

AI-Lab: Instructor Handout

Ethan Dickey, Andres Bejarano, Rhianna Kuperus, Bárbara Fagundes
Purdue University | dickeye@purdue.edu | abejara@purdue.edu

These notes follow our CS251 Spring 2025 prelab, lecture notes, and slides. Change the topic and homework to fit your course. Keep the prelab, lecture, hands-on activity, and reflection together.

Before class

- Pick a course topic where GenAI repeatedly makes mistakes. Try the activity yourself, including follow-up prompts. You need to be comfortable using the tool before teaching students how to use it.
- Prepare the topic lecture and a prompt for the prelab. In CS251, we used Tarjan's algorithm. Students got a short introduction before class, then we taught the algorithm during lecture.
- Decide which homework will require GenAI, what students must submit, and how it will be graded. Update the tool-access instructions and course policy before giving students the materials.

Prelab

Have students start with a few general questions, then ask about something they already know from the course. Once they are comfortable with the interface, give them your prompt for the new topic. This is an introduction. The details come in lecture.

Prelab prompt to adapt

Act as a [subject expert, with the relevant teaching setting]. I want you to give me a short introduction to [topic]. I already know [relevant previous topics]. The learning objectives for this section are as follows: [learning objectives].

Ask students to follow up on anything they don't understand. Tell the tool exactly what they want explained and give it context. They can talk to it as they would to a person.

During lecture (50 minute version)

- 1. Teach the course topic (20 minutes).** Our lecture covers connected components, strongly connected components, and Tarjan's algorithm, including pseudocode and a worked example, before turning to GenAI. We typically assign a short at-home introduction to the topic, usually combined with the prelab so students who haven't used AI before can get more comfortable with it.
- 2. GenAI overview (up to 15 minutes, including prompting).** Start with **3 minutes of open discussion** about what GenAI is useful for. Write down students' answers, then show examples live. Cover conceptual questions, tutoring, practice questions, and test cases. Also cover when not to use it: asking for homework answers without understanding the topic, or believing an answer because it sounds confident.

Show how to prompt and interact. Be clear, concise, and specific. Our lecture breaks a prompt into 2.5 parts:

- **Context:** the expert to act as, the setting, and what the student already knows.
- **Specific request:** what you want it to do, on which topic, and in what form.
- **Goals (the optional 0.5):** where you want the interaction to go. Useful for a larger task, not necessary for every short question.

It's a chatbot. Use the chat. Tell it what you liked or disliked, refine one part at a time, and remind it of the context or start a new chat when needed.

- 3. GenAI experience (at least 15 minutes).** Start by minute 35 at the latest. Assign pairs, have students use their laptops, and give them the activity on the reverse. Call on groups to share what they tried and what went wrong. Finish with homework and course-policy reminders.

Tutoring prompts

"Provide suggestions for how to start without solving my problem."

"Find my bug but don't tell me exactly where it is, help me discover it for myself."

Students need to experience its mistakes themselves, not just watch you point them out.

AI-Lab: In-Class Activity and Postlab

Ethan Dickey, Andres Bejarano, Rhianna Kuperus, Bárbara Fagundes | Purdue University

In-class activity: student instructions

Replace Tarjan's with your selected topic. These are the three choices from our CS251 lecture.

Your turn! Work in pairs.

Choose one of the following options:

1. Have GenAI create a Tarjan's example, run through the steps, and spot its mistakes.
2. Have it create exam practice problems for Tarjan's algorithm with answers. Critique the question, answer, or both.
3. Have it create practice problems for writing pseudocode on a topic from this semester, using one of the algorithms you've been taught.

Be clear, concise, and specific. Talk to it and tell it what you want changed. These aren't meant to be hard or big tasks. Have fun with it. I'll call on groups to share what they did and what mistakes they saw.

Have students work through the response, not just ask whether it is correct. If they find a mistake, encourage them to try fixing it through the conversation. They can also try to find what makes the tool fail or how to prevent the mistake.

Postlab: use GenAI on homework and write a short reflection

Assign a homework task with required GenAI use. Our DSA version required students to use only GenAI until they hit a breaking point, then report and reflect. Set the problem(s), submission details, and grading for your course.

Reflection to include with the assignment

Write one short paragraph about your use of GenAI on [assigned problem(s)]. Include a link to your conversation and address the following:

Which version of GenAI did you use? If it failed, when and why did it fail? If it didn't, how did you prompt it and interact with it so that it gave you a good answer?

Our notes also include a no-use reflection: explain why you didn't use GenAI and describe your solution using course concepts. Use that only on work where GenAI is optional, not the AI-required task.

Course policy

The AI-required assignment is a specific exception, not permission to submit generated answers on other work. Use your own syllabus wording. In CS251, we reminded students:

- If you submit work that isn't your own, it is still cheating. We still evaluate correctness.
- You won't have these tools on exams, so prepare accordingly. You are responsible for your responses, even when the AI gave you something wrong.

Materials and references

Prelab, lecture notes, slides, and the original workshop handout:

cs.purdue.edu/homes/dickeye/#resources

Teaching materials: AI-Lab CS251 S25, by Ethan Dickey, Andres Bejarano, and Chirayu Garg. The prelab was originally written by Chirayu Garg and Ethan Dickey.

Original framework: E. Dickey, A. Bejarano, and C. Garg. AI-Lab: A Framework for Introducing Generative Artificial Intelligence Tools in Computer Programming Courses. SN Computer Science 5, 720 (2024). DOI: 10.1007/s42979-024-03074-y.

Evaluation: E. Dickey, A. Bejarano, R. Kuperus, and B. Fagundes. A Scaffolded GenAI Lab in Early Undergraduate CS: A Mixed-Methods, Multi-Course Evaluation. arXiv:2505.00100.

Please credit the original framework when using these materials. Supported by Purdue Innovation Hub (IH-AI-23002).

Resources



Full paper



arXiv:2505.00100

I'm very interested in hearing how this goes in your course.

Please reach out with questions, and let me know what worked, what didn't, or what you changed.

Ethan Dickey | dickeye@purdue.edu