

## Welcome to AI-Lab: A Framework for Introducing Generative AI Tools in Computer Programming Courses

A streamlined approach for educators to safely introduce GenAI in CS courses.

**Article:** Dickey, Ethan, Andres Bejarano, and Chirayu Garg. "AI-Lab: A Framework for Introducing Generative Artificial Intelligence Tools in Computer Programming Courses." *SN Computer Science* 5.6 (2024): 720.

### Key Steps in the AI-Lab:

- *Instructor Preparation:* Select a topic where GenAI makes mistakes, usually a course topic, review and update course policies, and become comfortable enough using the tool to teach its educational use to students.
- *Prelab Activities:* Conduct pre-surveys, familiarize students with GenAI, and introduce students to the selected topic in a flipped lecture using a carefully tailored prompt to the GenAI tool.
- *In-Class Lab:* Instruct students in best practices for educational use and engage them with a live GenAI demonstration, hands-on exploration (which includes GenAI failure), and group discussion.
- *Post-Lab Reflection:* Students complete homework with required GenAI usage and critical evaluation.

### Practical Examples of In-Class Activities:

- *Debugging Assistance:* Use GenAI to propose fixes for a buggy piece of code, then have students validate and refine the solution. Alternatively, use GenAI to generate buggy code for student evaluation.
- *Concept Reinforcement:* Begin with a lecture topic (e.g., Huffman Coding or Tarjan's) and ask GenAI to introduce the topic. Students compare its explanation with textbook definitions and examples and discuss discrepancies.
- *Problem Generation:* Ask GenAI to generate a set of practice problems on a familiar topic (e.g., for an exam), and have students critique either the question or the answer (or both).

### Tips for Success:

- Most importantly, *the instructor needs to be familiar with the selected GenAI tool*. This requires spending a nontrivial amount of time using the tool. We recommend our GAIDE framework to make that time fruitful.
- Nearly as importantly, *the students must experience its failure for themselves*. This moves notions from, "well, they couldn't do it, but they just couldn't figure it out," to, "even with prompt training, I can't make it work."
- Be specific when crafting prompts. Include the context, desired output, and role of the expert (e.g., "Assume the role of a collegiate professor..."). See the full article for specific recommendations on this.
- Employ iterative feedback: Engage with the tool conversationally—refine its responses just as you would provide feedback to a content development TA.
- Use GenAI as a learning aid rather than a shortcut. Emphasize that while it can offer guidance and feedback, mastering core concepts requires active, critical engagement.
- Encourage students to find where GenAI makes mistakes, or to figure out how to prevent them. Gamify it.

### Frequently Asked Questions:

- *Q: How does AI-Lab mitigate over-reliance on GenAI and help prevent the "Junior-Year Wall"?* A: It combines proper educational use training, deliberate exposure to GenAI's limitations, and reflective discussions to reinforce core programming skills alongside AI use, ensuring students build mastery for advanced coursework.
- *Q: What role does the instructor play in AI-Lab?* A: Instructors select suitable topics, prepare precise prompts, and guide live discussions to maximize both AI benefits and skill development.
- *Q: How specific do AI prompts need to be?* A: The more specific, the better. Include details like course level, topics, and desired format. Be clear and concise in your communication.

**Stay Connected:** Questions? Want to discuss further? Contact us at [dickeye and abejara @purdue.edu]

# AI-Lab Framework for Integrating GenAI in Programming Courses

