

Radar / AI & ML

Teaching Programming in the Age of ChatGPT

How computing instructors plan to adapt to ChatGPT, GitHub Copilot, and other AI coding assistants (ICER 2023 paper)

By [Sam Lau](#) and [Philip Guo](#)

July 18, 2023



Check mark (source: [Pixabay](#))

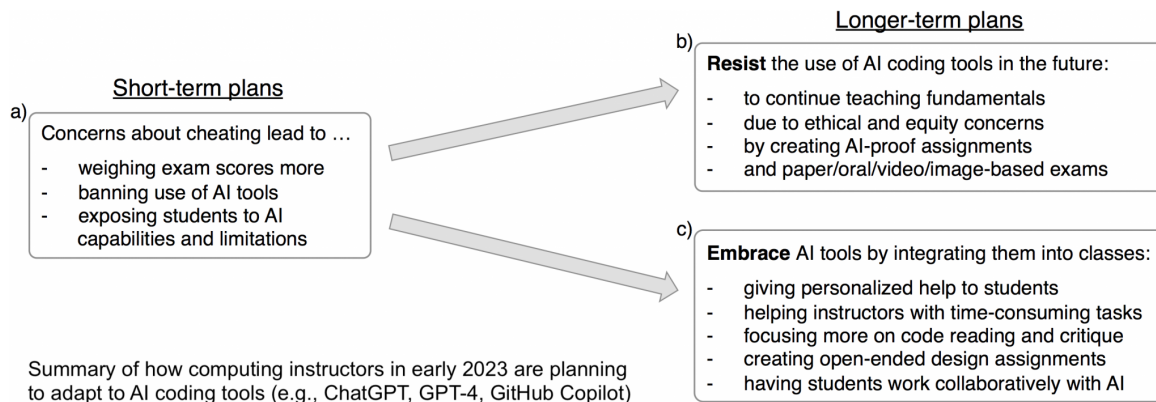
Imagine for a minute that you're a programming instructor who's spent many hours making creative homework problems to introduce your students to the world of programming. One day, a colleague tells you about an AI tool called ChatGPT. To your surprise (and alarm), when you give it your homework problems, it solves most of them perfectly, maybe even better than you can! You realize that by now, AI tools like ChatGPT and GitHub Copilot are good enough to solve all of your class's homework problems and affordable enough that any student can use them. How should you teach students in your classes knowing that these AI tools are widely available?

I'm Sam Lau from UC San Diego, and my Ph.D. advisor (and soon-to-be faculty colleague) Philip Guo and I are presenting a research paper at the International Computing Education Research conference (ICER) on this very topic. We wanted to know:

How are computing instructors planning to adapt their courses as more and more students start using AI coding assistance tools such as ChatGPT and GitHub Copilot?

To answer this question, we gathered a diverse sample of perspectives by interviewing 20 introductory programming instructors at universities across 9 countries (Australia, Botswana, Canada, Chile, China, Rwanda, Spain, Switzerland, United States) spanning all 6 populated continents. To our

Here's a summary of our findings:



Short-Term Plans: Instructors Want to Stop Students from Cheating

Even though we didn't specifically ask about cheating in our interviews, all of the instructors we interviewed mentioned it as a primary reason to make changes to their courses in the short term. Their reasoning was: If students could easily get answers to their homework questions using AI tools, then they won't need to think deeply about the material, and thus won't learn as much as they should. Of course, having an answer key isn't a new problem for instructors, who have always worried about students copying off each other or online resources like Stack Overflow. But AI tools like ChatGPT generate code with slight variations between responses, which is enough to fool most plagiarism detectors that instructors have available today.

The deeper issue for instructors is that if AI tools can easily solve problems in introductory courses, students who are learning programming for the first time might be led to believe that AI tools can correctly solve any programming task, which can cause them to grow overly reliant on them. One instructor described this as not just cheating, but "cheating badly" because AI tools generate code that's incorrect in subtle ways that students might not be able to understand.

To discourage students from becoming over-reliant on AI tools, instructors used a mix of strategies, including making exams in-class and on-paper, and also having exams count for more of students' final grades. Some instructors also explicitly banned AI tools in class, or exposed students to the limitations of AI tools. For example, one instructor copied old homework questions into ChatGPT as a live demo in a lecture and asked students to critique the strengths and weaknesses of the AI-generated code. That said, instructors

interviewed them for our study.

Longer-Term Plans (Part 1): Ideas to Resist AI Tools

In the next part of our study, instructors brainstormed many ideas about how to approach AI tools longer-term. We split up these ideas into two main categories: ideas that resist AI tools, and ideas that embrace them. Do note that most instructors we interviewed weren't completely on one side or the other—they shared a mix of ideas from both categories. That said, let's start with why some instructors talked about resisting AI tools, even in the longer term.

The most common reason for wanting to resist AI tools was the concern that students wouldn't learn the fundamentals of programming. Several instructors drew an analogy to using a calculator in math class: using AI tools could be like, in the words of one of our interview participants, "giving kids a calculator and they can play around with a calculator, but if they don't know what a decimal point means, what do they really learn or do with it? They may not know how to plug in the right thing, or they don't know how to interpret the answer." Others mentioned ethical objections to AI. For example, one instructor was worried about recent lawsuits around Copilot's use of open-source code as training data without attribution. Others shared concerns over the training data bias for AI tools.

To resist AI tools practically, instructors proposed ideas for designing "AI-proof" homework assignments, for example, by using a custom-built library for their course. Also, since AI tools are typically trained on U.S./English-centric data, instructors from other countries thought that they could make their assignments harder for AI to solve by including local cultural and language context (e.g. slang) from their countries.

Instructors also brainstormed ideas for AI-proof assessments. One common suggestion was to use in-person paper exams since proctors could better ensure that students were only using paper and pencil. Instructors also mentioned that they could try oral exams where students either talk to a course staff member in-person, or record a video explaining what their code does. Although these ideas were first suggested to help keep assessments meaningful, instructors also pointed out that these assessments could actually improve pedagogy by giving students a reason to think more deeply about why

Longer-Term Plans (Part 2): Ideas to Embrace AI Tools

Another group of ideas sought to embrace AI tools in introductory programming courses. The instructors we interviewed mentioned several reasons for wanting this future. Most commonly, instructors felt that AI coding tools would become standard for programmers; since "it's inevitable" that professionals will use AI tools on the job, instructors wanted to prepare students for their future jobs. Related to this, some instructors thought that embracing AI tools could make their institutions more competitive by getting ahead of other universities that were more hesitant about doing so.

Instructors also saw potential learning benefits to using AI tools. For example, if these tools make it so that students don't need to spend as long wrestling with programming syntax in introductory courses, students could spend more time learning about how to better design and engineer programs. One instructor drew an analogy to compilers: "We don't need to look at 1's and 0's anymore, and nobody ever says, 'Wow what a big problem, we don't write machine language anymore!' Compilers are already like AI in that they can outperform the best humans in generating code." And in contrast to concerns that AI tools could harm equity and access, some instructors thought that they could make programming less intimidating and thus more accessible by letting students start coding using natural language.

Instructors also saw many potential ways to use AI tools themselves. For example, many taught courses with over a hundred students, where it would be too time-consuming to give individual feedback to each student. Instructors thought that AI tools trained on their class's data could potentially give personalized help to each student, for example by explaining why a piece of code doesn't work. Instructors also thought AI tools could help generate small practice problems for their students.

To prepare students for a future where AI tools are widespread, instructors mentioned that they could spend more time in class on code reading and critique rather than writing code from scratch. Indeed, these skills could be useful in the workplace even today, where programmers spend significant amounts of time reading and reviewing other people's code. Instructors also thought that AI tools gave them the opportunity to give more open-ended assignments, and even have students collaborate with AI directly on their work, where an assignment would ask students to generate code using AI and then iterate on the code until it was both correct and efficient.

instructors are just starting to form opinions about this fast-growing phenomenon but have not yet converged to any consensus about best practices. Using these findings as inspiration, we synthesized a diverse set of open research questions regarding how to develop, deploy, and evaluate AI coding tools for computing education. For instance, what mental models do novices form both about the code that AI generates and about how the AI works to produce that code? And how do those novice mental models compare to experts' mental models of AI code generation? (Section 7 of our paper has more examples.)

We hope that these findings, along with our open research questions, can spur conversations about how to work with these tools in effective, equitable, and ethical ways.

Check out our paper here and email us if you'd like to discuss anything related to it!

[From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot.](#) Sam Lau and Philip J. Guo. ACM Conference on International Computing Education Research (ICER), August 2023.

Post topics: [AI & ML](#), [Artificial Intelligence](#)

Post tags: [Research](#)

Share: [Tweet](#) [Share](#) [Share](#)

ABOUT O'REILLY

[Teach/write/train](#)

[Careers](#)

[Press releases](#)

[Media coverage](#)

[SIGN IN](#)[Try Now](#)[Diversity](#)[O'Reilly for marketers](#)

SUPPORT

[Contact us](#)[Newsletters](#)[Privacy policy](#)

INTERNATIONAL

[Australia & New Zealand](#)[Hong Kong & Taiwan](#)[India](#)[Indonesia](#)[Japan](#)

DO NOT SELL MY PERSONAL INFORMATION



© 2023, O'Reilly Media, Inc. All trademarks and registered trademarks appearing on oreilly.com are the property of their respective owners.

[Terms of service](#) • [Privacy policy](#) • [Editorial independence](#)



WATCH ON YOUR BIG SCREEN

View all O'Reilly videos, Superstream events, and Meet the Expert sessions on your home TV.

