

Post:
GenAI and how can we address it in our teaching!

Isaac Alpizar-Chacon

July, 2024

<https://isaacalpizar.info/posts/2024/07/genAI-teachers-1/>

GenAI is here to stay! We need to explicitly address what is and isn't allowed, teach about AI's technical and ethical aspects, and change our teaching practices.

In one of the first papers about GenAI (Generative AI) in Computing Education (CED), Finnie-Ansley et al. [7] stated, “we cannot put the genie back in the bottle!” As teachers, we cannot deny that GenAI is here and will impact our teaching and students' learning. We are responsible for explicitly addressing GenAI and adapting our courses accordingly. As teachers, we need to take action. Many challenges and opportunities of GenAI in CED have already been identified, such as increased productivity, better understanding of code and concepts, overreliance, and concerns about academic integrity ([1, 5, 12]). We need to implement changes to mitigate these challenges and leverage the opportunities!

What can we do as teachers?

1. **Become familiar with GenAI tools:** If you haven't already, take the time to explore a range of GenAI tools. Test their functionality by applying them to your course assignments and exams. This will help you understand both their capabilities and limitations and how easily (or not) they can solve the tasks you set for your students. Refs. [10].
2. **Set clear GenAI policies:** Explicitly state what is allowed and what isn't in your course. Explain to students why you have chosen these policies. Review the learning objectives of your course and determine if they can be achieved using GenAI. Refs. [8, 10].
3. **Teach technical and ethical aspects of AI:** Students need to understand how GenAI models work to reason about their capabilities and limitations. This AI literacy can also be coordinated at the program level.
4. **Adapt your teaching practices:** Adjusting to GenAI can be done defensively, incorporatively, or by combining both. Shortly, accreditation agencies or evaluation committees may require evidence of how your program addresses GenAI. This includes demonstrating how exams and other assessments are designed to be LLM-proof or how GenAI tools are integrated into courses to enhance learning. Refs [2, 4].
5. **Update your assessment methods:** Assume students are using GenAI. Adjust grading components by assigning less weight to tasks easily solved with GenAI. Consider new assignments like oral exams, presentations, group projects, and closed testing. Refs. [9, 10].
6. **Explore new teaching practices:** Use GenAI to generate novel exercises, create sample solutions, to explain concepts, etc. At Utrecht University, we have compiled a set of practices for integrating GenAI in Programming Education. Available at <https://www.uu.nl/en/research/generative-ai-for-computing-education>
7. **Incorporate assignments on GenAI tools:** Design tasks that teach the limitations and capabilities of GenAI tools and assignments that show when and how to use these tools effectively.

8. **Focus on new skills for working with GenAI:** Emphasize skills like program specification, refactoring, prompt engineering, verification/testing, and decomposition/planning. Refs. [5].
9. **Test new tools developed with LLMs:** New tools are being developed with LLMs in their code. Explore tools for feedback generation [11], developing code comprehension [6], creating Parson’s problems [3], and explaining concepts within Jupyter notebooks (I am working on this project with some students).

These are basic actions that we can start implementing as teachers. GenAI is changing rapidly, and we need to continuously adjust our practices. Also, evaluate your changes. See how effective they are, and take into account students’ opinions.

Note 1: Spanish & Dutch versions of this post will be available in the coming weeks.

Note 2: This post is based on my participation in the panel discussion on Generative AI in SE Education at the SEN Symposium 2024 a few weeks ago¹.

References

- [1] B. A. Becker, P. Denny, J. Finnie-Ansley, A. Luxton-Reilly, J. Prather, and E. A. Santos. Programming is hard-or at least it used to be: Educational opportunities and challenges of ai code generation. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 500–506, 2023.
- [2] J. Berrezueta-Guzman and S. Krusche. Recommendations to create programming exercises to overcome chatgpt. In *2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T)*, pages 147–151. IEEE, 2023.
- [3] A. del Carpio Gutierrez, P. Denny, and A. Luxton-Reilly. Automating personalized parsons problems with customized contexts and concepts. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, pages 688–694. 2024.
- [4] P. Denny, J. Leinonen, J. Prather, A. Luxton-Reilly, T. Amarouche, B. A. Becker, and B. N. Reeves. Prompt problems: A new programming exercise for the generative ai era. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, pages 296–302, 2024.
- [5] P. Denny, J. Prather, B. A. Becker, J. Finnie-Ansley, A. Hellas, J. Leinonen, A. Luxton-Reilly, B. N. Reeves, E. A. Santos, and S. Sarsa. Computing education in the era of generative ai. *Communications of the ACM*, 67(2):56–67, 2024.
- [6] P. Denny, D. H. Smith IV, M. Fowler, J. Prather, B. A. Becker, and J. Leinonen. Explaining code with a purpose: An integrated approach for developing code comprehension and prompting skills. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, pages 283–289. 2024.
- [7] J. Finnie-Ansley, P. Denny, B. A. Becker, A. Luxton-Reilly, and J. Prather. The robots are coming: Exploring the implications of openai codex on introductory programming. In *Proceedings of the 24th Australasian Computing Education Conference*, pages 10–19, 2022.
- [8] Y. Jin, L. Yan, V. Echeverria, D. Gašević, and R. Martinez-Maldonado. Generative ai in higher education: A global perspective of institutional adoption policies and guidelines. *arXiv preprint arXiv:2405.11800*, 2024.
- [9] S. Lau and P. Guo. 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. In *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1*, pages 106–121, 2023.

¹<https://www.sen-symposium.nl/program/>

- [10] J. Prather, P. Denny, J. Leinonen, B. A. Becker, I. Albluwi, M. Craig, H. Keuning, N. Kiesler, T. Kohn, A. Luxton-Reilly, et al. The robots are here: Navigating the generative ai revolution in computing education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*, pages 108–159. 2023.
- [11] L. Roest, H. Keuning, and J. Jeuring. Next-step hint generation for introductory programming using large language models. In *Proceedings of the 26th Australasian Computing Education Conference*, pages 144–153, 2024.
- [12] C. Zastudil, M. Rogalska, C. Kapp, J. Vaughn, and S. MacNeil. Generative ai in computing education: Perspectives of students and instructors.(2023). *arXiv preprint arXiv:2308.04309*, 2023.