talk notes · october 14, 2026 · bradshaw library

AI is more than a chatbot.

What changes when you give a model something to read and something to reach, three ways I use it in my teaching, how to start with one folder, and the harder question it leaves about what we teach.

> the idea
>
> Most of us use AI to answer questions and polish emails. That is a chatbot, and it is a small fraction of what the model behind it can do. The model starts doing real work when you give it **something to read** (your own material) and **something to reach** (the systems it can act in). And once every student has the same tool, the bigger question is not how we teach but what we teach.

A talk with a live demo for faculty and staff at FGCU's AI Community of Practice, hosted by the Bradshaw Library and the Dendritic Institute, followed by questions. The room ran from daily users of AI agents to people who had tried a chatbot and stopped. I trained as a mechanical engineer, spent ten years on Wall Street at BNP Paribas and Citadel, and now run treasury in my day job while teaching finance at FGCU. The slides and handout are at **akram.education/talks**.

The gap between the chatbot and the model

ChatGPT launched in November 2022, built on GPT-3.5. The models came first; the chat window was the killer app, and it was striking enough that most people stopped there. But the model behind a chatbot can sit in many other places, and the distance between what chatbots do and what the models can do is a gap I would like to spend my career closing.

The number that made the point for me: an AI agent that reviews my lectures scored my verbal delivery **3 out of 5**, flat for 28 lectures in a row. A transcript cannot hear tone, so I had it count something a transcript can show. In one lecture I improvised, I said "like" **202 times**. Nobody had ever told me. My filler words went from **2.2 a minute** last spring to about **1.3** on a good day this fall. ChatGPT could never have told me that. It has never seen me teach.

read: what it can look at	reach: where it can act
Lecture transcripts, your syllabus and notes, a teaching rubric, last semester's reviews	Canvas, the printer, your files, your calendar

Driver, road and car

Three words matter when you build with AI: the **model**, the **context** and the **harness**. Think of a driver, a road and a car. A chatbot gives you the fastest Formula One driver in the world on speed dial: brilliant, but he only talks. Put him in a car (a harness such as Claude Code, which can open files and run tools) and

he can act. Give him a road (your context: files, courses, rules) and he has somewhere to go. The driver is replaceable: I use Claude for most of my work and hand some tasks to other models, and if a provider raised its price sharply I would swap the driver and keep the car and the road.

Case 1: the coach

I knew my material, but I had never stood in front of students, and explaining a call option to an investor is not explaining it to someone who has never seen one. I am a practitioner, not a trained teacher, so I asked AI to research what good teaching is, teach it to me, and build a rubric to score my lectures against. It turned into four habits: **a hook on the board** before class starts; **"explain it back to me"** instead of "is this clear?", which only gets nods; **calling on the quiet students** (the review reports how much of the class students spent talking); and **a timer five minutes before the end**, so I always close with a recap.

Every class is recorded (audio only, with my students' permission), transcribed and scored against the rubric, with evidence from the transcript and things to try next time. My average last spring was **3.87 out of 5**; a lecture this October scored **4.30**. Once AI learns a job, it runs the job: record, transcribe, score and review all happen without me, and the review is waiting by the time I get home.

Case 2: the teaching assistant

Most of us do not have a TA. You can teach AI to be one:

- **Canvas.** It builds modules, writes reading-check quizzes and schedules announcements. For my seven-week online course, all seven modules were live before the course opened; once I had taught it how, each module took minutes of my time. I did not even know how to connect it to Canvas. I told it I use Canvas, asked for help, and it set it up.
- **Reading checks that tell me where to teach.** A short quiz before each lecture gets students reading. The agent pulls the results from Canvas, so I walk in knowing which ideas they missed.
- **Handouts with the intuition in them.** A chatbot will tell you a straddle is a long call plus a long put. My agent was in the lecture: it knows what students asked and the point I made, that a straddle is **a bet on volatility**.
- **A plan every class morning.** It knows my syllabus, how far I got last time, where students struggled and, I will admit, whether I said something backwards. A two-page plan for the day is waiting on my printer.
- **Explainer videos,** for students who learn from a picture or a real company rather than a formula. The cash cycle is a good one: the days a business has to finance itself between paying for inventory and collecting from customers. Amazon's is negative (about −42 days in its 2025 annual report), because it collects before it pays its suppliers. A whiskey distiller pays for grain and barrels while the whiskey ages, sometimes ten years or more, before a bottle sells.

One rule makes it safe: **AI drafts, I review, it posts.** It is not perfect, and nothing reaches a student without my yes.

Case 3: the researcher

AI's superpower is reading more than I can, fast: think of someone who has read a billion books and remembers every fact. I no longer do the first cut of research myself, for the same reason an accountant does not build the balance sheet by hand. The gathering is automated, so I can spend my time on what it

means.

- **A weekly digest.** I get about 120 emails a day, half of them newsletters, and follow about thirty podcasts plus posts, threads and the news. Every weekend an agent reads it all and returns two lists: **recurring themes**, the ideas many sources repeated, and **novel ideas**, the ones said for the first time. I pick the handful that matter and have it dig further.
- **A daily market note, with a verify step.** Every weekday it writes a note for my team at work. One morning it called the 10-year Treasury yield "unchanged" when it had just hit a 19-year high: its data feed was behind. Now every number is checked against its source and its date. Trust it, then build the check into the job.
- **Every SEC filing.** An agent reads every quarterly report and insider filing from two public companies I follow for work. It caught a fund quietly building a stake of nearly 20% in one of them, which explained a sudden jump in trading volume. An investor that crosses 5% must disclose it on Schedule 13D, or the short-form 13G (this one was a 13G). The news never picked it up, so no chatbot would have.

That is the skill I want students to have: **go to the primary source, then check the work.**

Getting started takes one folder

You need a folder and a harness. **Claude Cowork** is a desktop app that asks you to pick a folder (the road) before you start. **Claude Code** does the same in a terminal and installs with one command from claude.com/claude-code. Despite the name, it is not just for coders.

In the demo I gave it a screenshot of a folder layout (one folder per course, each with a context file, a to-do list and a status file) and a sentence about who I am, and it built the workspace. The first file it wrote was **CLAUDE.md**, the first file any agent reads when it opens the folder: it is how the agent remembers you, so you never re-explain yourself. Then I dropped the course syllabus in as a PDF and asked for a course plan. It read the screenshot and the PDF without any reformatting from me. The next step is scheduled tasks, for example: every Tuesday at 11 a.m., pull the latest lecture recording and send me a summary.

Privacy and student data

Your files stay on your computer, but unless you run a local model, the inference happens in the vendor's data center. Under commercial and enterprise terms, such as Anthropic's, the vendor may not train its models on your data. Consumer plans have a setting that controls whether your chats are used for training. A university-licensed tool such as Copilot works the same way: no training, but your files are still processed on the vendor's servers.

Whether that counts as sharing the data is not settled; I am not a lawyer, and much of our privacy law predates AI. What is clear: **FERPA binds**. I keep grades and student records out: my agent builds the course and posts announcements, and it does not post grades. Start with your own material, and check your institution's guidance before sharing anything that identifies a student.

What it costs

I pay about **$3,000 a year** for AI, all in. Most of it is a $200-a-month Claude plan, because I want the newest models; the rest is basic plans for two other models, a narration service and a transcription service. You do not need any of that to start: open-source models and harnesses are free to download, which brings

the software cost close to zero.

How we teach, and what we teach

AI will change how we teach and how we work, as much as email, the internet and the calculator did. That is the easy half. The harder question is **what** we teach, because our students have the same tool.

We invented the calculator and still teach arithmetic, so students develop an intuition for numbers. We still assign term papers, so students struggle a little and learn what good looks like. I tell my students that any take-home quiz can be pasted into AI and solved, and that they will learn nothing. If I need to lose weight, I can run three miles or drive them. Driving is faster and more comfortable. Only running works.

AI is a force multiplier, but it has to multiply something. Zero times anything is still zero. So push students to use AI, and move what we teach from formulas toward intuition, creativity and extending what we know. If your value at work is regurgitating knowledge, that value is now zero. If it is using knowledge and extending it, it has never been worth more.

From the questions

- **Employers expect AI-fluent graduates.** The junior hire has always been expected to know the newest tools; my first two years on Wall Street were Excel and PowerPoint. Telling students not to use AI is bad advice. The difference is that Excel could not do your homework, and AI can. How to teach fluency without it becoming a crutch is a problem I have not solved.
- **Animations are cheap; voices are not.** Most of my animations need nothing but Python, which is free. What costs money is AI narration.

Takeaways

- Give AI something to read and something to reach. A better prompt does not fix a chatbot that has never seen your work.
- Write your context down once, in one folder. Show it a job once, then let it run the job.
- AI drafts, you review, it posts. Nothing reaches a student without your yes.
- Trust it, then check the source and the date, and teach students the same habit.
- Teach the intuition, not just the formula. That is the part a model cannot hand your students.

Go further

- Claude Code documentation: code.claude.com/docs.
- SEC EDGAR full-text search: sec.gov/edgar/search, every public company filing, free.
- US Department of Education, Student Privacy Policy Office: studentprivacy.ed.gov, guidance on FERPA.
- My explainer videos: *Operating Cycle and Cash Cycle Explained* (youtu.be/6hI6Nj9eE6g) and *Financial Ratios and DuPont Explained* (youtu.be/MZwCar8RAFo).

Akram Ayyash · Adjunct Professor of Finance, Florida Gulf Coast University · akram.education
Version 1.0, October 2026. My own notes, free to use for learning. Not an official FGCU document.