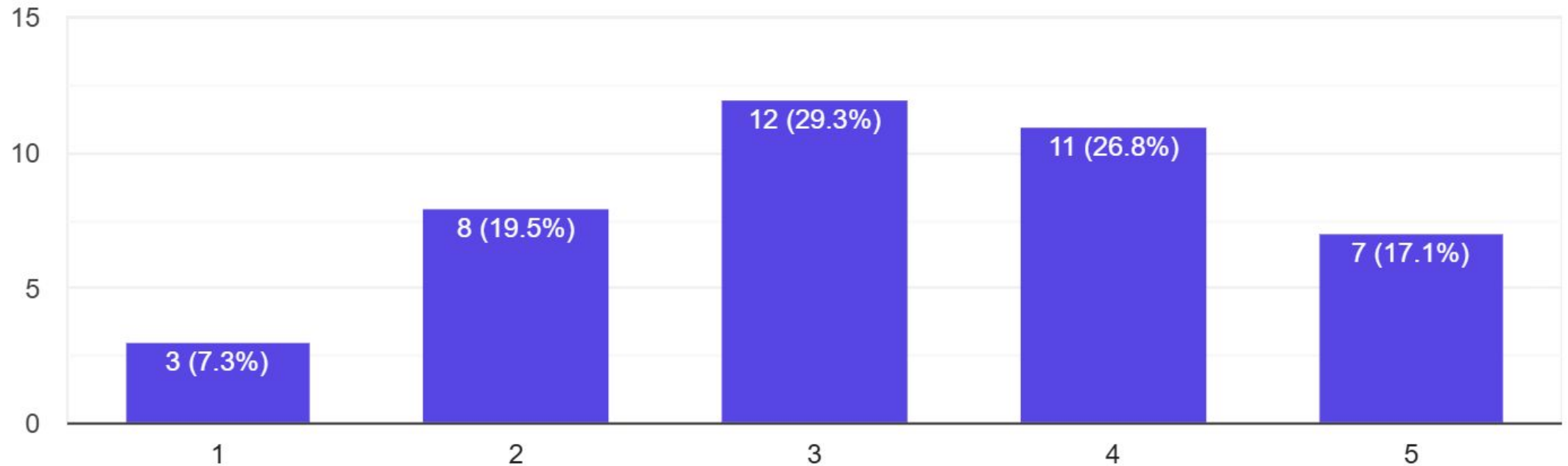
- I easily create fully professional production-ready apps
- I can create projects at the level of a nearly-graduated CMU CS major
- I can create projects typical of someone who has completed CMU 15-112, 15-412
- I could code something like Tetris from scratch, but it might be difficult and I might not have time to
- I don't think I would be able to code something like Tetris, even with lots of help

HW1 survey results

How often do you worry that you are not as proficient or experienced as others in your programming class(es)?

 [Copy chart](#)

41 responses

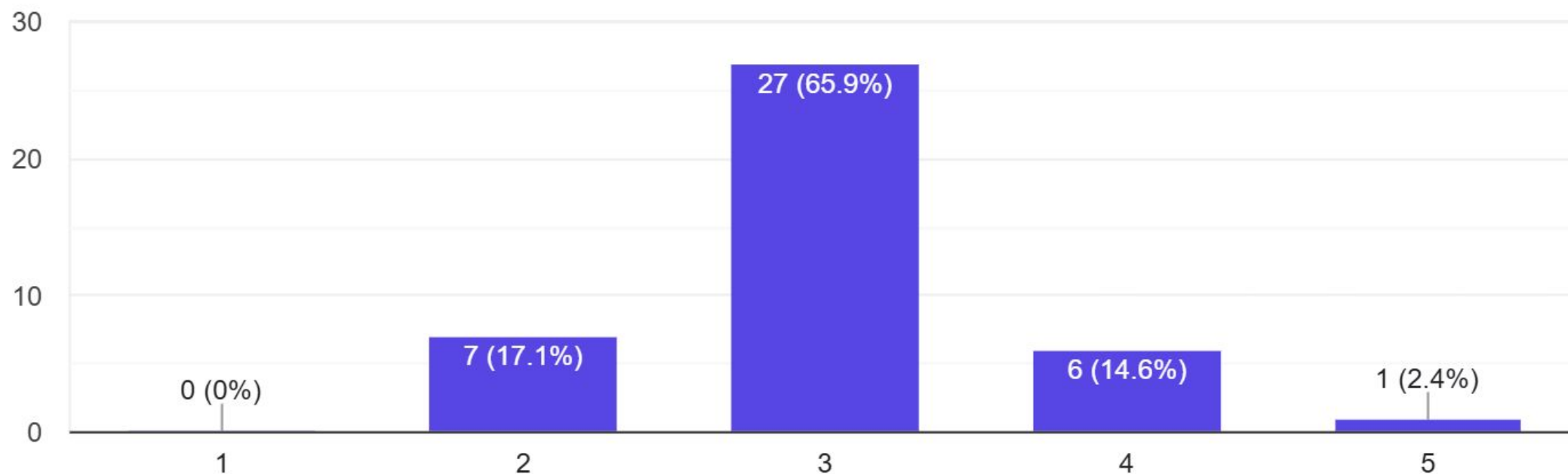


HW1 survey results

How often do you think *other students* in programming classes are worried about their own proficiency or experience?

 [Copy chart](#)

41 responses

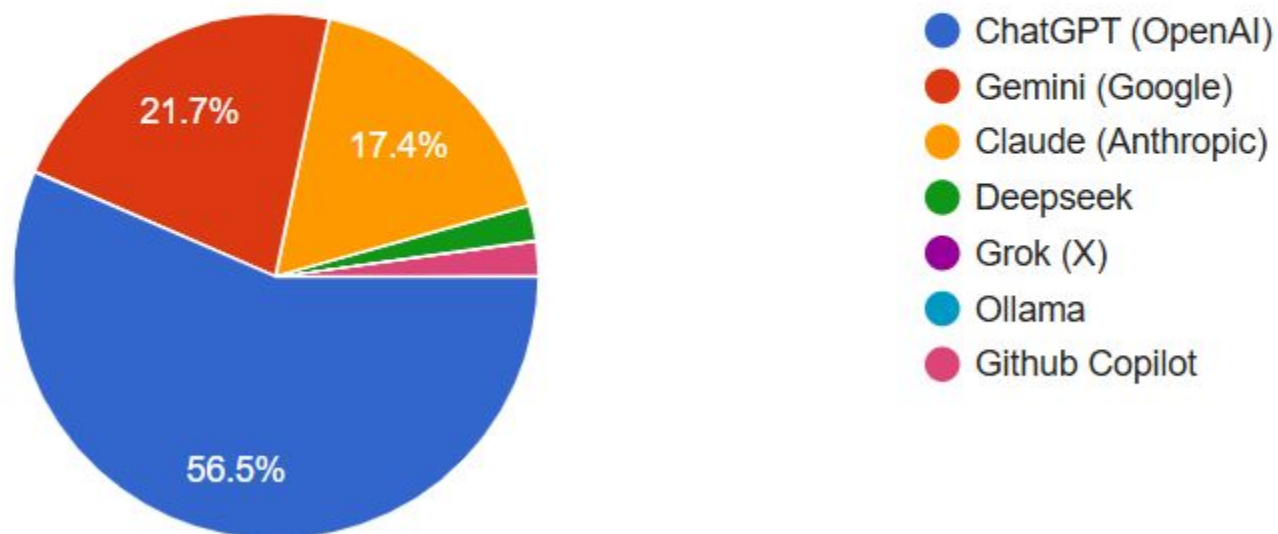


Spring 2026

For coding tasks, which family of models do you tend to use *most* often?

46 responses

 [Copy chart](#)

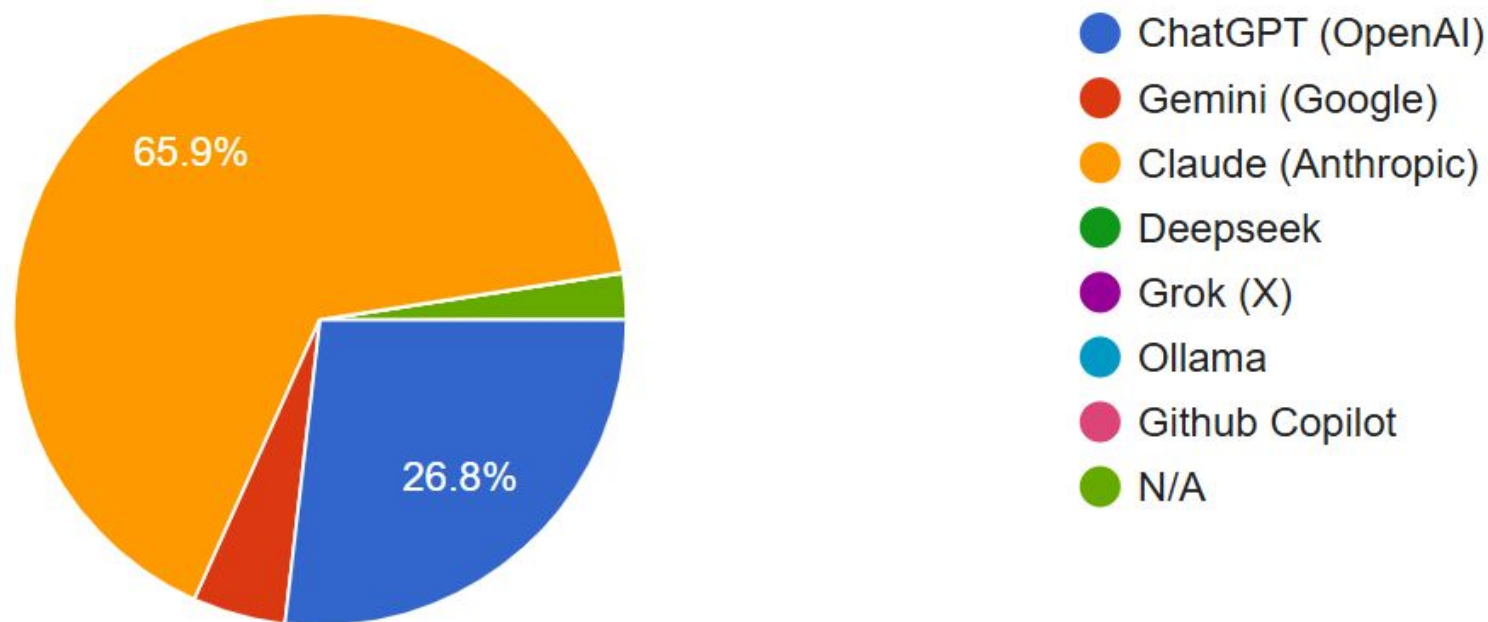


HW1 survey results

For coding tasks, which family of models do you tend to use *most* often?

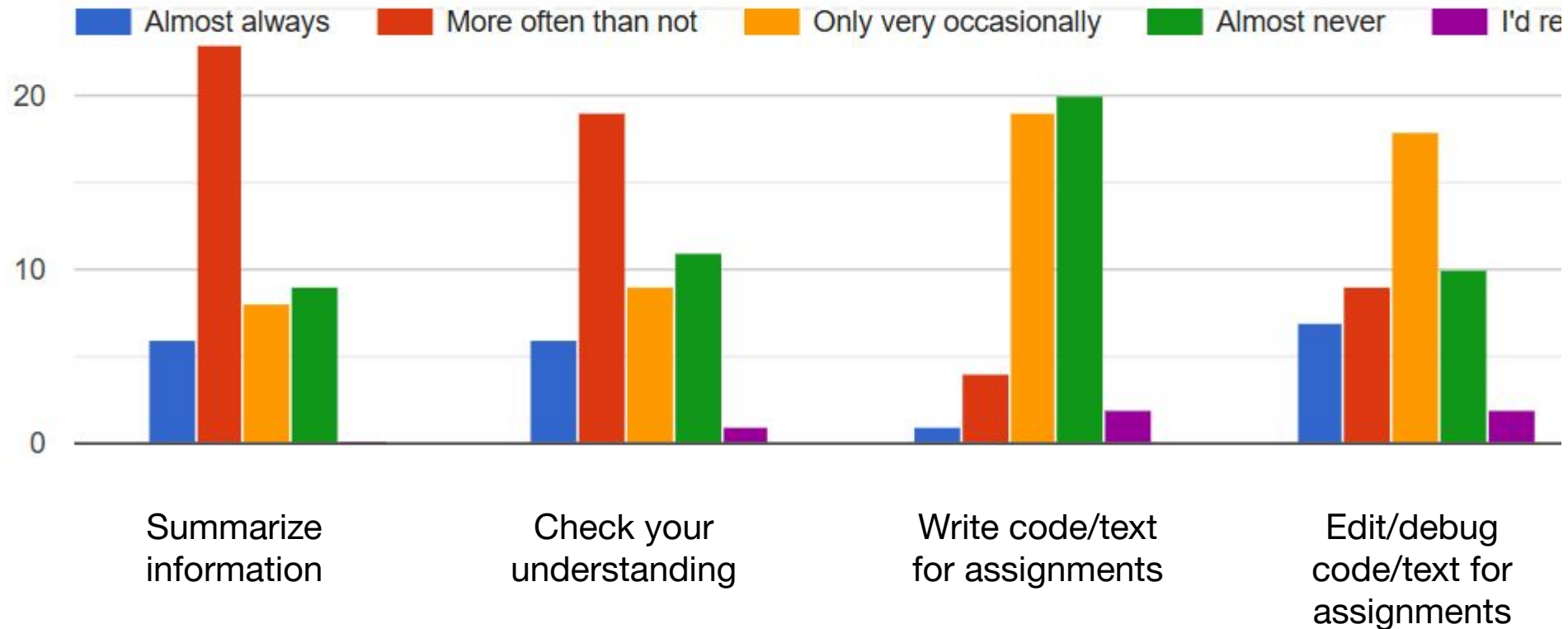
 [Copy chart](#)

41 responses



Spring 2026

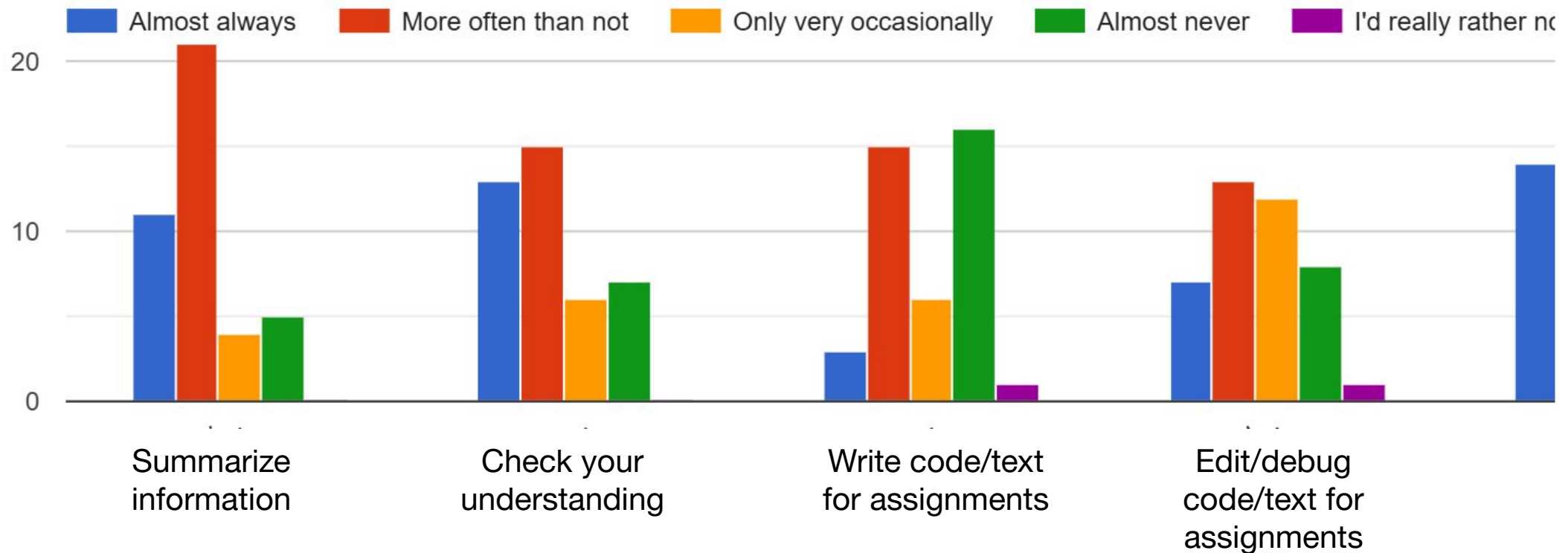
No judgment here. How often do you use generative AI or LLMs (i.e. ChatGPT, Claude, Gemini, etc.) to...



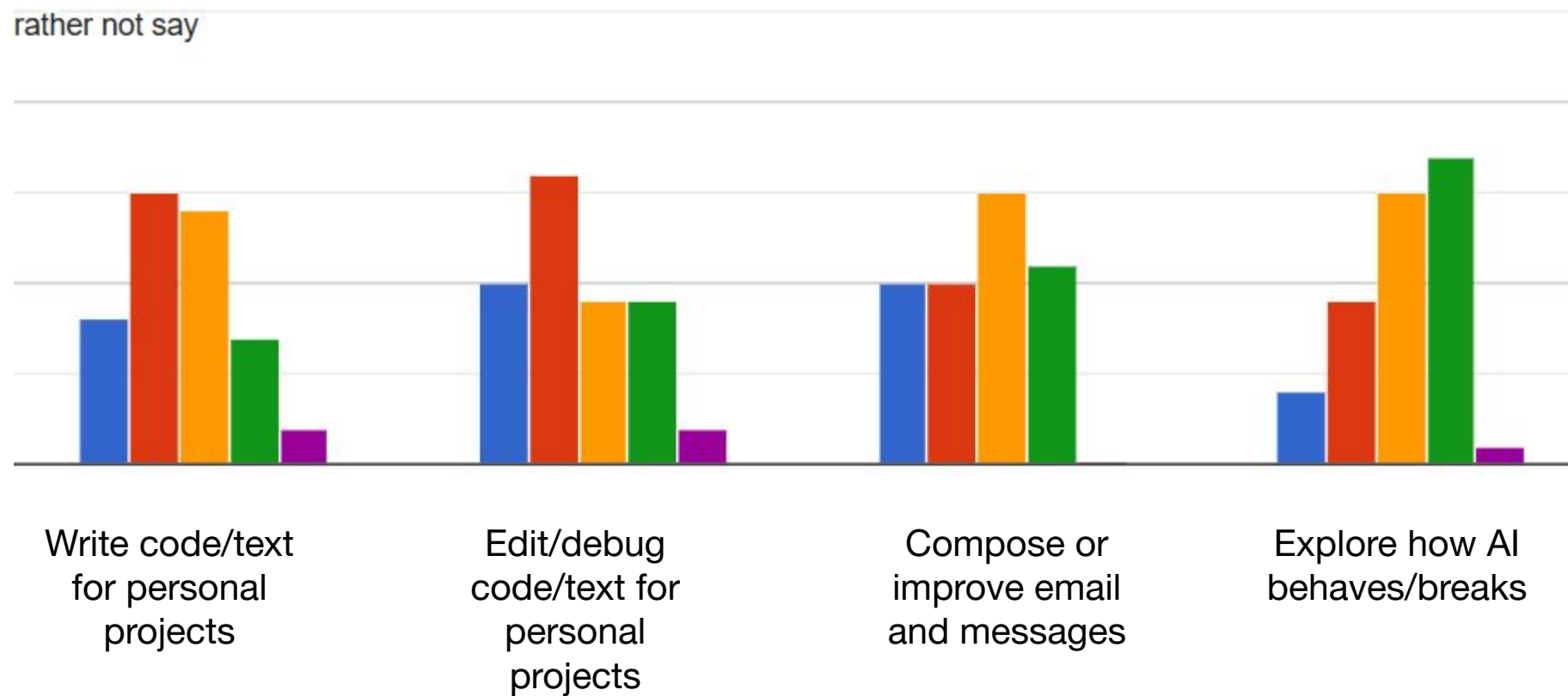
HW1 survey results

No judgment here. How often do you use generative AI or LLMs (i.e. ChatGPT, Claude, Gemini, etc.) to...

 [Copy chart](#)

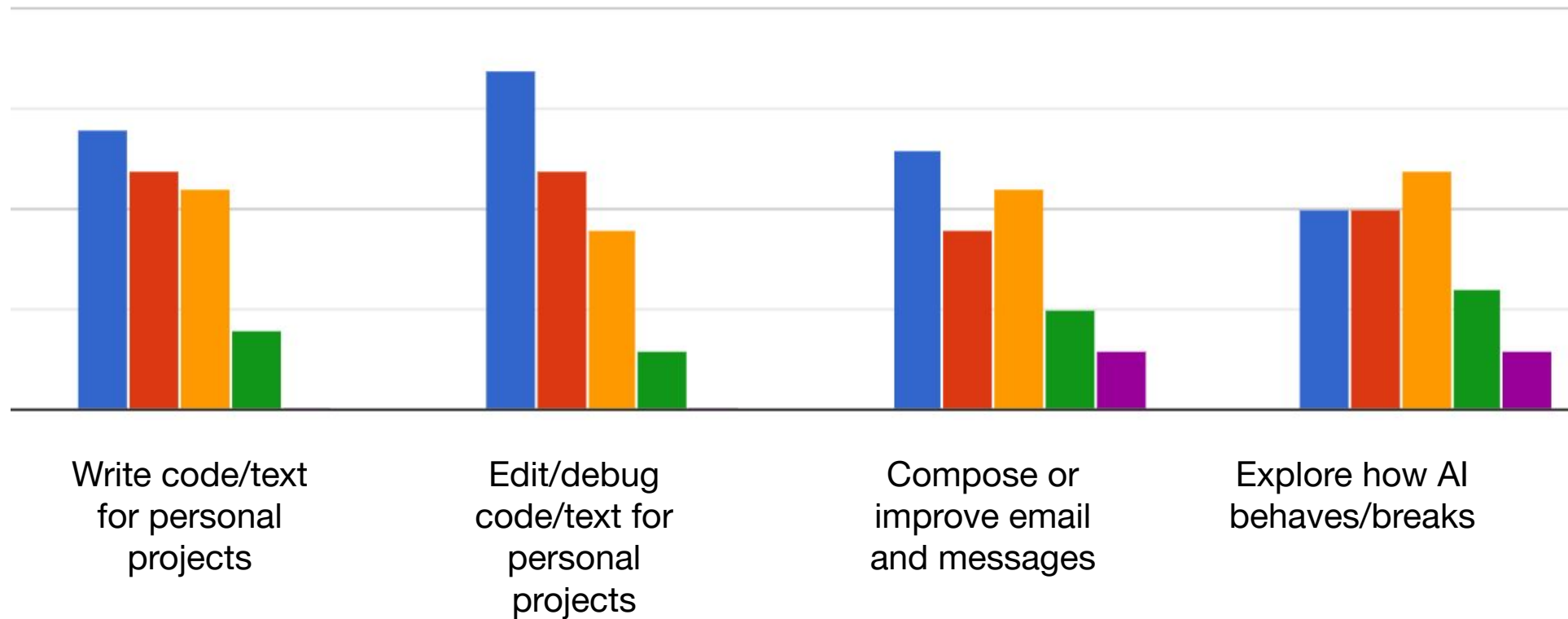


Spring 2026



HW1 survey results

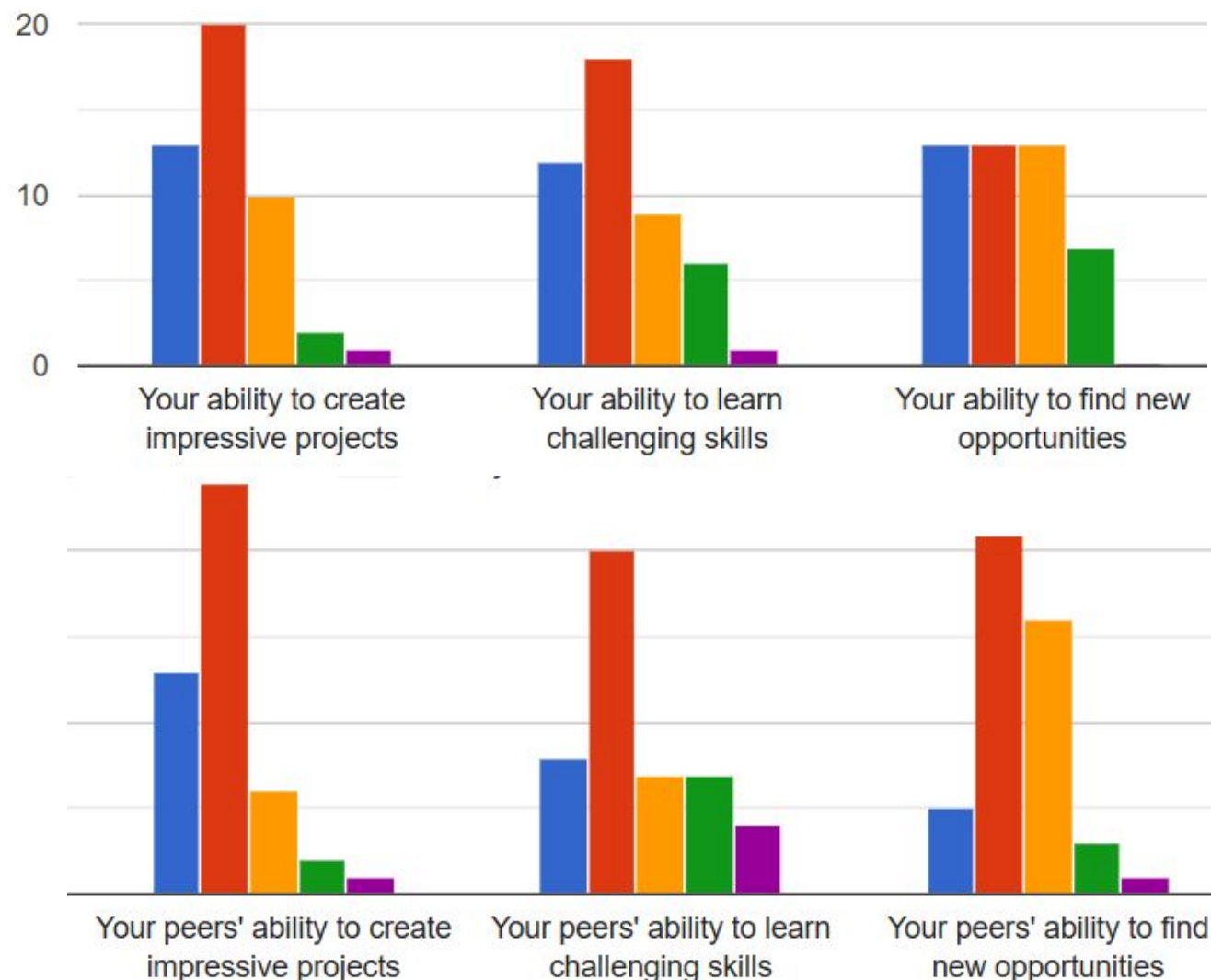
rather not say



Spring 2026

How much do you feel AI has either generally helped or generally hindered the following:

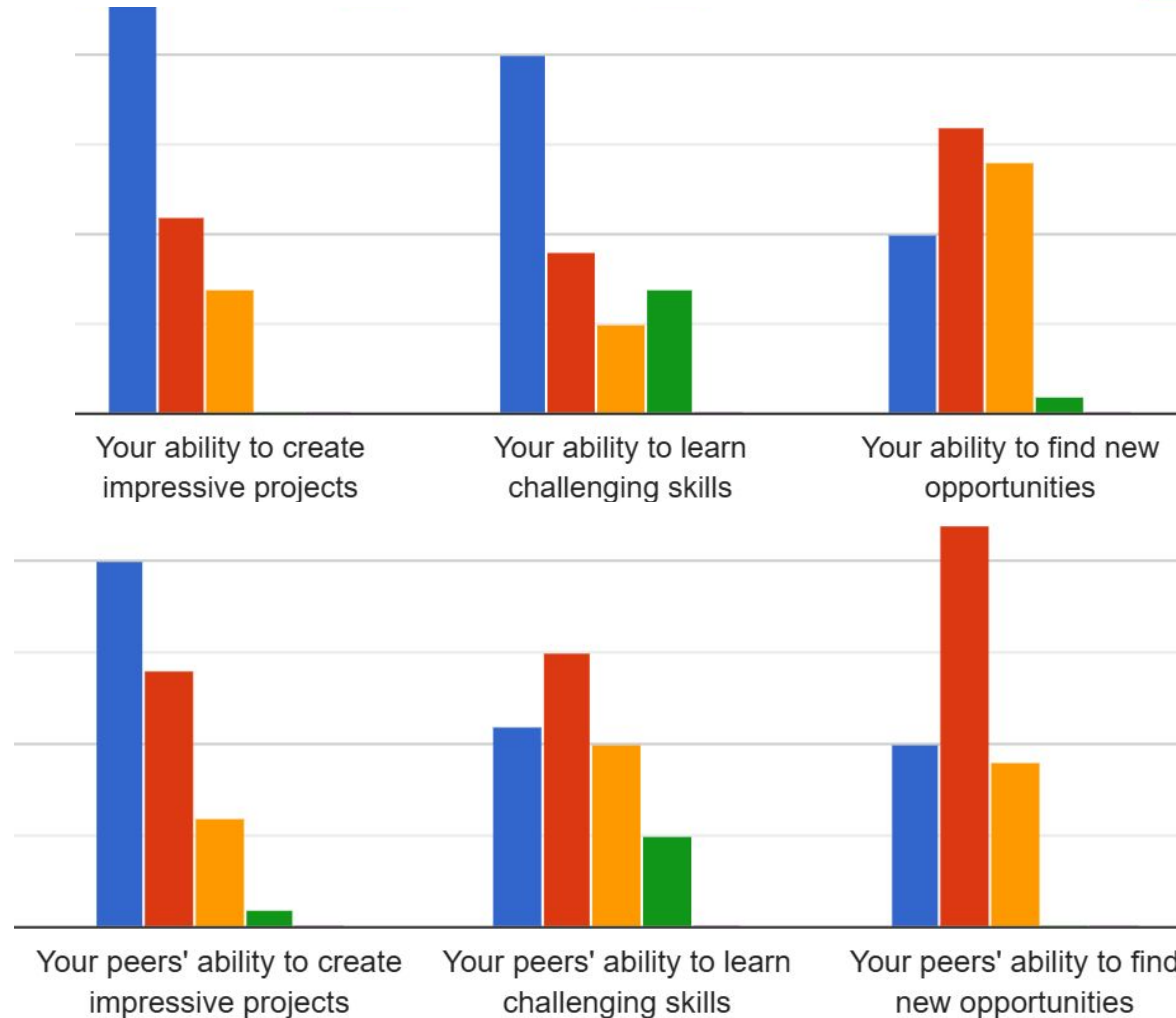
Greatly helped Helped a little Neither helped nor hindered Hindered a little Greatly hindered



HW1 survey results

How much do you feel AI has either generally helped or generally hindered the following:

■ Greatly helped ■ Helped a little ■ Neither helped nor hindered ■ Hindered a little ■ Greatly hindered



The free responses are all over the map but largely match what we saw in S26

- Attitudes about AI in general range from highly skeptical to highly praiseworthy, but everyone seems to be very open-minded. Experience levels are highly variable.
- Common themes:
 - People struggle to fully materialize their ideas with AI (as of right now)
 - Plenty of anxiety for what the future holds / what will be valued
 - Some concerns about not feeling ownership/authorship
 - Awareness that naive AI use might hinder personal growth
- General optimism for being able to learn best practices through experimentation

From the lecture poll last week about websites

Thinking about the "15-112 Fanpage" what are some things that you/your neighbor think we need to do in order to make a website *good*?

47 responses



Response Summary

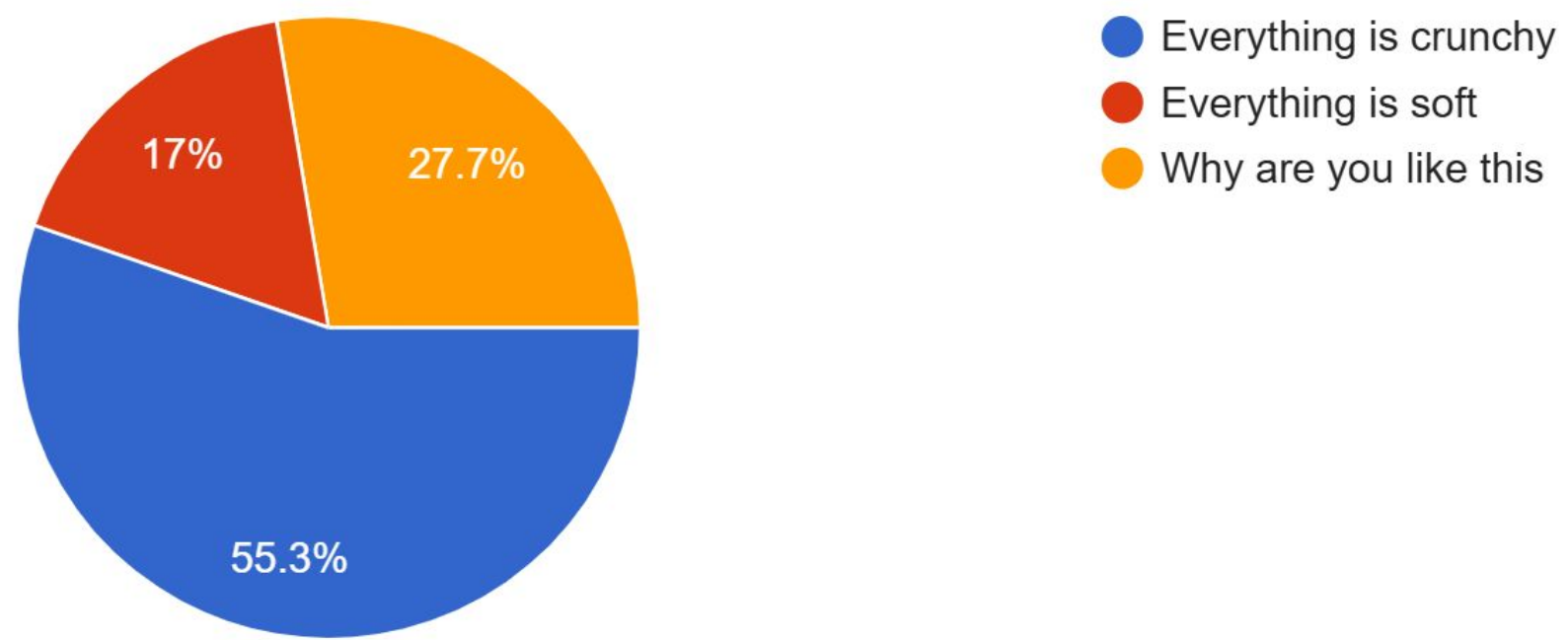


- **Content Quality and Relevance:** Prioritize the inclusion of accurate, original, and substantive course-related information, such as the syllabus, project examples, and student reviews, while removing filler, AI-generated "slop," and excessive slang.
- **Strategic Planning:** Develop a clear design plan, defined project goals, and specific prompts before using AI, treating the LLM as a tool to execute a well-defined vision rather than the sole creator.
- **Layout and Navigation:** Focus on efficient information architecture that is easy to navigate, readable, and structured purposefully rather than relying on generic, cluttered, or "gimmicky" layouts.
- **Aesthetic Design:** Implement professional, clean visual elements, including carefully chosen color palettes that ensure text readability and a design that avoids common, uninspired AI tropes.

Alternate discussion topic: You have been cursed by a wizard. All of your food will now be either crunchy or soft, but you may choose your fate.

 [Copy chart](#)

47 responses



Spend more time getting to know your neighbor, and fill this out as you go

- Fill this out to let us know! <https://tinyurl.com/49rydvja>



With the remaining time: A quick IDE demo

-

Prompting strategies

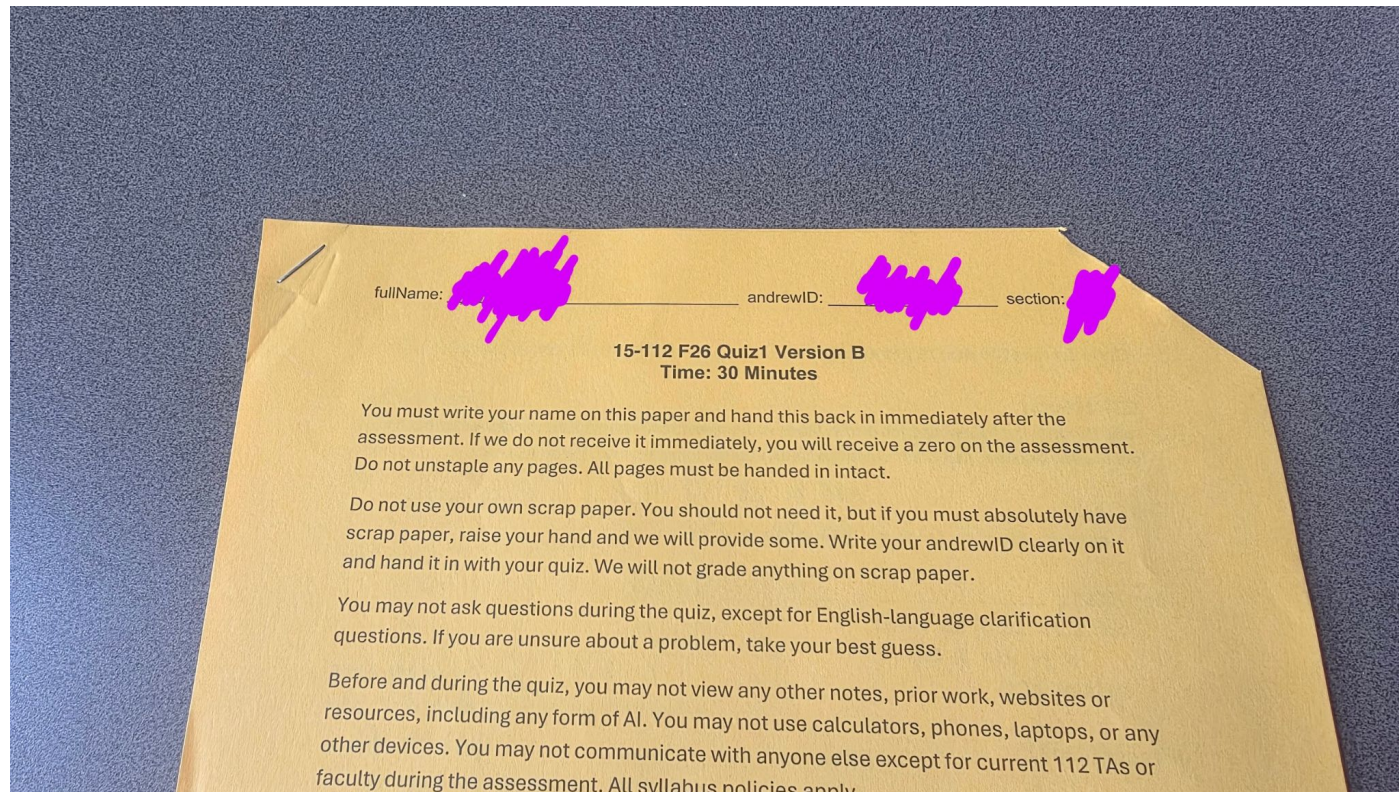
15-113 – Wednesday 9/2

Reminders!

- Check your inboxes for information from our TAs about your pre-deadline meeting with the TAs
 - Bring a work-in-progress for your portfolio for feedback!
 - Ideally try to have it deployed to Github Pages so that they can help you if it's not working (we'll do a demo today)
- A few more people emailed me about ChatGPT validation. I'm still waiting to hear back from OpenAI, but I've asked them to add you manually. Fingers crossed that that will happen soon.
- Mike is very sleepy today (for a reason we'll discuss)

(Briefly) Why is Mike sleepy?

- We're attempting to build an AI-augmented grading pipeline for 15-112. We tested it yesterday, though oddly the things that slowed us down were mostly human-related...



Credit where credit is due:

- One of our best TAs, Jiao, has been architecting the pipeline and testing it extensively. Fun fact: She took 15-113 last semester!
- And many other TAs have been helping us with the process. It's been rapidly coming together over about three weeks
- We're making our own because even though there are tons of people begging us to use their grading pipelines, we want something that we fully control, fully trust, plays to the strengths of our staff, and that makes no design compromises for 15-112 specifically

Goals

- We will *not* replace our TAs for grading. It's important for students to get feedback from real people who are thoughtful and who have gone through the 15-112 process as students
- And we must protect student privacy! (No training allowed)
- ...but perhaps we can speed up our 3-6 hour weekly grading process through pre-grading, which we then carefully QA and use as suggestions, like a buddy grader
- ...and perhaps we can *start* by knowing where the most common mistakes are, so that we can design the rubric to best reflect the feedback we want to provide
- ...and maybe we can increase the accuracy overall

Rough pipeline steps

- Scan the paper quizzes (we can't do computer-based quizzes at a scale of 500-600 students right now, sadly)
- Offline, programmatically de-identify the scans
- Agents 1a and 1b use two frontier models to independently transcribe hand-written solutions and produce a confidence value based on agreement
- Agent 2 Uses mid-range models with prior TA grading as a normalizing reference, review a subset of transcriptions and propose a rubric based on learning objectives and historic norms
- Expert grading TAs reshape/rewrite that rubric to ensure alignment

Rough pipeline steps

- Agent 3 does a quick evaluation based on that rubric and creates a queue of edge cases for expert graders to review (i.e. should this be a minor or major error, should this be considered an off-rubric deduction, etc). Experts process that queue to produce better alignment.
- Agent 3 grades according to the rubric, generating a confidence score and flagging concerns
- Whole staff manually grades each submission with the pre-grading as a starting point
- Extensive manual QA prior to grade release
- Snacks

Let's get back to websites

- Deployment demo on Github Pages

How far along are you?

- Fill this out to let us know! <https://tinyurl.com/yskk8vs3>



A note on 'prompt engineering'

- Remember when prompt engineering was all the rage?
- It still matters, but the exact wording you choose seems to matter a little less these days. AI is remarkably good at giving you what you want as long as you ask for it specifically, and your sentence structure is a little less important now.
- Focus instead on... am I specifying the right details? Am I providing the right context? Conveying my expectations? Letting it know when it's done?

Some common prompt suggestions:

- Avoid ambiguity. Be specific about what you want.
- Start with a high-level goal and then add details.
- Provide examples where relevant (e.g. “Data should be formatted like _____”)
- Offer additional information: “Feel free to ask any clarifying questions before you begin.”
- Ask for extra care: “Double-check that the code meets our initial specifications before presenting it to me, and iterate if necessary.”

Some common iteration suggestions:

- Provide feedback during iteration (e.g. “This is an improvement, but we still need ____.”)
- Add restrictions (e.g. “Make the game appear more visually modern, but do not change the underlying game logic.”)
- If iteration isn’t solving a bug, tell the AI to choose a different approach (e.g. “Do not hardcode piece rotations. Rotate algorithmically instead.” or “Rewrite the collision logic to ensure that collisions are correctly detected.”)
- Ask for debugging output: “To help me test your code, print `f'collision ({object1}, {object2})'` whenever a collision is detected”

Many sources online (for code and in general)

- <https://mitsloanedtech.mit.edu/ai/basics/effective-prompts/>
- <https://cobusgreyling.medium.com/12-prompt-engineering-techniques-644481c857aa>
- <https://ai.google.dev/gemini-api/docs/prompting-strategies>
- <https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/claude-4-best-practices>
- https://medium.com/@the_manoj_desai/code-whispering-practical-prompting-techniques-that-make-llms-code-like-senior-devs-91f276d8c994

Magical Super-Prompts

- Lots of articles and reddit posts claim to have discovered the Very Best Do-Everything Prompt that will Make You Rich and Popular and Happy
- Mike's advice: Approach these with caution. Some may work well for specific tasks, but a one-size-fits-all approach will not likely fit *well*. Never discard your thoughtfulness.
- If you do find a prompt that works well, look at the *structure* and try to learn from that.

One last bit: Role instructions

- You can redefine how most models behave with a well-written role prompt. Several browser-based tools allow you to explicitly save and use these roles if you find a good one.
- Let's try a few in chatGPT:
 - First, for fun, let's have the AI role-play as a fictional character
 - Let's try one that's a bit more functional, and ask it to get to the point while avoiding empty phrases like "You are absolutely right and so smart for noticing that mistake."

Options, if we have time:

- More demos for creating and modifying portfolio websites (making new pages, embedding videos and images, making sense of HTML/CSS)
- More tool-oriented demos for IDE tools like Copilot / Claude / OpenAI add-ons
- VR: Attempt to add gyro support to Pac Dog
- In-class work time / OH for Project 1