How to shape Computer Science Education in the AI Era?

Bridging Technology, Humanities, and Inspiring the Desire to Learn.

Paul-Antoine Bisgambiglia, Marie-Laure Nivet, Evelyne Vittori
SISU, UMR CNRS 6134, SPE Laboratory
University of Corsica
Corte, France
bisgambiglia_pa@univ-corse.fr, nivet_m@univ-corse.fr, vittori_e@univ-corse.fr

Given the rapid growth of generative AI, Undone Science could fundamentally question our educational role in light of a tool for which it is really difficult to define progress and limits. This question can be seen as a near incursion on the side of science fiction, where individuals must coexist with a ”knowing entity”. How can individuals sustain their motivation to acquire knowledge, find their place, cultivate personal growth, discover their purpose, encourage the development of a critical mind, establish and maintain autonomy and independence? Considering that reflection is always more difficult in the absolute, we, as computer scientists, decided to focus on teaching programming and computational thinking in higher education.

Some experts predict the end of programming as we know it today [Wel23]. Many others wonder about the impact of specialised tools such as Copilot and generative AI on the learners and professional developers who are already using them [Ove23]. Either way, there seems to be no doubt that the developer profession is about to change radically. Faced with this evidence, it is our responsibility to ask ourselves about the nature and modalities that must guide computing curricula in the future.

If we hypothesise that generative AI will become increasingly powerful and able to solve most of the programming tasks we actually give to students, what about the need to learn programming? More generally, what about the need to learn? What skills will be needed? How to teach in this context? How - from the students' point of view - can the desire to learn, the motivation, be maintained?

Historically, the teaching of programming skills has followed technological advances. From the first imperative languages, teaching methods and tools have reflected the evolution of the sector. The developer, who could be considered a code maker, has become a technical solution architect. He plays an increasing role in the design and implementation of complex systems. Search engines marked a turning point. The teacher has become a guide rather than a repository of knowledge. The increased accessibility of information has allowed for greater autonomy for students, but also their tendency to ”copy/paste” without real understanding.

It has long been clear that the teaching of programming needs to focus on understanding concepts and methods. It is a matter of computational thinking rather than simply learning programming skills. Despite this, there is no real consensus on what these concepts and methodologies should be [Bel22], [LSd+20]. Our task is to educate and shape students' minds in a technological context that is constantly changing. There will always be emerging languages, as well as economic, environmental, social and ethical challenges. Our students must constantly adapt. Learning to learn is more important than ever.

There is no doubt that we are entering a new dimension with generative AI. It could be related to the questions, pitfalls and controversies surrounding the use of calculators in education when it was introduced in schools [Suy80], [Ban11] even on a smaller scale.

AI improves productivity and helps with some tasks. They can help students produce code and text quickly. They can support learning by providing immediate feedback on their work. Teachers can also use them to prepare their courses and exercises, grade or create assignments, or summarise articles [LG23].

However, these tools are not perfect. Among many limitations, they make mistakes, ”hallucinations”, most of them are unable to cite their sources, they are biased and do not understand the concepts they manipulate [RTT23], [MPF+23]. Far from being negative, these imperfections can be viewed as an educational asset. They can be used as a starting point to educate the student's critical mind. Knowing the shortcomings of these tools, students are encouraged to be careful and attentive to the proposals in order to validate and improve them [UNE23]. In this context, students must have acquired the minimum knowledge necessary to analyse the proposed solutions.

If the technology were to be halted in its development, we could reasonably consider incorporating generative AI as a novel pedagogical tool that does not challenge the significance of fundamental skills. What if, tomorrow, the proposed solutions became ”perfect” in the sense of ”without error”? The advantages of the current limitations could disappear. What will students have to learn if AI can immediately give good answers and optimal code with respect to a given criterion? What will be the new basic skills in computer sciences?

Considering all these facts, it seems that a major rethink is needed. As scholars and educators in tertiary education, it is imperative that we collaborate to determine the essential skills necessary for the forthcoming computational thinking foundation [LSd+20]. While these core skills will necessarily be part of traditional computing courses, they will need to be complemented by a broader knowledge base.

In fact, in a world where technology has such a huge impact on our daily lives, the role of the developer cannot be limited to their technological domain bubble. Computer scientists need to be educated to understand not only how to build systems, but also why they should build them and with what kind of impact. We need to make students aware of the social, political and cultural consequences of their actions. They need to be active and aware citizens, with a sharp critical mind. Interdisciplinary cooperation will be essential. The integration of courses in the humanities and social sciences into computing curricula should enable students to gain a holistic view of their work. Philosophy, ethics, sociology, psychology, but also law courses are areas that will enrich the education of computer scientists. They could be open to the tools of these external disciplines, providing them with the means to question and contextualise their practices. Nor will we be able to avoid talking about the environmental impact of these AI solutions and, more generally, of technology.

The question of pedagogical approaches also seems essential. How will the psychology of learning be affected by the use of AI tools? Will the traditional trial-and-error approach always be relevant [MRMG22]? How to encourage effort, which seems to be essential for the growth of motivation and self-esteem [JSH+22]?

Even if "perfect" AI is still science fiction, we believe that it is crucial to anticipate it in order not to react, as usual, to a technological evolution imposed by digital companies. We must be prepared for this evolution in order to best train the next generation of computer scientists, who will play a fundamental role in the next 50 years in balancing social mutations. This reflection questions our role as academics, but also as citizens or, more generally, as social beings in a world in which we will live continuously with an AI that could be of superior reliability. So what is an infallible AI? Who should and who can define it? How can the public education and research system, everywhere in the world, help to define what must be, what should be the nature of these tools? All of these fundamental questions require a real interdisciplinary approach, which we have been dreaming of for so long. We will have to address the ethical use of AI, its energy and environmental impact, as well as its societal impact, when defining computing curricula. Finally, our role will be to educate autonomous and enlightened citizens cultivating their desire to learn.

To conclude, we would like to call on the educational science community, philosophers, ethicists, social scientists, neuroscientists, economists, digital sobriety experts and more generally all colleagues interested in working together to define the knowledge, skills and teaching methods needed to train the next cohort of technology developers in the era of AI.

References

[Ban11] Sarah Banks. *A Historical Analysis of Attitudes Toward the Use of Calculators in Junior High and High School Math Classrooms in the United States Since 1975.* Cedarville University, 2011.

[Bel22] Héctor Belmar. Review on the teaching of programming and computational thinking in the world. *Frontiers in Computer Science*, 4, 2022.

[JSH+22] Huw Jarvis, Isabelle Stevenson, Amy Q. Huynh, Emily Babbage, James Coxon, and Trevor T.-J. Chong. Effort Reinforces Learning. *J. Neurosci.*, 42(40):7648–7658, October 2022. Publisher: Society for Neuroscience Section: Research Articles.

[LG23] Sam Lau and Philip 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 V.1*, pages 106–121, Chicago IL USA, August 2023. ACM.

[LSd+20] Yeping Li, Alan H. Schoenfeld, Andrea A. diSessa, Arthur C. Graesser, Lisa C. Benson, Lyn D. English, and Richard A. Duschl. Computational Thinking Is More about Thinking than Computing. *Journal for STEM Educ Res*, 3(1):1–18, April 2020.

[MPF+23] Kamil Malinka, Martin Peresíni, Anton Firc, Ondrej Hujnák, and Filip Janus. On the Educational Impact of ChatGPT: Is Artificial Intelligence Ready to Obtain a University Degree? In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, ITiCSE 2023, pages 47–53, New York, NY, USA, June 2023. Association for Computing Machinery.

[MRMG22] Yeray Mera, Gabriel Rodríguez, and Eugenia Marin-Garcia. Unraveling the benefits of experiencing errors during learning: Definition, modulating factors, and explanatory theories. *Psychon Bull Rev*, 29(3):753–765, June 2022.

[Ove23] Stack Overflow. Stack Overflow Developer Survey 2023, 2023.

[RTT23] Jürgen Rudolph, Samson Tan, and Shannon Tan. ChatGPT: Bullshit spewer or the end of traditional assessments in higher education? *Journal of Applied Learning and Teaching*, 6(1), January 2023. Number: 1.

[Suy80] Marilyn N. Suydam. International Calculator Review: Working Paper on Hand-Held Calculators in Schools. Technical report, Information Reference Center (ERIC/IRC), The Ohio State Univ, March 1980. ERIC Number: ED190408.

[UNE23] UNESCO. Guidance for generative AI in education and research - UNESCO Bibliothèque Numérique. Technical report, UNESCO, 2023.

[Wel23] Matt Welsh. The End of Programming. *Commun. ACM*, 66(1):34–35, January 2023.