Motivations for (and Effects of) Using AI Tools in an Introductory Programming Class

Alberto Varone[1], Michael Fudge[2], Kevin Crowston[2] & Francesco Bolici[1]

[1]University of Cassino and Southern Lazio [2]Syracuse University

Over-arching questions:

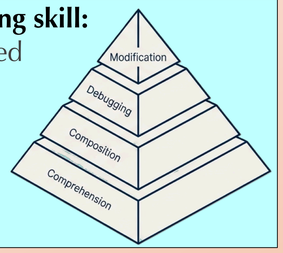
- What is the effect of using AI tools on skill development and maintenance?
- Short-term benefit: AI boosts task performance — especially for novices
- Long-term cost: Use may erode the practice that builds real skill
- Why would students risk losing skills?

Theory: Skill and skill acquisition:

- Ability to produce something of value
- Expandable with practice
- Socially determined

Programming skill:

- Hierarchy of skills needed for programming
- AI can now do much of the work
- How does AI use affect skill acquisition?



Hypotheses:

- **H1a**: Using AI to solve problems (executive help) harms learning; **H1b**: using it for support (adaptive help) helps learning
  - Skill acquisition theory: practice and proceduralization drive skill development. Executive help-seeking bypasses both; adaptive help-seeking supports both.
- **H2a**: Extrinsically-motivated students use AI to solve problems; **H2b**: intrinsically-motivated students use AI to support learning
  - Self-determination theory: intrinsic motivation drives mastery-seeking; extrinsic motivation drives completion-seeking.
- **H3a & b**: Students under time or academic pressure will use AI to get work done
  - Under cognitive load and time pressure, effortful instrumental use becomes less likely; quick delegation more tempting.

Study design:

- Three semesters of data (S25, F25, S26) from an introductory Python course for non-majors
- Students had a course-specific AI chatbot

Data:

- **Survey** (F25, S26)
  - *Academic Motivation Scale* (AMS)
    - Intrinsic & extrinsic motivation subscales plus amotivation and required course
  - *Time and academic pressure*
  - N = 124 usable responses (high level of failed attention checks)
- **AI usage logs** (S25, F25, S26)
  - Sessions, prompts, response length
  - Prompt & response coded for help seeking
  - N = 198 students (21,233 interactions)
- **Exam grades** (S25, F25, S26)
  - Midterm exam score (45 question in class on paper programming skill assessment)
  - N = 273
- **Pretest grades** (S26 only)
  - N = 104
  - Higher for those with prior class
- Many non-users of the class bot
  - Use fluctuated from semester to semester
  - N = 192 with AI use and midterm grade
  - N = 89 with survey and AI use

Example interactions:

- **Executive help seeking**
  - "How should I go about creating a story for this homework? [problem text] can you give me the full code"
  - "give me steps for sentiment function and steps for main function"
  - "Were you prepared for this assignment?"
- **Adaptive help seeking**
  - "how can I skip a line in a file that I am printing if it has a str I don't want?"
  - "I dont understand this type, int, str stuff... what does it mean?"
  - "...dont tell me the code but i want to know if for step 1 i am supposed to…"

Coding process:

- Interactions coded by 2 research assistants
- Interactions discussed in weekly meetings and disagreements resolved
- Currently at 90% agreement (Krippendorf α)
- Now working on LLM coding

Chatbot evolution:

- Spring 2025
  - Half had chatbot with RAG on class
- Fall 2025
  - All had access to RAG chatbot
- Spring 2026
  - Chatbot defaulted to "tutor mode", but can be switched to "answer mode"
  - Significant decrease in length of answers and how much code the chatbot provides

Findings—H1:

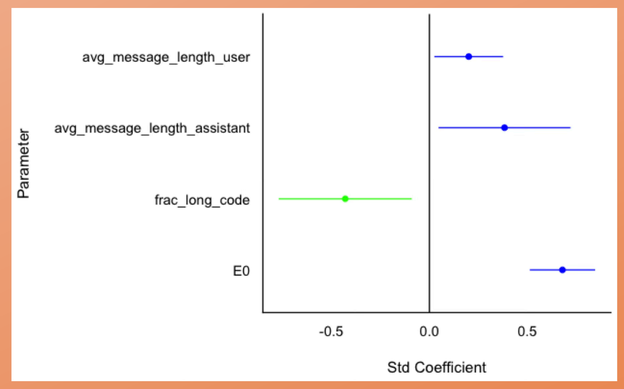
- Chatbot users had higher midterm grade than non-users (Δ= 2.05, t(150.25)=2.31, p=0.022), a small difference (Cohen's d=0.38)
- However, slight negative correlation between total prompts and midterm grade (Spearman's $\rho = -0.15$, $p = 0.044$)

Findings—H1a & b:

- **H1a**: significant **negative** correlation between fraction of **executive** help seeking prompts and midterm grade (ρ = –0.45, p<0.01, N=38)
- **H1b**: significant **positive** correlation between fraction of **adaptive** help seeking prompts and midterm grade (ρ = 0.37, p<0.05, N=38)

Findings—H1a & b:

- Fraction of AI answers with >6 lines of code (proxy for executive help seeking) predicts lower midterm grade
- (N= 82, $R^2 = 0.46$, adj. $R^2 = 0.43$, $F(4, 82) = 17.29$, $p < .001$)



Findings—H3a:

- Significantly more use of the AI tool on days assignments are due
  - 20 sessions vs. 13 (t (70.8)=2.76 p=0.007)
  - 110 prompts vs. 65 (t (64.4)=3.22, p=0.002)
- More answers with long code on due days: 48% vs. 40% ($\chi^2(1) = 79.93$, $p < 0.001$).

Other findings:

- No significant correlation between self-reported use of AI and logged use of AI
- No significant survey predictors of:
  - Overall AI use (sessions or prompts)
  - Detailed AI use
- Only term effects for:
  - Message length (chatbot)
  - Amount of code or proportion of long code

Implications:

- Measurement matters
  - Self-reported AI use is unreliable. Studies must use behavioral logs
  - Aggregate frequency is the wrong metric—type of interaction is what counts.
- Design for verification, not restriction
  - Rather than banning AI, design assignments that require students to engage instrumentally—explain, extend, critique—not just extract solutions.
- Teach the AI orchestration cycle
  - Students need explicit instruction in framing, prompting, interpreting and especially verifying AI outputs.
  - Verification intent is learnable.

Future work:

- Complete interaction coding—determine executive vs. adaptive proportions for more students
- Test intervention: examine differences in use and outcomes for the pedagogic chatbot vs. answer-focused chatbot
- Extend model beyond programming to other knowledge-intensive learning domains

Contact: crowston@syr.edu https://crowston.syr.edu/

Funded in part by a grant from the Alfred P. Sloan Foundation, Grant G-2025-79202

Thanks to Kaitlyn Mann and Cassandra Rivera