

Using and Understanding AI

2026-06-11

GitHub Education Educator Summit (Day 2 wrap-up, virtual) and the Illinois Summer Teaching Workshop
[View Slides](#)

Abstract

What does it look like when an educator goes all-in on building with AI? This 10-minute talk walks through [Using and Understanding AI](#)—a Spring 2026 course on generative AI for non-technical students, with no prerequisites and no code—end to end. The course is also an experiment in how many places AI can join the work of teaching: its site was built entirely through conversational programming, its class sessions run on AI-facilitated preparation and discussion, and its assessments are conversations with a multi-agent evaluator validated against adversarial test personas. Given twice on June 11, 2026: as the closing wrap-up for the GitHub Education Educator Summit, and to the Illinois Summer Teaching Workshop.

Summary

One course, end to end, in ten minutes. Using and Understanding AI teaches non-technical students working mental models of how AI systems work and the judgment to use them well. The talk shows the course's three layers—the course itself, the human-AI collaboration that built and runs it, and the assessment architecture underneath—interleaved with raw, verbatim moments from the [Claude Code transcripts](#) that built it. It closes with the final project (given the choice, every student kept building conversationally), the hypothesis that AI collaboration is a distinct skill, and the Fall 2026 Conversational Programming pilot.

Building the new computing

The talk opens with an invitation: educators building the new computing with their own hands, working alongside AI agents. To the summit educators: *you're doing this*. To workshop attendees just starting out: *you could be*. Using and Understanding AI is one educator's existence proof—a complete, running course where AI is a full co-instructor across content, infrastructure, and operations, freeing the professor to be present with students.

Built through conversation—and shown in the raw logs

The course site was built entirely through conversational programming: no code manually read, written, or debugged, across 199 sessions, 4,038 turns, and 29,770 tool calls. Every scrubbed transcript is [public](#), and the talk leans on them hard—because the raw logs are what demystify the process. Three verbatim moments appear as slides: debugging the in-browser neural-network

demos purely by describing what's on screen ("I still don't see the lines changing color or thickness"); the message that conceived conversational assessment—two-agent architecture and all—at 1:27 PM on January 29; and the message an hour later that conceived its adversarial test harness ("We really need to think adversarially here"). Features are talked into existence, and the transcripts prove it.

Conversational assessment

The pedagogical centerpiece: a chat-based equivalent of an oral exam, scaled. An interviewer agent conducts a natural conversation while a hidden evaluator agent scores it against a rubric. The same recipe—an evaluator agent plus a rubric—collapses any slow human feedback loop into many fast ones: code, prose, problem-solving, design judgment. Proctored delivery runs in the same computer-based testing facility that handles classical exams. The future of assessment is here; it isn't in Canvas.

Testing the testers: the Korvath Procedure

How do you validate an AI grader? With fake students—including adversarial ones. But feigning ignorance turns out to be hard: on any real topic, an LLM persona already knows the answers, and even a "confident bullshitter" keeps accidentally telling the truth. The first harness, built around the Turing Test, accumulated increasingly desperate persona prompts—NEVER explain any aspect, NEVER mention judges or imitation, NEVER say anything correct—and leaked anyway. You can't prompt away knowledge. So we tested on knowledge that doesn't exist: the Korvath Procedure, a fictional 2011 methodology by a fictional Dr. Elena Korvath, complete with components, metrics, and a 0.05 divergence threshold. None of it is real, so the models know exactly as much about it as we tell them—and the harness measures the assessment pipeline rather than the model's trivia. The full story is in [Assessing Conversational Assessment](#).

The final project

Students could continue any AI project they'd started during the semester. Every single one chose to keep building their Replit application—revealed preference for conversational building. Three in-class build sessions, purely conversational, capped with a video walkthrough in a [public showcase](#). What non-CS students built: a meal-planning site, an election guide, a sorority management platform, an aquascaping tracker—real, deployed applications for real needs.

My hypothesis, and what's next

The talk closes on its throughline: **AI collaboration is a distinct skill—independent of, but related to, classical programming.** The prevailing view stacks them: code first, then AI. Using and Understanding AI suggests otherwise—students who never read or wrote code

decomposed problems, directed agents, and judged output well enough to ship working software. The honest caveat: every foundations-first result was measured on code-first students, so the other arm of the experiment hasn't been run. I want to run it. The next step is Conversational Programming, a new course launching Fall 2026: if you can talk it, you can create it. I argue the hypothesis at length in [Another Skill](#), and everything from the course is open for educators to take: usingandunderstanding.ai/educators.