Integrating Generative AI in CS Education: Trends, Challenges, and Pedagogical Innovations - An ACM-Based Literature Review*

Mauricio Ricardo Viana[1] and Sirazum Munira Tisha[2]
[1,2]Department of Mathematics and Computer Science
Rollins College
Winter Park, FL
mviana@rollins.edu, stisha@rollins.edu

Abstract

The rapid advancement of large language models (LLMs), such as GPT-4 and Claude, is transforming the landscape of computer science (CS) education. This paper presents an initial systematic review (SLR) of the 30 most recent peer-reviewed ACM publications from 2023 to 2025 to examine the integration of generative AI in CS instruction. Guided by four research questions, the review investigates how LLMs are used in teaching, their reliability and accuracy, ethical concerns raised, and frameworks proposed for responsible use. The findings reveal diverse applications including AI-driven tutoring, grading automation, prompt engineering, and curriculum redesign predominantly in introductory courses. While LLMs show promise in augmenting instruction and supporting learners, challenges remain in overreliance, assessment validity, and ethical governance. The review concludes with recommendations for inclusive, transparent, and pedagogically sound AI integration strategies, and highlights research gaps in long-term impact, advanced course adoption, and educator support.

1 Introduction

The emergence of large language models (LLMs) such as GPT-3.5, GPT-4, and Claude has introduced transformative possibilities in computer science (CS) education [20]. These tools, built on advancements in generative artificial intelligence (GenAI), have demonstrated capabilities in code generation, explanation, debugging, and interactive dialogue. As a result, educators and researchers have begun rethinking traditional pedagogical approaches to incorporate LLMs as instructional aids, tutors, graders, and even ethical discussion facilitators [20].

While much of the initial discourse around LLMs focused on concerns regarding plagiarism and academic integrity [15], the landscape has since broadened. There is now growing recognition that, when used appropriately, LLMs can augment learning by supporting critical thinking, customizing feedback, and reducing teaching workloads [30]. However, the rapid pace of technological evolution presents significant challenges for designing, evaluating, and governing the use of LLMs in educational contexts. Institutions are grappling with how to integrate these tools into their curricula while maintaining educational rigor, inclusivity, and ethical standards [19].

This study conducts a systematic literature review (SLR) to investigate the current state of LLM adoption in CS education. We examine how these tools are being used in instructional design, grading automation, and personalized tutoring. We also explore their perceived benefits and limitations, ethical implications, and proposed frameworks for responsible implementation. Our review is guided by four core research questions:

- RQ1: How are LLMs and AI tools used in CS education?
- RQ2: How accurate and reliable are these tools in instructional contexts?
- RQ3: What ethical concerns are raised in the literature?
- RQ4: What frameworks or guidelines are proposed for responsible use?

Through a synthesis of most recent 30 peer-reviewed articles published by Association of Computing Machinery (ACM) [1] between 2023 and 2025, this review provides a detailed account of trends, gaps, and innovations in the application of LLMs in computer science education in the latest practices.

2 Methodology

This study employs a Systematic Literature Review methodology following the structured three-stage approach outlined by Brereton et al. [2]: planning,

[1] https://www.acm.org/

conducting, and reporting. In the planning phase, we formulated four focused research questions (RQs) and established a comprehensive protocol to guide the review process. The conducting phase involved systematic literature searching and the application of explicit inclusion and exclusion criteria to ensure study quality and relevance. Finally, the reporting phase synthesized findings to address our research questions, as presented in this paper.

We conducted our literature search exclusively through the ACM Digital Library. Our search encompassed publications from January 2023 to May 2025, a timeframe chosen to capture developments following ChatGPT's public release, which fundamentally transformed the educational landscape. The search strategy employed multiple keywords including: "AI in CS education," "LLM in CS education," and "Generative AI in CS education."

Studies were selected based on the following inclusion criteria:

1. Full papers with a minimum of six pages (excluding references)
2. Peer-reviewed journal or conference publications
3. Written in English
4. Directly focused on computer science education

We excluded workshop posters, short papers, opinion articles, and studies not centered on CS education contexts.

The initial search yielded 246 research papers, comprising 216 conference proceedings and 30 journal articles. Two reviewers independently screened all records using our predefined criteria, with disagreements resolved through discussion. Records deemed potentially relevant underwent full-text assessment for relevance, methodological clarity, and reported outcomes. This two-stage screening process resulted in a final corpus of 94 studies (13 journal articles and 82 conference proceedings) selected for in-depth analysis.

During the analysis phase, we tracked emerging themes aligned with our research questions. After approximately 30 reviews, thematic saturation was achieved as no new themes emerged. The remaining studies were used to confirm and refine existing thematic categories, with 30 studies directly cited in this paper's findings and discussion sections.

3 Results

Following systematic literature review guidelines [2], this study synthesizes trends across the selected publications rather than conducting empirical experiments. For each publication, we extracted key information including study design, educational context, reported accuracy, ethical considerations, and any proposed frameworks or guidelines. Table 1 presents a summary of representative papers identified in our review. These works represent diverse research methodologies, target populations (students, instructors, TAs), educational levels (CS1, CS2, graduate), and implementation goals (instruction, ethics, grading, tutoring). However, we organised our revived paper as per our four research questions.

Table 1: Comprehensive Summary of Papers on LLM Use in CS Education

References	**Focus Area**	**Key Contributions**	**Related RQs**
[1, 5, 7, 8, 9, 10, 11, 13, 14, 17, 18, 20, 24, 25, 26, 30, 31]	AI integration	Curriculum design, automated grading, AI tutoring, HCI education, validates response accuracy, student feedback on interaction quality, prompt engineering and code accuracy	RQ1, RQ2
[5, 8, 10, 14, 16, 17, 20, 22, 23, 24, 28, 31]	Accuracy and student perception	Positive experiences with improving accuracy, cost effectiveness	RQ2, RQ3
[3, 4, 6, 12, 16, 17, 18, 20, 21, 22, 23, 27, 29]	Ethics Awareness	Discuss impacts, threats, overreliance and ethical concerns and awarness	RQ3
[6, 7, 9, 12, 13, 18, 20, 21, 29]	Framework	Redesigned curriculum focusing on inclusivity, prompt engineering etc, ethics module for CS/non-CS majors,policy reviews, integration strategies	RQ4

RQ1: How are LLMs and AI tools used in CS education?
Numerous studies explore the integration of LLMs in computer science education. For instance, researchers in [7] present curriculum designs that incorporate prompt engineering and emphasize ethical AI usage. AI-powered tutors specifically developed for introductory CS courses (CS1) are discussed in [14] and [5], while researchers in [30] examine the role of LLMs in grading support. Additionally, authors in [31] evaluate an interactive prompting system (iGPT) aimed at enhancing programming performance.

According to Yeh et al. (2025) [31] and Abolnejadian et al. (2024) [1], Over 80% of the reviewed studies focus on undergraduate CS1 and CS2 courses. Common applications include feedback generation, code explanation, assignment scaffolding, and project support. The tools examined most frequently are GPT-3.5, GPT-4, and GitHub Copilot [17, 26]. A few papers, such as Vadaparty et al. (2024) [26], investigate curriculum-wide redesigns that embed LLMs as central instructional tools. Beyond programming, LLMs have also been explored in contexts such as ethics education [4].

RQ2: How accurate and reliable are these tools in instructional contexts?
The accuracy and reliability of LLMs in educational settings are addressed across several studies. For example, researchers in [14] evaluates the correctness of LLM-generated tutor responses by validating them against course materials, while researchers in [30] compare AI-generated feedback with instructors' expectations to assess alignment with pedagogical goals. The study by [31] demonstrates that interactive prompting significantly enhances student outcomes by fostering engagement and guidance. Similarly, the research by [23] analyzes student perceptions of LLM support, revealing both the perceived benefits and noted limitations of these tools.

Majority of the reviewed articles report that LLMs perform consistently well on straightforward tasks, particularly in routine code generation and syntax correction scenarios [25]. However, their performance declines in tasks requiring higher-order cognitive skills such as abstraction, debugging, or creative problem-solving [10]. To address these challenges, some studies have implemented retrieval-augmented generation (RAG) systems that ground model responses in course-specific materials, showing promising improvements in contextual accuracy [28]. Additionally, interactive mechanisms continue to demonstrate positive impacts on learning outcomes [31].

Despite these advances, several limitations persist. Many models remain highly sensitive to prompt phrasing, often exhibiting brittle behavior when prompts are slightly altered. Moreover, hallucinated responses where LLMs produce plausible but factually incorrect information remain a significant concern, as highlighted in [18]. These issues underline the need for further refinement to ensure dependable integration of LLMs in instructional contexts.

RQ3: What ethical concerns are raised in the literature?
A range of ethical concerns including overreliance on AI, academic integrity violations, and the spread of misinformation are examined across several studies. For instance, researchers in [4] explores ethical implications from both educator and student perspectives, while other works [21, 20, 13] highlight the need for integrating ethics into the curriculum and address risks associated with AI-generated content.

Some of our reviewed studies explicitly identify ethical hazards in their findings. Common issues include plagiarism, excessive dependence on LLMs, and biased outputs rooted in the models' training data [16]. Faculty members often express concerns regarding detection of misconduct, enforcement of academic policies, and equitable access to AI tools. On the other hand, students may exhibit unwarranted trust in AI-generated responses, potentially overlooking errors or biases [10]. These concerns are further intensified by inconsistent institutional policies and disparities in tool accessibility, underscoring the need for thoughtful, context-sensitive integration of LLMs in educational environments.

RQ4: What frameworks or guidelines are proposed for responsible use?
Several studies propose structured approaches to support the responsible integration of LLMs and AI tools in computer science education. For instance, [7] outline curriculum-based frameworks that emphasize principles such as prompt engineering, transparency, and critical reflection. [21] introduces a SWOT-based (Strengths, Weaknesses, Opportunities, Threats) framework to help educators and institutions assess the implications of adopting AI in academic settings. Furthermore, [4] applies global ethical standards—such as UNESCO's framework—to promote interdisciplinary and responsible use of LLMs in educational contexts.

Despite these efforts, few studies provide fully developed or widely adopted frameworks. According to Raihan et al. [20], most existing guidelines are fragmented, locally developed, and still evolving. Some researchers, such as Abolnejadian et al. [1], recommend structured, prompt-based instruction to scaffold responsible use. Others, including Deb et al. [4], advocate for embedding ethical reasoning directly into technical coursework. Rather than endorsing blanket bans on AI tools, these studies emphasize fostering critical thinking, promoting transparency, and encouraging metacognitive engagement. A recent contribution by further underscores the importance of adaptive frameworks that evolve alongside technological and pedagogical advancements.

In summary, our analysis reveals that majority of the studies focus on undergraduate programming courses, particularly CS1. Most of the reviewed literature finds that LLMs are effective for basic code generation tasks but struggle with complex, multi-step, or abstract prompts. Ethical concerns—such as bias, over-reliance, and plagiarism—are explicitly addressed in the studies. While many papers emphasize general principles such as promoting AI literacy and encouraging critical thinking, researchers also propose concrete instructional frameworks or implementation guidelines.

4 Discussion

4.1 Key Findings and Implications

Our systematic literature review reveals a rapidly evolving landscape of AI integration in computer science education, with significant opportunities alongside notable challenges. The concentration of research on undergraduate programming courses, particularly CS1 and CS2 [31, 1], reflects both the accessibility of these contexts for initial experimentation and the fundamental importance of introductory programming education in shaping students' computational thinking skills.

4.2 The Promise and Limitations of Current AI Integration

The finding that 65% of studies report consistent LLM performance on straightforward tasks, while noting degraded performance on higher-order cognitive challenges [25, 10], highlights a critical tension in AI-assisted education. This pattern suggests that current LLM implementations excel as sophisticated code completion and syntax assistance tools but fall short of supporting the deep conceptual understanding and creative problem-solving skills that define expert programmers. The reliance on tools like GPT-3.5, GPT-4, and GitHub Copilot across the majority of studies [17, 26] indicates a convergence around commercially available platforms, potentially limiting the diversity of pedagogical approaches and creating dependencies on proprietary systems.

The success of retrieval-augmented generation (RAG) systems in improving contextual accuracy [28] points toward a promising direction for future development. By grounding AI responses in course-specific materials, these approaches address one of the fundamental challenges of generic LLMs: their tendency to provide technically correct but pedagogically inappropriate responses. However, the persistent issues with prompt sensitivity and hallucinated responses [18] underscore the need for more robust safeguards and instructor oversight.

4.3 Ethical Considerations and Institutional Challenges

The identification of ethical concerns in the reviewed studies reveals a significant gap between awareness and systematic attention to these issues. The prevalence of concerns about plagiarism, over-dependence, and biased outputs [16, 10] suggests that the integration of AI tools is outpacing the development of appropriate ethical frameworks and detection mechanisms. The disparity between faculty concerns about academic integrity enforcement and student tendencies toward uncritical trust in AI-generated content highlights a fundamental misalignment that requires targeted intervention.

The inconsistent institutional policies and disparities in tool accessibility mentioned across studies point to broader equity concerns. As AI tools become increasingly central to programming practice, unequal access could exacerbate existing disparities in computer science education. This suggests that successful AI integration requires not only technical solutions but also institutional commitment to equitable access and comprehensive policy development.

4.4 The Framework Gap and Implementation Challenges

Perhaps most concerning is the finding that studies propose concrete instructional frameworks, despite widespread recognition of the need for structured approaches to AI integration [1, 4]. This gap between identifying challenges and providing actionable solutions suggests that the field is still in an exploratory phase, struggling to translate experimental findings into scalable pedagogical practices.

The fragmented nature of existing guidelines, as noted by Raihan et al. [20], reflects the rapid pace of technological change and the diversity of educational contexts. However, this fragmentation also indicates a need for more collaborative, systematic approaches to framework development. The emphasis on principles like transparency, critical thinking, and metacognitive engagement across multiple studies [7, 21] suggests emerging consensus around core values, even if specific implementation strategies remain diverse.

4.5 Implications for Educational Practice

The dominance of undergraduate-focused research, while providing valuable insights into foundational programming education, leaves significant gaps in our understanding of AI's role in advanced computer science topics and graduate-level instruction. The limited exploration of applications beyond programming, such as the few studies examining ethics education [4], suggests untapped potential for AI integration across the broader CS curriculum.

The interactive prompting successes demonstrated in several studies [31, 14] highlight the importance of designing AI tools as collaborative partners rather than replacement systems. This finding aligns with broader educational research on the value of scaffolded learning and suggests that effective AI integration requires careful attention to the balance between assistance and independence in student learning.

4.6 Future Research Directions

Our analysis reveals several critical areas requiring further investigation. First, longitudinal studies examining the long-term impact of AI tool usage on programming skill development are notably absent from the current literature. Understanding whether early exposure to AI assistance enhances or diminishes students' fundamental programming capabilities is crucial for informed pedagogical decision-making.

Second, the limited attention to advanced CS topics and graduate-level education represents a significant research gap. As AI tools become more sophisticated, their potential applications in areas such as algorithm design, systems programming, and theoretical computer science warrant systematic investigation.

Third, the development of robust, empirically-validated frameworks for responsible AI integration remains an urgent priority. The current emphasis on general principles, while valuable, needs to be complemented by specific, actionable guidelines that can be adapted across diverse institutional contexts.

4.7 Limitations and Considerations

The rapid evolution of AI technology presents inherent challenges for literature reviews in this domain. Many of the tools and capabilities examined in the reviewed studies may already be superseded by more advanced systems, highlighting the need for ongoing research that can keep pace with technological development.

Additionally, the concentration of research in certain geographic regions and institutional types may limit the generalizability of findings. The effectiveness of AI integration strategies likely varies significantly across different cultural, linguistic, and resource contexts, suggesting the need for more diverse research perspectives.

5 Conclusion

This initial literature review of 30 ACM publications (2023-2025) reveals how large language models are reshaping computer science education. Our analysis demonstrates concentrated adoption in undergraduate programming courses [31], where LLMs show consistent performance on basic tasks but struggle with complex problem-solving [10]. While ethical concerns about plagiarism and over-dependence appear some studies [16], some also propose concrete implementation frameworks [1], indicating a critical gap between recognition of challenges and actionable solutions.

Key findings reveal that current AI tools excel as code completion aids but require significant development for deeper educational applications. The success of interactive prompting approaches [31] suggests designing AI as collaborative learning partners offers the most promising direction, though persistent issues with accuracy and bias demand continued oversight.

Future research must address three critical priorities: longitudinal studies on learning outcomes, exploration of AI in advanced CS domains, and development of robust implementation frameworks. The field requires sustained collaboration between educators, researchers, and technologists to realize AI's transformative potential while preserving the human elements essential to effective computer science education.

Acknowledgement

This project is supported by John Hauck Foundation SFCR Fund, Student-Faculty Collaborative Research Fund, Colling-Clint Foundation, Bertoni-Clint Foundation Scholar.

References

[1] Mohammad Abolnejadian, Sharareh Alipour, and Kamyar Taeb. "Leveraging ChatGPT for Adaptive Learning through Personalized Prompt-based Instruction: A CS1 Education Case Study". In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. CHI EA '24. Honolulu, HI, USA: Association for Computing Machinery, 2024. ISBN: 9798400703317. DOI: 10.1145/3613905.3637148. URL: https://doi.org/10.1145/3613905.3637148.

[2] Pearl Brereton et al. "Lessons from applying the systematic literature review process within the software engineering domain". In: *Journal of systems and software* 80.4 (2007), pp. 571–583.

[3] Doga Cambaz and Xiaoling Zhang. "Use of AI-driven Code Generation Models in Teaching and Learning Programming: a Systematic Literature Review". In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024. Portland, OR, USA: Association for Computing Machinery, 2024, pp. 172–178. ISBN: 9798400704239. DOI: 10.1145/3626252.3630958. URL: https://doi.org/10.1145/3626252.3630958.

[4] Debzani Deb et al. "Enhancing University Curricula with Integrated AI Ethics Education: A Comprehensive Approach". In: *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSETS 2025. Pittsburgh, PA, USA: Association for Computing Machinery, 2025, pp. 248–254. ISBN: 9798400705311. DOI: 10.1145/3641554.3701953. URL: https://doi.org/10.1145/3641554.3701953.

[5] Paul Denny et al. "Desirable Characteristics for AI Teaching Assistants in Programming Education". In: *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. ITiCSE 2024. Milan, Italy: Association for Computing Machinery, 2024, pp. 408–414. ISBN: 9798400706004. DOI: 10.1145/3649217.3653574. URL: https://doi.org/10.1145/3649217.3653574.

[6] Tony Haoran Feng et al. "From Automation to Cognition: Redefining the Roles of Educators and Generative AI in Computing Education". In: *Proceedings of the 27th Australasian Computing Education Conference*. ACE '25. Association for Computing Machinery, 2025, pp. 164–171. ISBN: 9798400714252. DOI: 10.1145/3716640.3716658. URL: https://doi.org/10.1145/3716640.3716658.

[7] Amanda S. Fernandez and Kimberly A. Cornell. "CS1 with a Side of AI: Teaching Software Verification for Secure Code in the Era of Generative AI". In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024. Portland, OR, USA: Association for Computing Machinery, 2024, pp. 345–351. ISBN: 9798400704239. DOI: 10.1145/3626252.3630817. URL: https://doi.org/10.1145/3626252.3630817.

[8] Wen-Jung Hsin. "The Effect of ChatGPT: Student Perspective and Performance Achievement". In: *J. Comput. Sci. Coll.* 39.6 (Apr. 2024), pp. 20–29. ISSN: 1937-4771.

[9] Lorraine Jacques. "Teaching CS-101 at the Dawn of ChatGPT". In: *ACM Inroads* 14.2 (May 2023), pp. 40–46. ISSN: 2153-2184. DOI: 10.1145/3595634. URL: https://doi.org/10.1145/3595634.

[10] Chris Kerslake et al. "Exploring Student Reactions to LLM-Generated Feedback on Explain in Plain English Problems". In: *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSETS 2025. Pittsburgh, PA, USA: Association for Computing Machinery, 2025, pp. 575–581. ISBN: 9798400705311. DOI: 10.1145/3641554.3701934. URL: https://doi.org/10.1145/3641554.3701934.

[11] Ahmed Kharrufa and Ian Johnson. "The Potential and Implications of Generative AI on HCI Education". In: *Proceedings of the 6th Annual Symposium on HCI Education*. EduCHI '24. New York, NY, USA: Association for Computing Machinery, 2024. ISBN: 9798400716591. DOI: 10.1145/3658619.3658627. URL: https://doi.org/10.1145/3658619.3658627.

[12] Vassilka D. Kirova et al. "Software Engineering Education Must Adapt and Evolve for an LLM Environment". In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024. Portland, OR, USA: Association for Computing Machinery, 2024, pp. 666–672. ISBN: 9798400704239. DOI: 10.1145/3626252.3630927. URL: https://doi.org/10.1145/3626252.3630927.

[13] Ed Lindoo and Mohamed Lotfy. "Generative AI and its Impact on the CS Classroom and Programmers". In: *J. Comput. Sci. Coll.* 40.2 (Oct. 2024), pp. 35–50. ISSN: 1937-4771.

[14] Rongxin Liu et al. "Teaching CS50 with AI: Leveraging Generative Artificial Intelligence in Computer Science Education". In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024. Portland, OR, USA: Association for Computing Machinery, 2024, pp. 750–756. ISBN: 9798400704239. DOI: 10.1145/3626252.3630938. URL: https://doi.org/10.1145/3626252.3630938.

[15] Robin Maisch, Nathan Hagel, and Alexander Bartel. "Towards Robust Plagiarism Detection in Programming Education: Introducing Tolerant Token Matching Techniques to Counter Novel Obfuscation Methods". In: *Proceedings of the 6th European Conference on Software Engineering Education*. ECSEE '25. Association for Computing Machinery, 2025, pp. 11–19. ISBN: 9798400712821. DOI: 10.1145/3723010.3723019. URL: https://doi.org/10.1145/3723010.3723019.

[16] Eric D. Manley et al. "Examining Student Use of AI in CS1 and CS2". In: *J. Comput. Sci. Coll.* 39.6 (Apr. 2024), pp. 41–51. ISSN: 1937-4771.

[17] Bradley McDanel and Ed Novak. "Designing LLM-Resistant Programming Assignments: Insights and Strategies for CS Educators". In: *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSETS 2025. Pittsburgh, PA, USA: Association for Computing Machinery, 2025, pp. 756–762. ISBN: 9798400705311. DOI: 10.1145/3641554.3701872. URL: https://doi.org/10.1145/3641554.3701872.

[18] James Prather 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*. ITiCSE-WGR '23. Turku, Finland: Association for Computing Machinery, 2023, pp. 108–159. ISBN: 9798400704055. DOI: 10.1145/3623762.3633499. URL: https://doi.org/10.1145/3623762.3633499.

[19] Keith Quille, Brett A. Becker, and Lidia Vidal-Meliá. "The European Commission and AI: Guidelines, Acts and Plans Impacting the Teaching of AI and Teaching with AI". In: *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2*. SIGCSE 2023. Toronto ON, Canada: Association for Computing Machinery, 2023, p. 1410. ISBN: 9781450394338. DOI: 10.1145/3545947.3576353. URL: https://doi.org/10.1145/3545947.3576353.

[20] Nishat Raihan et al. "Large Language Models in Computer Science Education: A Systematic Literature Review". In: *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSETS 2025. Pittsburgh, PA, USA: Association for Computing Machinery, 2025,

pp. 938–944. ISBN: 9798400705311. DOI: 10.1145/3641554.3701863. URL: https://doi.org/10.1145/3641554.3701863.

[21] Jordan Roberts and Abdallah Mohamed. "Generative AI in CS Education: Literature Review through a SWOT Lens". In: *Proceedings of the 26th Western Canadian Conference on Computing Education*. WCCCE '24. Kelowna, BC, Canada: Association for Computing Machinery, 2024. ISBN: 9798400709975. DOI: 10.1145/3660650.3660657. URL: https://doi.org/10.1145/3660650.3660657.

[22] James S. Sharpe, Ryan E. Dougherty, and Sarah J. Smith. "Can ChatGPT Pass a CS1 Python Course?" In: *J. Comput. Sci. Coll.* 39.8 (Apr. 2024), pp. 128–142. ISSN: 1937-4771.

[23] C. Estelle Smith et al. "Early Adoption of Generative Artificial Intelligence in Computing Education: Emergent Student Use Cases and Perspectives in 2023". In: *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. ITiCSE 2024. Milan, Italy: Association for Computing Machinery, 2024, pp. 3–9. ISBN: 9798400706004. DOI: 10.1145/3649217.3653575. URL: https://doi.org/10.1145/3649217.3653575.

[24] Marwa Soudi et al. "Generative AI-Based Tutoring System for Upper Egypt Community Schools". In: *Proceedings of the 2023 Conference on Human Centered Artificial Intelligence: Education and Practice*. HCAIep '23. Dublin, Ireland: Association for Computing Machinery, 2023, pp. 16–21. ISBN: 9798400716461. DOI: 10.1145/3633083.3633085. URL: https://doi.org/10.1145/3633083.3633085.

[25] Nghia D. Tran et al. "Exploring ChatGPT's Ability to Solve Programming Problems with Complex Context". In: *J. Comput. Sci. Coll.* 39.3 (Oct. 2023), pp. 195–209. ISSN: 1937-4771.

[26] Annapurna Vadaparty et al. "CS1-LLM: Integrating LLMs into CS1 Instruction". In: *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. ITiCSE 2024. Milan, Italy: Association for Computing Machinery, 2024, pp. 297–303. ISBN: 9798400706004. DOI: 10.1145/3649217.3653584. URL: https://doi.org/10.1145/3649217.3653584.

[27] Samangi Wadinambiarachchi et al. "The Effects of Generative AI on Design Fixation and Divergent Thinking". In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI '24. Honolulu, HI, USA: Association for Computing Machinery, 2024. ISBN: 9798400703300. DOI: 10.1145/3613904.3642919. URL: https://doi.org/10.1145/3613904.3642919.

[28] Kevin Shukang Wang and Ramon Lawrence. "Quantitative Evaluation of Using Large Language Models and Retrieval-Augmented Generation in Computer Science Education". In: *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSETS 2025. Pittsburgh, PA, USA: Association for Computing Machinery, 2025, pp. 1183–1189. ISBN: 9798400705311. DOI: `10.1145/3641554.3701917`. URL: `https://doi.org/10.1145/3641554.3701917`.

[29] Chuhao Wu et al. "Reacting to Generative AI: Insights from Student and Faculty Discussions on Reddit". In: *Proceedings of the 16th ACM Web Science Conference*. WEBSCI '24. Stuttgart, Germany: Association for Computing Machinery, 2024, pp. 103–113. ISBN: 9798400703348. DOI: `10.1145/3614419.3644014`. URL: `https://doi.org/10.1145/3614419.3644014`.

[30] Irfan Yaqub et al. "A Novel Framework using Large Language Models to Automate Coursework Feedback for Computer Science modules". In: *Proceedings of the 2024 16th International Conference on Education Technology and Computers*. ICETC '24. Association for Computing Machinery, 2025, pp. 130–137. ISBN: 9798400717819. DOI: `10.1145/3702163.3702182`. URL: `https://doi.org/10.1145/3702163.3702182`.

[31] Thomas Y. Yeh et al. "Bridging Novice Programmers and LLMs with Interactivity". In: *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSETS 2025. Pittsburgh, PA, USA: Association for Computing Machinery, 2025, pp. 1295–1301. ISBN: 9798400705311. DOI: `10.1145/3641554.3701867`. URL: `https://doi.org/10.1145/3641554.3701867`.