Multidisciplinary Journal of Research in Engineering and Technology

ISSN: 2348-6953

Volume 13 Issue 02 (2026) | Pages: 274-279

The Role of AI in Shaping the Coding Practices of Beginner Developers: A Systematic Literature Review

Kalpana Dhende[1], Sahil Kumbhar[2], Om Lahamage[3], Kusaram Pawara[4], Krushna Marathe[5], Nihal Shaikh[6]

[1,2,3,4,5,6]Department of MCA, MES IMCC, Pune

[1]ksd.imcc@mespune.in, [2]sahilkumbharsk47@gmail.com, [3]omlahamge2003@gmail.com, [4]pawarakusaram@gmail.com, [5]marathek321@gmail.com, [6]nihalshaikh8499@gmail.com

Peer Review Information

Type: *Article*
Received: *20 March 2026*
Revised: *03 April 2026*
Accepted: *21 May 2026*
Published: *03 June 2026*

Abstract

The rapid advancement of Artificial Intelligence (AI), particularly generative tools such as ChatGPT and GitHub Copilot, is fundamentally reshaping the way novice developers approach programming education. This systematic literature review synthesizes findings from thirteen peer-reviewed studies published between 2021 and 2025, examining the psychological, practical, and pedagogical effects of AI on beginner programmers. Thematic analysis uncovers five major trends: (1) AI as an on-demand learning assistant; (2) structural alteration of coding habits; (3) motivational and engagement dynamics; (4) pedagogical and ethical challenges; and (5) tool-specific strengths and limitations. Newly identified themes include AI's influence on self-regulated learning and its differential impact across demographic groups. Critical research gaps in assessment integrity, long-term cognitive development, and equitable access are identified, with recommendations offered for AI-literate curriculum design.

Keywords: Artificial Intelligence; Beginner Developers; Programming Education; Generative AI; ChatGPT; GitHub Copilot; Cognitive Load; Academic Integrity; Self-Regulated Learning.

How to Cite This Article

Dhende, K., Kumbhar, S., Lahamage, O., Pawara, K., Marathe, K., & Shaikh, N. (2026). The role of AI in shaping the coding practices of beginner developers: A systematic literature review. *Multidisciplinary Journal of Research in Engineering and Technology*, 13(2), 274-279.

Introduction

Research Problem

The incorporation of AI-powered tools—most prominently ChatGPT [1] and GitHub Copilot [4]—into programming education has triggered wide-ranging global debate. While these tools provide immediate, context-aware support and generate executable code snippets, serious questions have emerged regarding their influence on learning effectiveness, conceptual comprehension, and autonomous problem-solving. Beginning developers, who are at the most formative stage of their technical identity, may be disproportionately affected by over-reliance on AI assistance [5], [8].

The challenge is not merely technical. Instructors face a transformed classroom environment in which traditional assessment methods are rendered inadequate [1], and students must navigate the fine line between legitimate tool use and academic dishonesty [7]. The cognitive processes that underpin deep programming competence—debugging, algorithm design, logical reasoning—risk being bypassed when AI delivers instant solutions [5], [6].

Need for Review

Despite a growing volume of literature on AI in general education, empirical research specifically targeting beginner developers remains fragmented. Prior reviews have largely examined AI's role in higher education broadly or focused on a single tool in isolation [2], [3]. This review addresses that gap by methodically examining recent peer-reviewed literature to identify convergent themes, unresolved tensions, and emerging research priorities. The review period (2021–2025) captures

Theme 4: Pedagogical and Ethical Challenges

Banerjee et al. [1] note that instructors report significant difficulty evaluating students' original thought when AI assistance is pervasive. Standard plagiarism detection tools are ill-equipped to identify AI-generated code [7]. A 2024 survey by Dewar et al. [4] found only 31% of institutions had formal AI use policies for programming courses, yet 78% of students reported using such tools regularly.

Abubakar et al. [15] propose a tripartite ethical framework distinguishing "transparent use" (disclosed, structured AI assistance), "grey use" (undisclosed but non-deceptive), and "dishonest use" (submission of AI work as entirely one's own)—providing institutions a practical basis for policy development beyond blanket prohibition.

Theme 5: Tool-Specific Strengths and Limitations

Dewar et al. [4] found GitHub Copilot excels at syntactic and boilerplate completion, while performance degrades on logic-heavy problems. ChatGPT demonstrates superior concept explanation and error interpretation but is prone to hallucinations in domain-specific contexts [1]. CodeT5 offers multilingual support but is underutilized due to limited IDE integration [2]. Codex demonstrates robust logic generation but its API-access requirement creates barriers for resourceconstrained students [8].

the inflection point at which generative AI tools became widely accessible and pedagogically consequential.

Objectives

- O1: Identify and synthesize research on AI tools used by beginner developers.
- O2: Analyze how AI influences learning strategies, motivation, and performance.
- O3: Evaluate ethical and pedagogical implications of AI use in programming education.
- O4: Propose a framework for responsible AI integration in programming curricula.
- O5: Explore differential impacts of AI tools across demographic and institutional contexts.

Methodology

Inclusion/Exclusion Criteria

- Inclusion: Peer-reviewed articles (2021–2025); AI tools in programming education; beginner/undergraduate students; English language; empirical or survey-based designs.
- Exclusion: Non-educational or expert-developer studies; grey literature; non-English publications; sample size < 20.

Databases Searched

IEEE Xplore, ACM Digital Library, SpringerLink, Google Scholar, Scopus, and Elsevier ScienceDirect.

Search Query & Keywords

Boolean strings: “AI in programming education”; “ChatGPT for novice coders”; “Generative AI tools for learning programming”; “AI code generation tools and beginners”; “self-regulated learning and AI programming”; “academic integrity generative AI students”.

Review Process

A four-stage PRISMA-aligned process was adopted: (1) Identification—retrieval of 520+ papers via keyword searches; (2) Screening—filtering via abstracts and inclusion criteria; (3) Eligibility Assessment—full-text review of 38 shortlisted articles; (4) Final Inclusion—selection of 13 high-relevance peer-reviewed studies.

Data Extraction and Synthesis

A structured coding framework tagged key themes, study designs, participant populations, AI tools, and outcomes. Thematic analysis followed Braun & Clarke’s six-phase methodology [10]. Inter-rater reliability was verified via double-coding (25% of corpus), yielding a Cohen’s Kappa of 0.81. Triangulation with expert commentary and student focus group data further validated emergent themes.

Findings & Thematic Analysis

Theme 1: AI as a Learning Assistant

AI tools have emerged as highly effective learning companions offering on-demand mentorship through instant explanations and contextual feedback. Rezaei et al. [6] demonstrate that learners frequently leverage AI for debugging, syntax correction, and grasping complex programming concepts. Rodriguez et al. [9] reported that over 70% of students regularly use AI tools for programming tasks.

Discussion

Gaps in Literature

- Longitudinal Impact: Scarcity of multi-semester or post-graduation follow-ups examining long-term competence development [2], [3].
- Cognitive Development: Underexplored effect of AI on debugging intuition, algorithmic thinking, and system design; most studies rely on task-completion metrics [5], [8].
- Instructor Perspective: Only 2 of 13 reviewed studies included instructor data; pedagogical adaptations remain under-documented
- [1], [4].
- Diverse Populations: Minimal representation of learners from lowincome, rural, or linguistically diverse backgrounds [11], [15].
- Assessment Validity: Existing grading rubrics were not designed for AI-augmented learning; redesign research is almost entirely absent [1], [7].

Trends & Future Research Directions

- AI Literacy Programs: Structured curricula teaching when, how, and why to use AI responsibly.
- Adaptive AI Tutors: Systems that gradually reduce support as student competence grows.
- Explainable AI in Education: AI tools that surface reasoning steps to promote metacognitive engagement.
- Equitable Access Research: Infrastructure requirements for resource-constrained environments globally.
- Ethical Assessment Frameworks: Policy templates distinguishing AI collaboration from academic dishonesty. Collaborative AI Platforms: AI-embedded peer learning where AI facilitates rather than replaces student interaction.

Curriculum Integration Framework

- Phase 1 — Foundations: Introduction to AI capabilities, limitations, and responsible use.
- Phase 2 — Guided Practice: Structured sessions with ChatGPT/Copilot requiring students to critique and explain AIgenerated code.
- Phase 3 — Ethics & Integrity: Modules on academic honesty and code ownership in the AI era.
- Phase 4 — Restricted Assessments: Alternating tool-assisted and tool-restricted evaluations to verify foundational competence.

Case Studies

Undergraduate Python Course with ChatGPT

Students reported greater comprehension and reduced frustration, but conceptual assessment performance lagged behind control cohorts. Instructors observed a progressive reduction in debugging effort corroborated by weekly reflective journals, suggesting a forming dependency that erodes critical learning processes without structured intervention [6].

GitHub Copilot in Introductory Java

65% of students credited Copilot with improving task completion speed, but only 42% felt it contributed to conceptual understanding. Teaching assistants documented a significant increase in semantic errors that students failed to recognize—indicating Copilot enhanced surface productivity without cultivating deeper comprehension [4], [8].

Self-Regulated Learning with Mixed AI Tools

70% of students felt more confident when using AI tools, with reduced anxiety and increased perseverance on challenging assignments.

Okonkwo & Awwal [11] found that students from underresourced institutions reported AI tools democratizing access to expert-level guidance previously unavailable in their environments. However, this benefit was contingent on internet access and device quality. Chen et al. [12] found AI-enabled explanations improved conceptual quiz performance by 18% among first-semester students versus textbook-only controls— most pronounced for students with lower prior programming exposure.

Theme 2: Alteration of Coding Habits

The integration of AI has produced a measurable shift from exploratory problem-solving to solution-retrieval behavior. Chowdhury et al. [5] observe that while AI reduces cognitive load, it may promote surface-level engagement. Lin et al. [2] confirm that many students accept AI-generated code without critically verifying correctness—a behavior termed “passive consumption.”

Kumar et al. [8] documented a 34% decline in time spent on manual debugging in Copilot-assisted cohorts, with corresponding increases in undetected semantic errors. Park & Kim [13] introduce “AI-mediated learned helplessness”: students highly dependent on AI during a semester demonstrated significantly lower autonomous problem-solving scores at semester’s end compared to moderate users.

Theme 3: Motivational and Engagement Dynamics

Zheng et al. [3] report increased satisfaction among AI users, though not consistently translating to stronger long-term retention. A critical distinction emerges between intrinsic and extrinsic motivation: students framing AI as a “collaborator” showed higher intrinsic motivation scores at semester end [3], [11], while students using AI primarily to complete faster exhibited declining self-efficacy over time.

Yamamoto et al. [14] found that structured AI interaction—requiring students to explain AI-generated code before submission—resulted in 27% higher conceptual assessment scores than unstructured AI use, establishing pedagogical design as the key moderating factor.

Yamamoto et al. [14] examined 180 undergraduate students using a combination of ChatGPT, Copilot, and CodeT5. Students demonstrating high self-regulated learning (SRL) behaviors—setting goals before AI use, reflecting afterward—scored 31% higher on end-of-semester exams than low-SRL peers with equivalent AI access, establishing SRL as a critical mediating variable.

Table 1. AI Tool Comparison Matrix

Tool	Primary Strengths	Key Weaknesses	Best Use Case
ChatGPT	Concept explanation, syntax correction, multilingual support	Hallucinations, context loss in long sessions	Debugging guidance, concept clarification
GitHub Copilot	Auto-completion, boilerplate/template generation, real-time suggestions	Poor with complex logic, encourages passive use	Boilerplate code, starter templates
CodeT5	Multilingual code support, open-source	Limited IDE integration, fewer documentation resources	Multi-language learning environments
Codex	Robust logic generation, strong on algorithmic tasks	Requires API access, steep learning curve, cost barriers	Advanced problem-solving, algorithm design
Gemini Code Assist	Integrated IDE support, multi-modal reasoning	Newer tool; limited peerreviewed evaluation to date	Full-stack scaffolding, documentation generation

Appendices

Appendix A: Study Characteristics Matrix

Table 2. Study Characteristics

Study	n	Tool Studied	Key Findings
Banerjee et al. [1]	300+	ChatGPT	Enhanced motivation; instructor evaluation challenges
Lin et al. [2]	120	Multiple	High engagement; low conceptual retention
Zheng et al. [3]	150	ChatGPT/Copilot	Short-term motivation gains; declining intrinsic drive
Dewar et al. [4]	95	ChatGPT & Copilot	Efficiency improved; 31% of institutions had AI policies
Chowdhury et al. [5]	80	ChatGPT	Reduced cognitive load; risk of surface-level understanding
Rezaei et al. [6]	100	ChatGPT	Reduced anxiety; improved syntax skills
Lewis et al. [7]	200	Multiple	Perception gaps on ethical AI use boundaries
Kumar et al. [8]	200	Copilot	34% decline in debugging time; increased semantic errors
Rodriguez et al. [9]	500+	Multiple	70%+ reported increased confidence with AI tools
Okonkwo & Awwal [11]	160	ChatGPT	AI democratizes access; equity concerns persist
Chen et al. [12]	240	ChatGPT & CodeT5	18% quiz improvement; hybrid tool strategy recommended
Park & Kim [13]	110	Copilot	AI-mediated learned helplessness identified
Yamamoto et al. [14]	180	Mixed	SRL mediates AI impact; 31% exam score differential

Appendix B: Sample Interview Questions from Literature

- How do you decide when to use AI tools during programming tasks?
- What have you learned from ChatGPT that you could not from documentation alone?
- Do you believe Copilot improves or hinders your long-term understanding?
- How do you verify AI-generated code is correct before submission?
- How do instructors handle AI-based academic integrity concerns?
- Is your institution's AI policy fair and clearly communicated?

Appendix C: Suggested Course Modules

- Module 1: Introduction to Generative AI — Capabilities, Architecture, and Limitations
- Module 2: Debugging With and Without AI — Comparative Skill Building
- Module 3: Assessing and Critiquing AI-Generated Code
- Module 4: Plagiarism, Ownership, and Ethics in the AI Era
- Module 5: Collaborative Coding with AI Assistants
- Module 6: Self-Regulated Learning Strategies for AI-Augmented Environments

Conclusion

AI tools present a dual-edged reality for programming education. When thoughtfully integrated, they reduce anxiety, accelerate skill acquisition, and democratize access to expertlevel guidance. When adopted without pedagogical structure, they risk substituting superficial solution retrieval for deep cognitive engagement.

Evidence from thirteen peer-reviewed studies establishes that AI-assisted programming outcomes are determined not by AI access alone, but by the pedagogical framework embedding that access. Self-regulated learning behaviors, structured reflection requirements, and alternating restricted/assisted assessments emerge as the most effective moderating interventions.

Institutions that embrace AI judiciously—with clear ethical frameworks, adaptive scaffolding, and AI literacy programs—will be best positioned to develop flexible, critically thinking, and ethically grounded programmers for the decades ahead.

References

1. P. Banerjee et al., "Understanding ChatGPT: Impact Analysis and Path Forward for Teaching Computer Science and Engineering," IEEE Access, vol. 13, pp. 12041–12058, 2025.
2. X. Lin et al., "Generative AI in Engineering and Computing Education: A Scoping Review," Computers & Education, vol. 210, pp. 104–119, 2024.
3. Y. Zheng et al., "Generative Artificial Intelligence in Computer Programming: Does It Enhance Learning Motivation?," J. Computer Assisted Learning, vol. 39, no. 4, pp. 892–907, 2023.
4. J. Dewar et al., "Investigating the Impact of Code Generation Tools (ChatGPT & GitHub Copilot) on Undergraduate Programming Education," in Proc. ACM SIGCSE, pp. 441–447, 2023.
5. B. Chowdhury et al., "Programming is Hard—Or at Least It Used to Be," in Proc. ACM SIGCSE, pp. 500–506, 2023.
6. R. Rezaei et al., "Students' Experiences of Using ChatGPT in an Undergraduate Programming Course," ACM Trans. Computing Education, vol. 24, no. 2, pp. 1–22, 2024.
7. J. Lewis et al., "Students' Perceptions and Use of Generative AI Tools for Programming," in Proc. IEEE Frontiers in Education Conf., pp. 1–6, 2023.
8. R. Kumar et al., "Studying the Effect of AI Code Generators on Supporting Novice Programmers," in Proc. ACM CHI, pp. 1–21, 2023.
9. M. Rodriguez et al., "A Large-Scale Survey on the Usability of AI Programming Assistance Tools," IEEE Trans. Education, vol. 66, no. 5, pp. 410–421, 2023.
10. V. Braun and V. Clarke, "Using Thematic Analysis in Psychology," Qualitative Research in Psychology, vol. 3, no. 2, pp. 77–101, 2006.
11. C. Okonkwo and A. Awwal, "AI Tools and Equity in Undergraduate Programming: A Cross-Institutional Study," Computers & Education Open, vol. 5, pp. 100–113, 2024.
12. H. Chen et al., "Comparative Effectiveness of AI-Assisted Learning Tools for First-Year Programming Students," J. Educational Technology & Society, vol. 27, no. 1, pp. 55–71, 2024.
13. S. Park and J. Kim, "AI-Mediated Learned Helplessness in Introductory Programming: A Longitudinal Analysis," Computers in Human Behavior, vol. 148, pp. 107–122, 2024.
14. T. Yamamoto et al., "Self-Regulated Learning as a Mediator of AI Tool Effectiveness in Programming Education," British J. Educational Technology, vol. 55, no. 3, pp. 789–808, 2024.
15. I. Abubakar et al., "An Ethical Framework for AI Use in Programming Courses," Ethics and Information Technology, vol. 26, pp. 1–15, 2024.