



GENERATIVE AI TOOLS AND PROGRAMMING SKILLS OF IT STUDENTS: A REVIEW OF BENEFITS AND THE RISK OF OVER-DEPENDENCE

Sarvesh Jadhav

Keraleeya Samajam's Model College, Dombivli East, Mumbai, Maharashtra, India.

Abstract- Generative Artificial Intelligence (GenAI) tools such as ChatGPT, GitHub Copilot and Gemini can write, explain and correct program code within seconds. These tools are now widely used by Information Technology (IT) students for practicals, assignments and projects. At the same time, practical examinations, vivas and job interviews still expect students to write and debug programs on their own. This review paper examines fifteen research studies published between 2022 and 2025 on the use of GenAI in programming education and on students' perceptions of these tools. The studies were grouped into five themes: the capability of AI code generators, their effect on novice learners and productivity, student perceptions and usage patterns, the risk of over-dependence and cognitive offloading, and the response of educators. The review finds that GenAI tools clearly improve speed and task completion and can support learning when used carefully. However, controlled studies also show that unrestricted use can reduce independent performance once AI support is removed, and that frequent use is linked with lower critical thinking. Based on these findings, the paper proposes a simple four-step "Learn-Generate-Verify-Explain" model for responsible use of AI in programming courses and identifies the need for quantitative studies on IT students in Indian cities.

Keywords: Generative AI, ChatGPT, GitHub Copilot, Programming Education, Coding Skills, Over-Dependence, Cognitive Offloading, Review.

INTRODUCTION

Programming is the central skill of every IT and computer science course. Students learn it by writing small programs, making mistakes, reading error messages and correcting their own logic. This slow process of trial and error is how problem-solving ability is built.

Since the public release of ChatGPT in November 2022, this process has changed. A student can now type a practical question into an AI tool and receive complete, working code along with an explanation. Tools such as GitHub Copilot go further and suggest code automatically inside the code editor. For students, this saves time and helps them understand difficult topics [1].

However, the same convenience can reduce the effort that students put into thinking through a problem. If code is copied without being understood, a student may submit a correct practical but be unable to write a similar program in an examination, a viva or a job interview, where AI help is not available.

Many studies on this subject have appeared in a short time, but they come from different fields, use different methods and sometimes reach different conclusions. The purpose of this paper is to bring these studies together, compare their findings and identify what is known and what is still missing about the effect of GenAI tools on the programming skills of students.

Objectives of the review

1. To review what current GenAI tools can do in programming tasks given to students.
2. To summarise the evidence on how these tools affect learning, performance and productivity.
3. To review how higher-education students perceive and use GenAI tools.

4. To examine the evidence on over-dependence and its effect on independent problem-solving.
5. To propose a model for responsible use of AI in programming courses and identify gaps for future research.

GENERATIVE AI IN PROGRAMMING EDUCATION

GenAI tools support students at almost every stage of programming. The main uses, and the concerns attached to them, are:

2.1 Code Generation: Students can obtain complete programs for practicals from a short text prompt. This saves time but may skip the learning step [2].

2.2 Debugging Support: AI tools explain error messages and suggest fixes, helping students move past problems that once took hours.

2.3 Concept Explanation: Students use AI as a personal tutor to explain topics such as recursion, object-oriented programming and data structures [1].

2.4 Learning New Languages: AI helps students translate logic they already know into an unfamiliar language such as Java, Python or JavaScript.

2.5 Risk of Over-Dependence: Frequent reliance on generated code may weaken independent problem-solving and critical thinking.

2.6 Unverified Code: AI can produce code that looks correct but contains logical errors, so students who do not test it may submit wrong work.

2.7 Need for Responsible Use: Colleges need clear guidance so that students use AI to learn, not only to finish tasks [1].

REVIEW METHODOLOGY

This paper follows a narrative (thematic) literature review approach. Research papers were searched in Google Scholar, the ACM Digital Library, Crossref and arXiv using combinations of the keywords "ChatGPT", "GitHub Copilot", "Codex", "generative AI", "large language models", "programming education", "novice programmers", "student perceptions", "over-reliance" and "critical thinking". The following selection criteria were applied:

- Published between 2022 and 2025, i.e. after AI code generators became publicly available.
- Published in a peer-reviewed journal or conference, or a widely cited preprint.
- Directly addresses either (a) GenAI in programming or computing education, or (b) higher-education students' use and perception of GenAI, or (c) the effect of GenAI use on learning and thinking.
- The full bibliographic record could be verified through its DOI or arXiv identifier.

Fifteen studies were selected. Each study was read for its aim, method, sample and main findings. The findings were then grouped into five themes, which are discussed in Section IV and summarised in Table 1. The review process is shown in Fig. 1.

Fig. 1. Review process followed in this study

1. Define objectives and search keywords →	2. Search Google Scholar, ACM Digital Library, Crossref and arXiv (2022–2025) →	3. Screen titles and abstracts against the selection criteria →	4. Select 15 studies and verify each through DOI / arXiv →	5. Extract aim, method, sample and findings → group into 5 themes →	6. Identify research gaps and propose a responsible-use model
--	---	---	--	---	---

LITERATURE REVIEW

4.1 Capability of AI Code Generators

Finnie-Ansley et al. tested OpenAI Codex on real questions taken from introductory programming examinations and compared its results with those of students who took the same exams under normal conditions. Codex outscored most of the students and passed many test cases for several versions of the well-known "Rainfall Problem". The authors also observed that the same prompt often produced very different solutions [2]. Becker et al., in a position paper, argued that such tools bring immediate opportunities and challenges to introductory programming, and that educators must act quickly to shape how they are used [3]. Prather et al. benchmarked several GenAI models on computing education datasets and showed that their capabilities were improving rapidly [4]. Denny et al. described how large language models capable of generating accurate source code from plain-language problem descriptions are changing the challenges faced by both computing educators and students [5].

Together, these studies establish that typical college programming exercises can now be solved by AI, which means that a completed practical is no longer clear evidence that the student has learned the skill.

4.2 Effect on Learning and Productivity

Kazemitabaar et al. conducted a controlled experiment with 69 novice learners aged 10–17 who completed 45 Python code-writing tasks, half of them with access to Codex. Learners with Codex achieved a 1.15 times higher completion

rate and 1.8 times higher scores, and their performance on manual code-modification tasks did not decrease. Learners with prior programming knowledge benefited most on retention tests one week later [6]. In a professional setting, Peng et al. found that developers using GitHub Copilot completed an HTTP server task in JavaScript 55.8% faster than a control group [7].

A different picture appears when AI support is removed. Bastani et al. ran a field experiment with nearly one thousand high-school mathematics students. Access to GPT-4 during practice improved grades by 48% with a standard ChatGPT-like interface and by 127% with a tutor designed to protect learning. However, when access was taken away, students who had used the standard interface performed 17% worse than students who never had access. The authors observed that students used the tool as a "crutch", and that the learning-focused safeguards largely removed this negative effect [8]. Although this study was in mathematics, it directly concerns programming practicals, where the same pattern of practising with AI and being examined without it applies.

4.3 Over-Dependence and Cognitive Offloading

Gerlich studied 666 participants of different ages and educational backgrounds using surveys and interviews. The study found a significant negative correlation between frequent AI tool use and critical thinking ability, explained largely by cognitive offloading, i.e. handing mental effort over to the tool. Younger participants showed higher dependence on AI tools and lower critical thinking scores than older participants [9]. Kasneci et al. similarly stressed that using large language models in education requires a strong focus on critical thinking, fact-checking and continuous human oversight [1].

For programming students, this suggests that the risk is not AI use as such, but AI use that replaces the student's own reasoning instead of supporting it.

4.4 Student Perceptions and Usage

Ravšelj et al. carried out the largest global study to date, with 23,218 higher-education students from 109 countries. Students mainly used ChatGPT for brainstorming, summarising texts and finding research articles. They found it useful for simplifying complex information, but considered it less reliable for providing information and less useful for developing critical thinking. Students also supported regulation of AI because of concerns about cheating and plagiarism [10].

Das and Madhusudan surveyed 162 undergraduate, postgraduate and doctoral students through Google Forms using convenience sampling. They found a positive perception of ChatGPT's academic applications, and found that perception was not significantly influenced by gender, academic programme or stream [11]. Blahopoulou and Ortiz-Bonnin surveyed 737 undergraduate students at a Spanish university. ChatGPT use was widespread, with 24/7 availability and time saving seen as its main advantages, while students worried that relying on it for assignments could devalue university education. Users generally supported AI integration in teaching, while non-users often favoured banning it [12]. Klimova et al. surveyed 252 undergraduate students in the United States and found that ChatGPT was widely used to understand complex concepts, generate ideas and structure assignments, but that students were concerned about over-reliance, its effect on critical thinking, and plagiarism [13].

Specifically in a computer science course, Aruleba et al. interviewed seven students after a ChatGPT activity involving code completion and analysis. Students valued its immediate responses and support for personalised learning, but were concerned that heavy reliance could harm their critical thinking and problem-solving skills [14]. Prather et al. also surveyed computing students and instructors from 20 countries and reported a wide range of attitudes towards GenAI in computing courses [4].

4.5 Response of Educators

Lau and Guo interviewed 20 introductory programming instructors across nine countries in early 2023. In the short term, many planned measures to discourage AI-assisted cheating. In the long term, opinions were divided: some wanted to ban AI tools and continue teaching programming fundamentals, while others wanted to integrate them to prepare students for future jobs [15]. Prather et al. interviewed 22 computing educators and used the ACM Code of Ethics to give practical advice to policy makers, educators and students [4]. Denny et al. argued that computing education must adapt to the presence of these tools rather than ignore them [5].

Table 1. Summary of reviewed studies

Ref.	Author(s), Year	Method / Sample	Key finding
[1]	Kasneci et al., 2023	Position paper	LLMs offer personalised learning but require critical thinking, fact-checking and oversight.
[2]	Finnie-Ansley et al., 2022	Codex tested on real CS1 exam questions	Codex outscored most students on introductory programming exams.
[3]	Becker et al., 2023	Position paper	AI code generation brings urgent opportunities and challenges to introductory programming.

[4]	Prather et al., 2023	Review of 71 papers, survey (20 countries), 22 interviews, benchmarks	GenAI will profoundly affect computing education; capabilities improving rapidly.
[5]	Denny et al., 2024	Viewpoint article	Educators and students must adapt to LLMs that generate accurate code.
[6]	Kazemitabaar et al., 2023	Controlled experiment, 69 novices	Codex raised completion (1.15×) and scores (1.8×) without reducing manual coding performance.
[7]	Peng et al., 2023	Controlled experiment, developers	Copilot users finished the task 55.8% faster.
[8]	Bastani et al., 2025	Field experiment, ~1,000 students	Unrestricted GPT-4 use lowered grades by 17% once access was removed; safeguards reduced harm.
[9]	Gerlich, 2025	Survey and interviews, 666 participants	Frequent AI use negatively correlated with critical thinking, via cognitive offloading.
[10]	Ravšelj et al., 2025	Global survey, 23,218 students, 109 countries	ChatGPT useful for simplifying information, less useful for critical thinking.
[11]	Das & Madhusudan, 2024	Google Forms survey, 162 students	Positive perception; no significant difference by gender, programme or stream.
[12]	Blahopoulou & Ortiz-Bonnin, 2025	Online survey, 737 students (Spain)	Widespread use; users support AI integration, non-users favour bans.
[13]	Klimova et al., 2025	Survey, 252 students (USA)	Improves efficiency; concerns about over-reliance and critical thinking.
[14]	Aruleba et al., 2023	Interviews, 7 CS students	ChatGPT supports programming learning; heavy reliance may harm problem-solving.
[15]	Lau & Guo, 2023	Interviews, 20 instructors, 9 countries	Instructors divided between banning and integrating AI coding tools.

DISCUSSION

The reviewed studies agree on three points. First, GenAI tools can solve most introductory programming tasks, so completed practicals alone can no longer measure a student's programming ability [2],[4]. Second, these tools increase speed and task completion for both students and professionals [6],[7]. Third, students themselves see clear benefits but are also aware of the risk of over-reliance [10],[13],[14].

The studies differ on whether AI use harms learning. In the programming experiment by Kazemitabaar et al., AI access did not reduce manual coding performance [6]. In contrast, Bastani et al. found a clear negative effect once AI was removed [8], and Gerlich found that frequent AI use was associated with lower critical thinking [9]. A likely explanation is the way the tool is used. In [6], learners were required to modify code by hand after every AI-assisted task, and in [8] the learning-focused tutor largely removed the negative effect. This suggests that AI supports learning when students are made to engage with the code, and harms learning when it simply replaces their effort.

This conclusion is important for IT courses in Indian colleges, where practicals and journals are often assessed on completed output, while external practical examinations and vivas are taken without AI support.

PROPOSED MODEL FOR RESPONSIBLE USE

Based on the findings above, this paper proposes a simple four-step model, "Learn–Generate–Verify–Explain", that students and teachers can follow for programming practicals (Fig. 2).

Fig. 2. Learn–Generate–Verify–Explain model for responsible AI use in programming

1. LEARN	2. GENERATE	3. VERIFY	4. EXPLAIN
Attempt the logic or a first version without AI; use AI only to explain concepts [1],[8]	Use AI to get hints, fix errors or compare with own solution	Run and test AI code with own test cases; never submit unchanged code [2]	Explain every line in own words in the journal or viva; rewrite without AI [6]

↻ repeated for every practical

For teachers, the model implies that practicals should require students to explain and modify their code, that vivas should test understanding rather than output, and that regular short coding exercises without AI should be included in the course [4],[15].

RESEARCH GAP

The review shows the following gaps in existing research:

1. Most large studies examine students in general, not programming students in particular [10],[11],[12],[13].
2. Studies that focus on programming are either controlled experiments with school-age novices [6] or small interview-based studies [14].
3. Very little quantitative evidence exists on how IT students in Indian cities, such as Mumbai, use GenAI specifically for programming.
4. No reviewed study tests whether the frequency of AI use for programming is statistically related to students' confidence and dependence when coding without AI.

PUBLIC SURVEY

To compare the review findings with the local situation, a pilot survey was conducted among IT students in the Mumbai region. An anonymous Google Form with 18 questions (respondent profile, use of AI tools for programming, checking of AI-generated code and six perception statements on an Agree/Neutral/Disagree scale) was shared through college groups, and every respondent gave consent before answering. 15 responses were received (46.7% MSc IT, 20.0% BSc IT, 6.7% MCA, 26.7% other programmes) and analysed using percentages.

RESULTS

14 respondents (93.3%) use a generative AI tool for programming and 9 (60.0%) use it daily (Fig. 3). ChatGPT is the most used tool (53.3%). AI is mainly used for understanding code (60.0%), debugging errors (53.3%) and projects (46.7%), followed by writing new code (40.0%), practicals and journals (33.3%) and learning a new language (26.7%). Two thirds (66.7%) use AI-generated code without changing it at least sometimes, while only 53.3% always test and understand it before submitting.

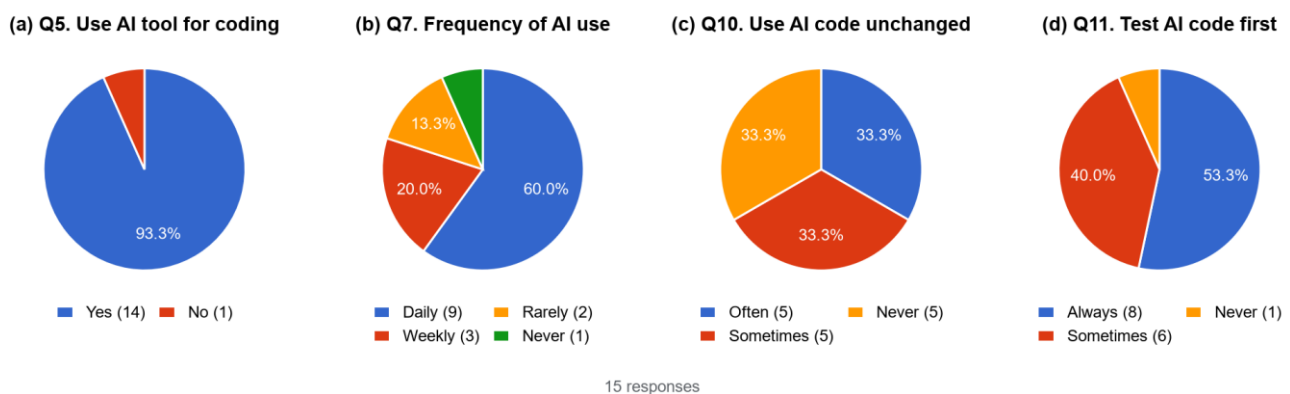


Fig. 3. Use of AI tools and checking of AI-generated code (N = 15)

60.0% agreed that AI tools have improved their understanding of programming concepts and 40.0% that the tools help them finish practicals faster. Only 33.3% agreed that they are confident writing a program without AI, 26.7% felt that they depend too much on AI and 20.0% find it hard to code in exams or vivas without AI. 40.0% agreed that colleges should teach responsible use of AI coding tools, while a large share of respondents chose "Neutral" for these statements.

HYPOTHESIS TESTING

Null Hypothesis (H0): There is no significant relationship between the frequency of AI use (Q7) and confidence in writing a program without AI (Q12). **Alternative Hypothesis (H1):** There is a significant relationship between them. The responses were grouped into a 2×2 contingency table (Table 2). As all expected frequencies were below 5, Fisher's Exact Test was used instead of the Chi-Square test, at $\alpha = 0.05$.

Table 2. Contingency table for Q7 and Q12

Frequency of AI use (Q7)	Agree (confident without AI)	Neutral / Disagree	Total
Daily	3	6	9
Weekly / Rarely / Never	2	4	6
Total	5	10	15

Fisher's Exact Test gave $p = 1.000 > 0.05$, so the null hypothesis is **not rejected**. In this pilot sample, one third of both daily and less frequent users were confident coding without AI. The result should be re-tested with a larger sample.

Overall, the survey findings agree with the reviewed literature and support the Verify and Explain steps of the proposed model.

CONCLUSION

GenAI tools have become a regular part of how IT students learn and practise programming. The reviewed literature shows that these tools solve most introductory programming tasks, help students finish work faster and act as a useful personal tutor. However, controlled experiments and surveys also show that unrestricted use can reduce independent performance once AI is removed and is associated with lower critical thinking. The benefit of these tools therefore depends on how they are used: students who test, modify and explain AI-generated code gain from it, while students who copy it without understanding risk losing the very skill the course is meant to build. Colleges should guide students to use AI as a learning aid, through approaches such as the proposed Learn–Generate–Verify–Explain model, and should design practicals and vivas that require students to explain their code.

FUTURE SCOPE

- Conduct a quantitative survey of IT students in Mumbai to measure the frequency and purpose of AI use for programming.
- Test statistically whether frequent AI use is related to lower confidence and greater dependence when coding without AI.
- Compare practical and examination marks of frequent and occasional AI users.
- Evaluate the proposed Learn–Generate–Verify–Explain model in real practical sessions.
- Include the views of faculty members on AI use in programming courses.

REFERENCES

- [1] E. Kasneci, K. Sessler, S. Küchemann et al., "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, Art. no. 102274, 2023, doi: 10.1016/j.lindif.2023.102274.
- [2] J. Finnie-Ansley, P. Denny, B. A. Becker, A. Luxton-Reilly, and J. Prather, "The robots are coming: Exploring the implications of OpenAI Codex on introductory programming," in *Proc. 24th Australasian Computing Education Conf. (ACE)*, 2022, pp. 10–19, doi: 10.1145/3511861.3511863.
- [3] B. A. Becker, P. Denny, J. Finnie-Ansley, A. Luxton-Reilly, J. Prather, and E. A. Santos, "Programming is hard – or at least it used to be: Educational opportunities and challenges of AI code generation," in *Proc. 54th ACM Technical Symp. on Computer Science Education (SIGCSE)*, 2023, pp. 500–506, doi: 10.1145/3545945.3569759.
- [4] J. Prather, P. Denny, J. Leinonen et al., "The robots are here: Navigating the generative AI revolution in computing education," in *Proc. 2023 Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR)*, 2023, pp. 108–159, doi: 10.1145/3623762.3633499.
- [5] P. Denny, J. Prather, B. A. Becker et al., "Computing education in the era of generative AI," *Communications of the ACM*, vol. 67, no. 2, pp. 56–67, 2024, doi: 10.1145/3624720.
- [6] M. Kazemitabaar, J. Chow, C. K. T. Ma et al., "Studying the effect of AI code generators on supporting novice learners in introductory programming," in *Proc. 2023 CHI Conf. on Human Factors in Computing Systems*, 2023, pp. 1–23, doi: 10.1145/3544548.3580919.
- [7] S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer, "The impact of AI on developer productivity: Evidence from GitHub Copilot," arXiv:2302.06590, 2023. Available: <https://arxiv.org/abs/2302.06590>
- [8] H. Bastani, O. Bastani, A. Sungu et al., "Generative AI without guardrails can harm learning: Evidence from high school mathematics," *Proceedings of the National Academy of Sciences*, vol. 122, no. 26, 2025, doi: 10.1073/pnas.2422633122.
- [9] M. Gerlich, "AI tools in society: Impacts on cognitive offloading and the future of critical thinking," *Societies*, vol. 15, no. 1, Art. no. 6, 2025, doi: 10.3390/soc15010006.
- [10] D. Ravšelj, D. Keržič, N. Tomaževič et al., "Higher education students' perceptions of ChatGPT: A global study of early reactions," *PLOS ONE*, vol. 20, no. 2, Art. no. e0315011, 2025, doi: 10.1371/journal.pone.0315011.
- [11] S. R. Das and J. V. Madhusudan, "Perceptions of higher education students towards ChatGPT usage," *International Journal of Technology in Education*, vol. 7, no. 1, pp. 86–106, 2024, doi: 10.46328/ijte.583.
- [12] J. Blahopoulou and S. Ortiz-Bonnin, "Student perceptions of ChatGPT: Benefits, costs, and attitudinal differences between users and non-users toward AI integration in higher education," *Education and Information Technologies*, vol. 30, no. 14, pp. 19741–19764, 2025, doi: 10.1007/s10639-025-13575-9.
- [13] B. Klimova, P. Bachmann, and D. Frutos-Bencze, "The use of ChatGPT in academia: Perspectives of higher education students," *Cogent Education*, vol. 12, no. 1, Art. no. 2508216, 2025, doi: 10.1080/2331186X.2025.2508216.
- [14] K. Aruleba, I. T. Sanusi, G. Obaido, and B. Ogbuokiri, "Integrating ChatGPT in a computer science course: Students perceptions and suggestions," arXiv:2402.01640, 2023. Available: <https://arxiv.org/abs/2402.01640>
- [15] S. Lau and P. 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 *Proc. 2023 ACM Conf. on International Computing Education Research (ICER)*, vol. 1, 2023, pp. 106–121, doi: 10.1145/3568813.3600138.