

The Influence of GenAI on the Evaluation of Computer Programming Students in Higher Education

Teresa Terroso ✉ 

ID+, ESMAD, Polytechnic Institute of Porto, Portugal

Mário Pinto ✉ 

ID+, ESMAD, Polytechnic Institute of Porto, Portugal

Abstract

Artificial Intelligence (AI) has assumed an increasingly prominent role in education, transforming the dynamics of teaching and learning while introducing new pedagogical opportunities and challenges. In computer programming education, generative AI tools have had a particularly profound impact. Historically, computer programming education has emphasized problem-solving skills, syntax accuracy, and code efficiency. However, the emergence of generative AI models capable of supporting automatic code generation, producing high-quality code snippets and entire programs, personalized explanations and tutoring, and real-time debugging, has triggered a paradigm shift. These tools make learning processes and assessment less effective and less clear about students' true knowledge. In this context, the paper explores three key dimensions: the broader impact of AI in education, the new challenges that AI presents in teaching and learning computer programming in higher education, and the implications for student assessment, an essential element of the educational process. To investigate these topics, we conducted an online survey targeting Portuguese higher education instructors teaching programming-related courses. Our primary objective was to understand the changes introduced in evaluation methods and criteria due to the growing use of generative AI tools, particularly those focused on code generation.

2012 ACM Subject Classification Computing methodologies; Applied computing → Computer-managed instruction

Keywords and phrases Generative Artificial Intelligence, Computer Programming Education, Student Assessment, Teaching and Learning, Higher Education

Digital Object Identifier 10.4230/OASICS.ICPEC.2025.18

1 Introduction

AI is rapidly transforming educational practices, offering capabilities ranging from intelligent tutoring systems to adaptive learning management platforms. These technologies enable personalized instruction by adapting the content and pace to each student's profile [19, 9, 3]. Generative AI encompasses systems capable of producing content, including text, images, video, and source code, as well as solving complex logical problems. In programming education, these tools are significantly shifting student's behavior and reshaping the development of skills such as logical reasoning, algorithmic problem-solving, and proficiency in programming languages [1]. Platforms like GitHub Copilot, ChatGPT, and OpenAI's Codex can generate code, explain complex programming concepts, and assist in debugging, often in real time [17]. This paper analyzes the pedagogical implications of generative AI tools in computer programming teaching and learning, with a particular focus on their influence on student's evaluation. It addresses the following research questions: (i) Which generative AI tools are most commonly used by students? (ii) What changes have these tools already brought to the evaluation process? (iii) What mitigation strategies or pedagogical approaches are educators adopting? and (iv) What challenges and concerns do teachers face in assessing



© Teresa Terroso and Mário Pinto;
licensed under Creative Commons License CC-BY 4.0

6th International Computer Programming Education Conference (ICPEC 2025).

Editors: Ricardo Queirós, Mário Pinto, Filipe Portela, and Alberto Simões; Article No. 18; pp. 18:1–18:8

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

students in this new context? The structure of the paper is as follows: section two describes the use of generative AI tools in computer programming learning. Section three provides a brief literature review on the assessment of programming skills in the age of AI. Section four outlines the survey methodology and questionnaire design, followed by results presentation and their analysis, and discussion. Finally, the last section presents some conclusions, key contributions, and directions for future research.

2 The Role of Generative AI Tools in Computer Programming Learning

The use of AI in software development has significantly accelerated with the rise of generative AI tools. Powered by large language models (LLMs), tools such as GitHub Copilot and ChatGPT are designed to understand prompts and generate relevant source code in real time [3]. GitHub Copilot, for instance, integrates directly into development environments, offering context-aware code suggestions. These capabilities challenge the validity of traditional take-home programming assessments, as they allow students to generate entire solutions with minimal effort. While AI-generated code often meets assignment requirements, it may lack well-structured design, clear documentation, or optimization, raising critical questions about what competencies are truly being assessed. Generative AI tools are currently being used in several stages of the software development lifecycle, including [4, 15, 14]:

- **Code Completion and Generation:** tools such as GitHub Copilot suggest real-time code snippets, reducing repetitive coding and accelerating development;
- **Bug Detection and Correction:** AI models can detect syntax and logic errors and propose fixes, often integrated into modern IDEs;
- **Documentation and Explanation:** LLMs can generate human-readable documentation, aiding code comprehension and collaboration;
- **Learning and Tutoring:** these tools act as intelligent tutors, providing beginners with example-driven learning and concept explanations

Despite these advantages, such as faster development, improved code quality, and a reduced learning curve for beginners, serious pedagogical challenges have emerged, particularly in foundational programming education [13, 18]:

- **Passive attitude to understanding and learning basic concepts:** beginners may rely heavily on AI-generated solutions, bypassing essential learning steps such as understanding syntax, logic, and debugging. This attitude can hinder the development of core programming skills;
- **Superficial Understanding of Programming Concepts:** generative AI often presents solutions without explaining the underlying reasoning, leading to a “black box” perception of programming. Students may adopt these outputs without truly grasping the principles involved;
- **Difficulties in debugging and handling code errors:** when AI-generated code fails, students unfamiliar with debugging practices often struggle to diagnose and resolve issues. Generative tools rarely provide detailed walk-throughs of their suggestions, impeding students’ ability to troubleshoot independently;
- **Problems related to ethics and plagiarism:** the boundary between legitimate assistance and academic dishonesty is increasingly blurred. Instructors face difficulty distinguishing original student work from AI-assisted submissions, raising concerns about plagiarism and assessment fairness;

- Difficulties in understanding and adapting the code: AI-generated code is not always correct or efficient. Beginners may lack the critical skills needed to assess or adapt the generated output, which can reinforce misconceptions and bad practices.

3 The Impact of Generative AI Tools on the Evaluation Process

While generative AI tools offer promising possibilities for enhancing personalized learning and supporting the development of more authentic, dynamic assessments, they also raise significant concerns related to academic integrity, equity, and the loss of critical thinking and creativity in student work. A primary concern is academic dishonesty. Students may use AI-generated content to complete assignments with minimal personal input, effectively bypassing the learning process. A recent EDUCAUSE survey [12] reported that more than 60% of faculty teachers observed an increase in AI-generated student submissions, intensifying concerns about plagiarism and authorship. Although tools like Turnitin's AI detector have been introduced, their accuracy and reliability remain under debate. Research on the integration of generative AI into assessment processes is still emerging. Although many studies highlight the potential of generative AI in education, relatively few provide empirical evidence on its influence on evaluation practices [8]. According to a recent study by Huber et al. [5], both educators and students recognize the profound impact of AI on traditional assessments, particularly in fields such as essay writing and coding. Although many educators advocate adapting assessment strategies to leverage AI in fostering higher-order thinking, students express mixed feelings, often concerned about potential losses in creativity and originality. According to some authors [7], more than 50% of students enrolled in programming courses admitted to regularly using AI tools, with more than 30% employing them for graded assignments. This raises challenges for instructors, as AI-generated code is often indistinguishable from human-written code, especially when slightly modified. To address these issues, researchers such as Kizilcec and Elaine Huber [5] propose a shift toward hybrid and process-oriented assessment models. These may include:

- Live coding exams, to evaluate real-time problem-solving and syntax fluency under supervision;
- Monitoring the code development process, through screen recordings or automated trace logs;
- Code explanation sessions, requiring students to justify and analyze their submissions;
- Version control analysis, using platforms like Git to track iterative work;
- AI-inclusive tasks, such as critiquing or improving AI-generated solutions;
- Project-based learning, which emphasizes design, collaboration, and testing rather than just code correctness.

Despite the possible benefits, these approaches face implementation challenges, particularly in large classes, due to increased demands on teaching time and the need for closer monitoring. As noted by Zawacki-Richter [19], institutions must invest in faculty training, establish clear academic integrity policies, and adopt scalable and process-oriented evaluation strategies that measure both code and understanding.

4 Survey Methodology and Questionnaire Design

To investigate the research questions about the impact of Artificial Intelligence, particularly Generative AI tools, on the evaluation process in computer programming subjects, an online survey was conducted and disseminated amongst Portuguese higher education professors,

who teach programming-related subjects. The target audience for this study was reached through two main channels: public institutional websites (where contact information of the teaching staff was available) and personal academic networks. Participants were informed that the questionnaire was anonymous, ensuring the confidentiality of the responses. A total of 31 responses were collected from professors from 12 different Portuguese higher education institutions, including both universities and polytechnic institutes. The respondents teach in higher professional technical courses (TeSP), bachelor's and/or master's degrees, and represent 3 class formats: in-person, online, and hybrid teaching. This heterogeneity enhances the richness of the dataset, reflecting a broad spectrum of perspectives within the national context. Although it is a relatively small sample size and may not fully capture the diversity of institutional contexts, qualitative studies with limited samples can still yield valuable insights into the subject under investigation [10]. Following other recent computing education research studies [11, 6, 2, 16], the survey combined closed- and open-ended questions to gather both quantitative indicators, useful to identify general trends, and qualitative insights, providing depth and context to participants' experiences and perceptions. The structure was as follows: Section A – Respondent characterization and curricular context: data on academic background, such as institution and teaching level, subject area, and class description; Section B – Perceptions of AI usage by students: teacher's point-of-view on students' awareness of AI tools, particularly code generation; Section C – Changes in Evaluation methods and Adopted Strategies regarding AI tools: investigation on whether and how assessment methods have been adapted in response to the AI usage; Section D – Future Perspectives/Challenges: forward-looking reflections about AI in computer programming assessment.

5 Results Analysis and Discussion

5.1 Respondent characterization and curricular context

The most representative institutions were Polytechnic of Porto and Instituto Superior Técnico in Lisbon, representing more than half of the respondents (53%). The remaining institutions had a more modest representation, varying from 1 to 3 respondents each. Overall, 69% of the professors teach computer-related course units in bachelor's degrees, 22% in master's degrees and only 8% in TeSP courses (Figure 1). Most courses are taught in person (86%), while only two respondents reported teaching in hybrid courses and another two in fully online formats.

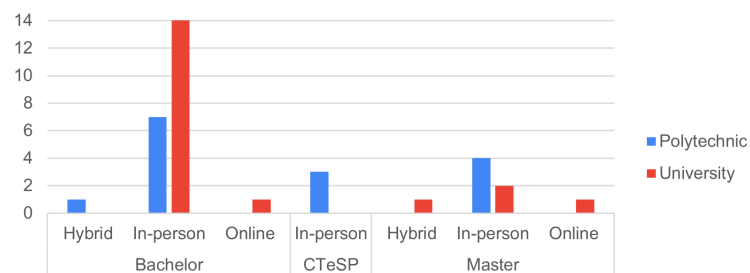


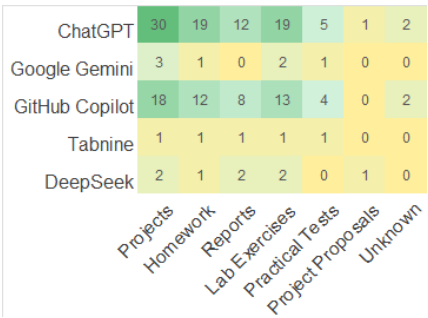
Figure 1 Respondents distribution between institution type, degree level and class format.

The most represented curricular units were at the introductory programming level (28%) and in web development (19%), while more advanced or specialized courses show less representation. In most cases, the number of students per class falls within the ranges of 11

to 20 students (32%) and 21 to 40 students (45%). Around 30% of the respondents reported having classes with more than 60 students, a pattern observed among some of those who indicated teaching theoretical classes, and among all who teach online ones.

5.2 Perceptions of AI usage by students

Professors are aware that their students use generative AI tools in evaluative programming tasks (only 1 respondent answer that he suspects yes, but with no confirmation of such). All respondents indicated ChatGPT as an AI tool used by students, most particularly in projects, laboratorial exercises, homework, and reports (Figure 2). GitHub Copilot also shows notable usage, especially in project-based and technical tasks. In contrast, tools like Google Gemini, DeepSeek, and Tabnine are less mentioned. These findings suggest that students predominantly use AI tools in open-ended and iterative tasks, for real-time assistance or code generation. The respondents were evenly divided regarding the impact of AI on students’ learning of computer programming: 35% believe such tools have a negative impact, 32% consider the impact positive, and the remaining (32%) consider AI neither positive nor negative. In an open-ended answer, respondents’ perceptions were diverse, but they tended to recognize both benefits and risks. Among the positive aspects, the acceleration of the coding process and the reduction of frustration stand out: “it speeds up the coding process. It reduces frustration with basic errors”. Support for autonomous learning is also valued: “when used well, it has a positive impact on accelerating the process, avoiding more routine tasks and facilitating the search for solutions”. However, many professors warn about superficial and uncritical usage of AI tools by students, pointing out the replacement of cognitive effort: “most students [...] use it with little critical sense, to solve exercises, without actually understanding the code”. Others highlight the growing dependence: “the degree of dependence on these tools is increasing and, to some extent, possibly debilitating”. In short, respondents found that the impact of AI depends heavily on how it is used, and it is essential to promote a critical and pedagogical use of these tools.



■ **Figure 2** Co-occurrence matrix between of AI tools usage by students with evaluation tasks.

5.3 Changes in Evaluation methods and Adopted Strategies regarding AI tools

The data collected reveals a significant shift in assessment strategies following the widespread availability of generative AI tools. Most respondents (68%) reported having partially or significantly modified their assessment methods (left chart in Figure 3. To mitigate the risks associated with AI-assisted cheating or superficial learning, instructors have adopted a variety of measures (Figure 4).

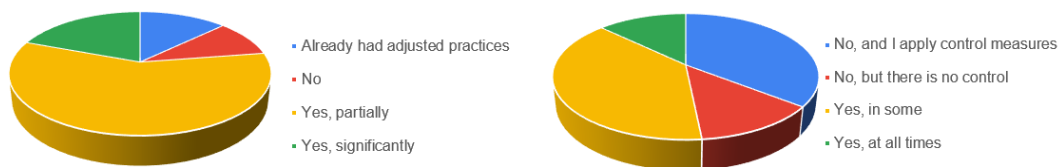


Figure 3 Results regarding changes in evaluation methods due to the emergence of generative AI tools (left) and explicit permission to use AI tools during assessment (right).

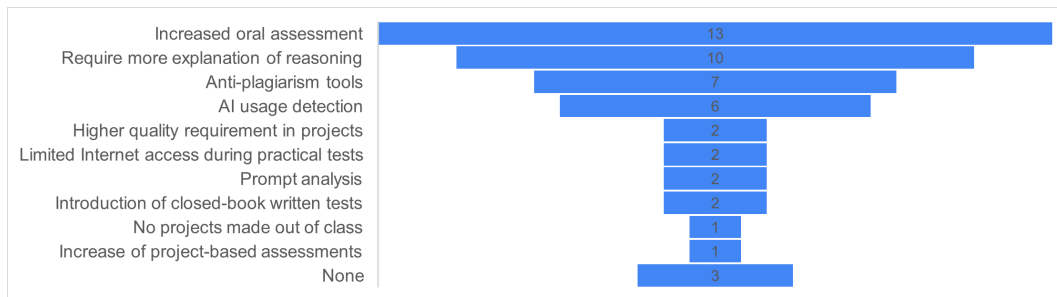


Figure 4 Most frequently adopted strategies/methods in evaluation moments.

Some correlations were found between evaluation changes and adopted strategies. Respondents who significantly changed their evaluation methods tend to use oral assessments or written tests, increase the projects' complexity, apply rigorous control measures, and do not allow AI in assessment moments. Those who partially changed their evaluation methods allow AI in some situations, especially in projects, and place greater value on the explanation of reasoning. Finally, professors who did not change or had already adjusted practices usually apply control measures, but with less emphasis on pedagogical adaptation. Another finding was that instructors who explicitly allow AI tools in some or all assessments (right chart in Figure 3) tend to emphasize project-based learning and critical engagement with AI, including practices such as requiring students to submit the prompts they used or reflect on the role of AI in their work.

5.4 Future Perspectives/Challenges

Major concern identified by the respondents was the lack of fundamental competencies by students (84%). As one professor noted, "too many shortcuts, where most of the code is generated rather than written, will likely prevent students from acquiring basic competencies". Other concerns were unfair assessment between those who use AI and those who do not (32%), difficulty in detecting the usage of AI (29%), and difficulty in ensuring authorship (12%). These concerns reflect a broader uncertainty about how to maintain academic integrity and fairness in an environment where AI tools are widely accessible and increasingly sophisticated. There is a strong consensus that institutional support is needed (61% said it was urgent and 16% think that it is required but not a priority), reflecting the pressure felt to adapt pedagogical practices. The desired support includes pedagogical training on the use of AI, definition of strategies and good practices, tools for detecting code generated by AI, and institutional discussion and sharing among teachers. Respondents who identify loss of skills and difficulty in ensuring authorship as the main challenges are also those who most frequently ask for specific training and clear institutional strategies.

6 Conclusion

This research outlines the impact of generative AI on evaluating students of computer programming courses in higher education. Results reveal that while these tools offer significant pedagogical opportunities (such as enhanced autonomy, real-time feedback, and support for project-based learning), they also pose substantial challenges and concerns, such as the lack of critical reflection, excessive reliance on these tools, and along with difficulties in ensuring authorship. Most professors reported changing their evaluation methods, adopting strategies such as oral assessments, project-based tasks, and emphasis on reasoning. The evidence indicates a definitive shift towards a more reflective, process-oriented, and diversified assessment strategy, with the hope of preserving the integrity of computer programming learning in the age of AI. There is also a clear call for integrated actions on an institutional level, rather than isolated efforts by individual teachers, and the urgency of such support is strongly linked to the deep feeling that AI is changing the way students learn and how they are evaluated. The study contributes by offering a case study from Portuguese higher education, offering practical implications regarding emerging assessment practices (e.g., reasoning and process description, prompt analysis, AI usage detection) and contributes to research about AI-integrated pedagogy, specifically focused on assessing programming skills in higher education. Many studies focus on the use of AI as a teaching support tool, without adequately addressing its impact on the evaluation phase of the teaching-learning process. This study seeks to fill this gap by analyzing real-world practices adopted by professors of computer programming courses. The intersection of AI, teaching programming, and evaluating programmatic skills requires an integrated and critical pedagogical approach. Technology alone cannot solve the challenges of education, but it can be a valuable ally when used intentionally and consciously. Teacher training, investment in infrastructure, and a commitment to ethical and inclusive education are necessary paths to consolidate these innovations in higher education. Future work should explore students' critical examination of generative AI tools, as well as their ethical and cognitive concerns regarding AI-assisted learning and shifting evaluation methodologies. Moreover, the evolving nature of generative AI tools warrants the need for a comprehensive focus on long-term studies on the impact these tools have on developing programming skills.

References

- 1 Maimoona Al Abri, Abdullah Al Mamari, and Zakria Al Marzouqi. Exploring the implications of generative-ai tools in teaching and learning practices. *Journal of Education and e-Learning Research*, 12:31–41, 2025. doi:10.20448/jeelr.v12i1.6355.
- 2 Isaac Alpizar-Chacon and Hieke Keuning. Student's use of generative ai as a support tool in an advanced web development course. *ITiCSE 2025, Computers and Society*, March 2025. URL: <http://arxiv.org/abs/2503.15684>.
- 3 Jonathan Alvarez Ariza, Milena Benitez Restrepo, and Carola Hernandez Hernandez. Generative ai in engineering and computing education: A scoping review of empirical studies and educational practices. *IEEE Access*, 2025. doi:10.1109/ACCESS.2025.3541424.
- 4 Gavina Baralla, Giacomo Ibba, and Roberto Tonelli. Assessing github copilot in solidity development: Capabilities, testing, and bug fixing. *IEEE Access*, 12:164389–164411, 2024. doi:10.1109/ACCESS.2024.3486365.
- 5 René F. Kizilcec, Elaine Huber, Elena C. Papanastasiou, Andrew Cram, Christos A. Makridis, Adele Smolansky, Sandris Zeivots, and Corina Radulescu. Perceived impact of generative ai on assessments: Comparing educator and student perspectives in australia, cyprus, and the united states. *Computers and Education: Artificial Intelligence*, 7, December 2024. doi:10.1016/j.caeai.2024.100269.

- 6 Alejandra J. Magana, P. Felkel, and J. Zara. Computer graphics instructors' intentions for using generative ai for teaching. *The Eurographics Association*, 2025.
- 7 Mary Maher, Yash Tadimalla, and Dhruv Dhamani. An exploratory study on the impact of ai tools on the student experience in programming courses: an intersectional analysis approach. In *IEEE Frontiers in Education Conference*, pages 1–5, October 2023. doi:10.1109/FIE58773.2023.10343037.
- 8 Bayode Ogunleye, Kudirat Ibilola Zakariyyah, Oluwaseun Ajao, Olakunle Olayinka, and Hemlata Sharma. Higher education assessment practice in the era of generative ai tools. *Journal of Applied Learning and Teaching*, 7:46–56, January 2024. doi:10.37074/jalt.2024.7.1.28.
- 9 Lino Oliveira and Mário Pinto. A inteligência artificial na educação - ameaças e oportunidades para o processo ensino-aprendizagem. Technical report, ESMAD, polytechnic of Porto, 2023. URL: <https://www.esmad.ipp.pt>.
- 10 Michael Quinn Patton. *Qualitative research & evaluation methods*. Sage Publications, 2002.
- 11 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 Jaromír Šavelka. 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 (ITiCSE-WGR '23)*, pages 108–159, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3623762.3633499.
- 12 Jenay Robert, Nicole Muscanell, Szymon Machajewski, and Hutson Jamie. 2023 educause horizon action plan: Generative ai. Technical report, Educase Publications, September 2023. doi:10.13140/RG.2.2.21013.09442.
- 13 V. Saravanan, S. Kavitha, S. Ravi, A. Seetha, Ch Rambabu, and Tatiraju Kanth. Generative ai in software engineering: Revolutionizing code generation and debugging. *International Journal of Computational and Experimental Science and Engineering*, 11, May 2025. doi:10.22399/ijcesen.1718.
- 14 Sebastian Simon, Raquel Coelho, Iza Marfisi-Schottman, and Roy Pea. Generative ai tools in an undergraduate computer science program. In *18th International Conference of the Learning Sciences (ICLS)*, pages 2133–2134, June 2024. doi:10.22318/icls2024.271910.
- 15 Zoltán Ságodi, István Siket, and Rudolf Ferenc. Methodology for code synthesis evaluation of llms presented by a case study of chatgpt and copilot. *IEEE Access*, 12:72303–72316, 2024. doi:10.1109/ACCESS.2024.3403858.
- 16 Lisa van der Heyden, Fatma Batur, and Irene-Angelica Chounta. Learning to program: Mapping errors and misconceptions of python novices to support the design of intelligent programming tutors. In *Proceedings of the 17th International Conference on Computer Supported Education (CSEDU 2025) - Volume 1*, pages 224–231. SciTePress, 2025. doi:10.5220/0013203100003932.
- 17 Ying Xie, Shaoen Wu, and Sumit Chakravarty. Ai meets ai: Artificial intelligence and academic integrity a survey on mitigating ai-assisted cheating in computing education. *Association for Computing Machinery*, 2023. doi:10.1145/3585059.
- 18 Ramazan Yilmaz and Fatma Gizem Karaoğlu Yilmaz. The effect of generative artificial intelligence (ai)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence*, 4:100147, June 2023. doi:10.1016/j.caeai.2023.100147.
- 19 Olaf Zawacki-Richter, Victoria I. Marín, Melissa Bond, and Franziska Gouverneur. Systematic review of research on artificial intelligence applications in higher education – where are the educators?, December 2019. doi:10.1186/s41239-019-0171-0.