ARCHIVIO ISTITUZIONALE DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Teaching Programming in the Age of Generative AI

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Martini, S. (2024). Teaching Programming in the Age of Generative AI. New York, NY : Association for Computing Machinery [10.1145/3649217.3653527].

Availability:
This version is available at: https://hdl.handle.net/11585/973399 since: 2024-07-04

Published:

DOI: http://doi.org/10.1145/3649217.3653527

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

31 December 2024

Teaching Programming in the Age of Generative AI

Simone Martini
University of Bologna
Bologna, Italy
simone.martini@unibo.it

Abstract

Programming has been considered the "essence of informatics"[5] since the beginning of computing as a discipline. But programming in the fifties was very different from what we know today, and one of the goals (or dreams) throughout the history of programming language technology, has been "automatic programming" [9]—the ability to automatically generate computer code starting from a high(er)-level description of the specification of that code. What this meant changed over the years, from punching paper tape [3], to compiling high-level programming languages [14], to program synthesis [6].

Today, however, the availability of machine learning artefacts that produce high-level code from natural language specifications has completely changed the traditional meaning. To the extent that some computer scientists have begun to question the received wisdom that the core of their discipline is deeply rooted in programming [19].

If programming and programming languages are no longer the essence of computer science, this changes the epistemology of the discipline itself. Moreover, if we are at the end of programming, we should also change the curriculum, where programming, algorithms and programming languages play a major role. Several recent papers reviewed the performance of code generators based on large language models on typical CS1 problems (e.g., from the many possible citations [2, 7, 13]) and how machine learning impacts K-12 teaching (e.g., [16, 18]).

Starting from this data, I will argue for the role of programming in the curriculum, distinguishing between programming taught as part of a holistic curriculum (as in some non-technical high schools) or as a vocational tool. I will use Simondon's notion of (closed and open) technical object [17] as an interpretive lens, together with Calvino's reflections on the availability of writing machines capable of replacing the poet and the author [4].

CCS Concepts

• **Social and professional topics** → **Computer science education**; *History of computing*; • **Computing methodologies** → **Artificial intelligence**; *Machine learning*.

Keywords

large language models, epistemology, programming

ITiCSE 2024, July 8–10, 2024, Milan, Italy
© 2024 Copyright held by the owner/author(s).
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024), July 8–10, 2024, Milan, Italy*, https://doi.org/10.1145/3649217.3653527.

ACM Reference Format:
Simone Martini. 2024. Teaching Programming in the Age of Generative AI. In *Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024), July 8–10, 2024, Milan, Italy.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3649217.3653527

Biography

Simone Martini (Ph.D., Pisa, 1987) is Professor of Computer Science at the University of Bologna. He has been a visiting scientist at the former Systems Research Center of Digital Equipment Corporation, Palo Alto; at Stanford University; at the École normale supérieure, Paris; at the Université Paris 13; at the University of California at Santa Cruz; and at the Collegium - Lyon Institute for Advanced Studies. The author of a textbook on the principles of programming languages [8], his research has focused on the foundations of programming languages for several decades. After joining the Commission on the History and Philosophy of Computing (HaPoC), his interests shifted more towards the epistemology and history of programming languages, and towards computer science education. He is particularly interested in topics at the interface between CS education and programming languages [15], mathematics [1, 10, 12], and epistemology [11].

Acknowledgments

My participation in the Commission on the History and Philosophy of Computing, and in particular in the French ANR project PROGRAMme led by Liesbeth De Mol (CNRS, Lille), has been a constant source of inspiration and enlightenment.

References

[1] Evmorfia-Iro Bartzia, Michael Lodi, Marco Sbaraglia, Simon Modeste, Viviane Durand-Guerrier, and Simone Martini. 2024. An Unplugged Didactical Situation on Cryptography between Informatics and Mathematics. *Informatics in Education* 23, 1 (2024), 25–56. https://doi.org/10.15388/infedu.2024.06

[2] Carlo Bellettini, Michael Lodi, Violetta Lonati, Mattia Monga, and Anna Morpurgo. 2023. DaVinci Goes to Bebras: A Study on the Problem Solving Ability of GPT-3. In *CSEDU 2023 - 15th International Conference on Computer Supported Education*, Vol. 2. SCITEPRESS - Science and Technology Publications, Prague, 59–69. https://doi.org/10.5220/0012007500003470

[3] Corrado Böhm. 1954. Calculatrices digitales. Du déchiffrage des formules logico-mathématiques par la machine même dans la conception du programme. *Annali di matematica pura e applicata* IV-37, 1 (1954), 1–51.

[4] Italo Calvino. 1967. Cibernetica e fantasmi (Cybernetics and ghosts). In *The Uses of Literature*. Harcourt and Co, 1986, San Diego. Lecture delivered in several Italian and European cities, November 1967.

[5] Michael E. Caspersen. 2023. Principles of Programming Education. In *Computer Science Education: Perspectives on Teaching and Learning, 2nd edition*, S. Sentance, E. Barendsen, N.R. Howard, and C. Schulte (Eds.). Bloomsbury Publishing, London, 219–236.

[6] Pierre Cointe. 2005. Towards Generative Programming. In *Unconventional Programming Paradigms*, Jean-Pierre Banâtre, Pascal Fradet, Jean-Louis Giavitto, and Olivier Michel (Eds.). Springer, Berlin, Heidelberg, 315–325.

[7] James Finnie-Ansley, Paul Denny, Brett A. Becker, Andrew Luxton-Reilly, and James Prather. 2022. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In *Proceedings of the 24th Australasian Computing Education Conference* (Virtual Event, Australia) *(ACE '22)*.

Association for Computing Machinery, New York, NY, USA, 10–19. https://doi.org/10.1145/3511861.3511863

[8] Maurizio Gabbrielli and Simone Martini. 2023. *Programming Languages: Principles and Paradigms, 2nd edition.* Springer, Cham.

[9] Saul Gorn. 1949(?). *Is Automatic Programming Feasible?* Technical Report. Aberdeen Proving Ground. Quoted by D.L. Parnas, Software aspects of strategic defense systems, CACM 28(12), 1985.

[10] Michael Lodi, Maria Cristina Carrisi, and Simone Martini. 2024. Big Ideas of Cryptography in Primary School. In *Proceedings of the 29th ACM Conference on Innovation and Technology in Computer Science Education Vol. 1* (Milan, Italy) *(ITiCSE '24)*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3649217.3653548

[11] Michael Lodi and Simone Martini. 2021. Computational Thinking, Between Papert and Wing. *Science & Education* 30, 4 (2021), 883–908. https://doi.org/10.1007/s11191-021-00202-5

[12] Michael Lodi, Marco Sbaraglia, and Simone Martini. 2022. Cryptography in Grade 10: Core Ideas with Snap! and Unplugged. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 1* (Dublin, Ireland) *(ITiCSE '22)*. Association for Computing Machinery, New York, NY, USA, 456–462. https://doi.org/10.1145/3502718.3524767

[13] James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 2023. 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* (Turku, Finland) *(ITiCSE-WGR '23)*. Association for Computing Machinery, New York, NY, USA, 108–159. https://doi.org/10.1145/3623762.3633499

[14] C. Rich and R.C. Waters. 1988. Automatic programming: myths and prospects. *Computer* 21, 8 (1988), 40–51. https://doi.org/10.1109/2.75

[15] Marco Sbaraglia, Michael Lodi, and Simone Martini. 2022. A Necessity-Driven Ride on the Abstraction Rollercoaster of CS1 Programming. *Informatics in Education* 20, 4 (2022), 641–682. https://doi.org/10.15388/infedu.2021.28

[16] R. Benjamin Shapiro, Rebecca Fiebrink, and Peter Norvig. 2018. How machine learning impacts the undergraduate computing curriculum. *Commun. ACM* 61, 11 (oct 2018), 27–29. https://doi.org/10.1145/3277567

[17] Gilbert Simondon. 1958. *Du mode d'existence des objets techniques.* Aubier, Paris.

[18] Matti Tedre, Tapani Toivonen, Juho Kahila, Henriikka Vartiainen, Teemu Valtonen, Ilkka Jormanainen, and Arnold Pears. 2021. Teaching Machine Learning in K-12 Classroom: Pedagogical and Technological Trajectories for Artificial Intelligence Education. *IEEE Access* 9 (2021), 110558–110572. https://api.semanticscholar.org/CorpusID:237000440

[19] Matt Welsh. 2022. The End of Programming. *Commun. ACM* 66, 1 (dec 2022), 34–35. https://doi.org/10.1145/3570220