15: Learning, careers, and responsible use

Use AI to learn, not to skip the learning, and own what you ship

Why this matters

- At work, the output is the point
- When you're learning, the output is a side effect; the struggle teaches
- The tool that speeds experts up lets learners skip the struggle
- Entry-level software hiring is under real pressure
- Whatever you ship is yours, whoever typed it

Learning a new library with AI (Anthropic, 2026)

52 mostly junior engineers learned Trio, a Python async library, then took a quiz.

	AI group	Hand-coding group
Quiz score	**50%**	**67%**
Time to finish	About 2 minutes faster (not significant)	Baseline

The biggest gap was on debugging questions.

How people used the AI mattered

Pattern	Quiz score
Asked only conceptual questions	65% or higher
Asked for code together with explanations	65% or higher
Generated code, then asked follow-ups to understand it	65% or higher
Delegated the task	Below 40%
Relied on AI more and more, or let it debug for them	Below 40%

Each subgroup had 2–7 people. Suggestive, not proof.

The pattern repeats

Study	Finding	Read it carefully
High-school math, ~1,000 students (PNAS 2025)	Plain GPT-4: ~17% worse on a later exam. Tutor version: no significant harm	Students mostly used plain GPT-4 for answers
319 knowledge workers (CHI 2025)	More confidence in AI, less critical thinking	Self-reported survey
"Your Brain on ChatGPT," 54 people (2025)	Weakest brain connectivity; trouble quoting their own essays	Preprint; 18 finished the key session
METR, 16 developers (2025)	19% slower, but felt 20% faster	Early-2025 tools

What it adds up to

Outsourcing the thinking hurts the skill you outsourced.

Using AI to think better mostly doesn't, and sometimes helps.

Feeling productive isn't evidence of learning.

Learn with it; don't outsource to it

- Attempt it alone for 15–30 minutes; note where you got stuck
- Ask for hints and concepts, not finished answers
- Explain it back, then have the model check your explanation
- Read every line; ask about any line you can't explain
- Debug yourself first; form a hypothesis before asking
- Rebuild it the next day without AI

Tell the tool how to help

```
I'm learning Python async with Trio. Don't write the code for me.
Here's my attempt and the error I get. Ask me one question that
points me toward the bug, and wait for my answer before saying more.
```

- Claude Code: `/output-style learning` leaves `TODO(human)` work for you
- ChatGPT has a study mode; other chat apps have similar modes
- Keep a "things I didn't understand" log; review it weekly
- Turn off autocomplete while you learn a new language

Academic integrity: common policy patterns

Pattern	Example	For students
Default restrictive	Stanford: study aids and surface edits, not deliverables, unless the course says otherwise	Assume no unless told yes
Course-provided AI only	CS50: its own AI tutor, the "duck"	Use the tutor the course provides
Allowed with disclosure	Many writing and project courses	Use it, say how, own the result
Required	Courses like this one	You're graded on judgment

Check each syllabus. When in doubt, ask the instructor in writing.

A three-tier course policy

Tier	What's allowed
AI-free	No AI tools. Tests fundamentals you must have without help
AI-as-tutor	AI may explain, hint, and review. It may not write code you submit
AI-accelerated	Any AI tools. Graded on the result, your decisions, and explaining every line

Plus an AI log, ownership of every bug, a 5-minute explain-back, and no pasting others' data.

Careers

What the evidence says, and what to get good at

Early-career jobs: what the data says

- Stanford Digital Economy Lab, using ADP payroll data on millions of workers
- Ages 22–25 in AI-exposed jobs: **19% below** less-exposed peers' pace
- Data through June 2026; the gap has widened since 2025
- Mostly fewer hires of young workers, not layoffs
- Concentrated where AI substitutes; flat or rising where it complements
- The authors call it descriptive, not causal

Counter-evidence and uncertainty

- Other causes: post-2022 over-hiring correction, interest rates, offshoring
- BLS still projects 10% growth for developers, 2025 to 2035
- Stack Overflow 2025: 84% use AI tools; only 33% trust them
- Top frustration, at 66%: "almost right, but not quite"
- If companies stop hiring juniors, they eventually run out of seniors

Skills that gain value

Skill	Why it matters more with agents
Specification	Agents do what you ask; say precisely what you want
Decomposition	Pieces an agent can do and you can verify
Reading and review	Spot the subtle bug in 400 plausible lines
Debugging	Someone has to find the real cause
Systems thinking	Agents optimize locally; architecture stays a human call
Product sense	More of your day goes to *what* and *why*

Also testing, security judgment, domain knowledge, communication, and taste.

The classes you thought you'd never use

Class	Why it matters more now
Software engineering and project management	Estimation, scheduling, dependencies, risk
System design	Choosing the design; knowing what to ask about scale, failure, cost
Algorithms and data structures	Spotting the accidentally quadratic; passing interviews, for now
Communication and product	Clear specs, demos, what users actually need

Brooks's law (1975): "adding manpower to a late software project makes it later." It applies to agents too.

Bryan's take: Ultimately the engineer is much more in product now. Some of the classes software engineers take, the ones they don't think will be useful, turn out to be among the most important. I used Gantt charts and quoted *The Mythical Man-Month* far more in my career than I expected. You still need to know the algorithms to pass the interviews, at least for now. Having good system design chops helps a lot too. You need to know what questions to talk to the agent about.

Portfolios and interviews

- Ten AI-generated apps prove little; show judgment instead
- One or two deep projects, maintained, ideally with real users
- READMEs on trade-offs, hard-bug write-ups, honest AI usage
- Interviews are split: some expect AI (Canva, June 2025), some forbid it
- Practice both: solve without AI out loud, and drive an agent
- Narrate the moment you reject its suggestion, and why

Responsible use

Data, licenses, and who answers for the result

What not to paste

- Passwords, API keys, tokens, private keys, connection strings
- Personal data about others: customers, patients, classmates, coworkers
- Proprietary code, unless that tool is approved for that use
- Other students' work
- Anything you'd hate to see in a breach notification

Consumer and business terms differ. Check your settings; defaults change.

Copyright, licenses, and attribution

- US Copyright Office (Jan 2025): "prompts alone" are unlikely to qualify
- Code you shape, review, and modify stands on firmer ground
- Models can reproduce GPL code, with obligations if you distribute it
- Use public-code filters and license scanners in CI
- Follow the project's rule: `Co-Authored-By:`, `Assisted-by:`, or none
- Attribution says how it was made; you're still the author of record

You answer for what you ship

- An agent can't be fired, sued, or held to a standard. You can.
- "The AI wrote it" is an explanation, never an excuse
- Check people-facing logic for bias, and generated UIs for accessibility
- Run only as many agents as you can actually review
- Don't send what you haven't read; that's "workslop"
- AI-first doesn't mean AI-only; keep skills for when tools fail

What "vibe coding" means here

> **Bryan's take:** "Vibe coding," as this course uses the phrase, means AI-accelerated engineering with a human in charge, not blind acceptance of whatever the model produces.

Try it

On your own, run a two-week learning experiment:

1. Pick a library or language you've never used
2. Week 1: build with an agent as usual; rebuild one feature by hand
3. Week 2: build again with the habits: attempt first, hints, explain-back
4. Compare in one page: what you learned, how fast, which habits stay

As a class or team, draft an AI use policy, swap drafts, hunt for loopholes, then vote.

Discussion

1. The AI group wasn't faster and learned less. When would you still use it?
2. Should an intro course ban AI, require it, or teach both?
3. If agents do junior work, where will future seniors come from?
4. An agent produced 300 lines that match GPL code. What do you do?

Read more

- Module 15: Learning, careers, and responsible use
- How AI Impacts Skill Formation (Shen, Tamkin, 2026)
- Canaries in the Coal Mine? (Brynjolfsson, Chandar, Chen, 2025, revised 2026)
- Generative AI without guardrails can harm learning (Bastani et al., PNAS, 2025)