

M. Kaminska, BA student
O. Pasichnyk, PhD in Ped., As. Prof., language advisor
Khmelnytskyi National University

THE IMPACT OF CHATBOT TECHNOLOGIES ON STUDENTS' PROGRAMMING SKILLS

Artificial intelligence (AI) is now one of the most prominent technologies. Its impact on educational practice is vast, influencing education across the whole world. It allows students to generate code from natural language, explain concepts, find errors, and suggest solutions changing the learning process in both introductory and advanced programming courses. While these assistants make tasks faster, they also raise questions about the development of core programming skills.

The topic is highly relevant because many students use AI tools daily for homework and projects. Unlike traditional syntax checkers, generative AI acts like a personal tutor that is always available. As a result, students can complete tasks quickly but may not develop deep understanding and independent thinking [2].

AI code generators offer clear benefits. Beginners often struggle with basic mistakes. ChatGPT and Copilot provide immediate correct examples and explanations. Scientists [3] emphasize that this reduces frustration, increases motivation, and makes programming more accessible for students from different backgrounds. These tools also enable personalized learning. Students receive answers adjusted to their level and can focus on higher-level concepts rather than minor errors.

However, the serious problem of applying AI in education is that students may have less logical thinking, especially in mathematics. Before creating good software, a student needs to develop strong logical thinking. Normally, it is built in subjects like Discrete Mathematics, Linear Algebra and Calculus. In these courses, logic is gained when solving the problems, understanding the formulas and thinking step-by-step [1]. When students use AI to quickly solve tasks or write proofs, they do not practice this thinking process. Because of this, they may find it difficult to solve real problems and turn them into programming tasks.

Another problem is the difference between writing code and understanding how it works. AI tools can easily generate correct code. However, in beginner programming courses, especially with languages like C++, students need to understand how computers work. Over-reliance on AI means they miss practice in debugging memory leaks or segmentation faults and may lack structural understanding needed for professional work [2].

Besides, using AI too much can be risky because AI can make mistakes. Sometimes it produces code that looks correct but has hidden errors. If students do not have enough knowledge, they cannot check this code properly. This means that instead of helping, AI can create new problems. That is why education should not only focus on writing code but also on checking it carefully and understanding how it works.

AI tools can negatively affect spatial thinking and creative design. In frontend development, students need to visualize system architecture. Early heavy use of Copilot for generating HTML or CSS may prevent them from building their own mental models and limit their ability to design complex applications [2].

Moreover, constant AI assistance can reduce the “flow” state and lower motivation. Mastery in programming develops through struggling with difficult tasks. Instant solutions may decrease persistence and leave students less prepared for advanced unstructured projects [1]. All together it can weaken technical communication skills. Programming is not only about writing code. Students also need to explain their ideas, discuss solutions, and cooperate with others.

Academic integrity remains a key issue. Clear institutional rules and new assessment methods, such as oral explanations and process portfolios, are needed to ensure responsible use [4].

In conclusion, ChatGPT and GitHub Copilot have a complex impact on students’ programming skills. They can accelerate and personalize learning when used properly. At the same time, over-reliance risks weakening logical thinking, debugging abilities, persistence, and communication skills. A balanced approach with clear guidelines, critical tasks, and ongoing research is essential to prepare future programmers who think deeply and work responsibly in an AI-supported world.

REFERENCES

1. Becker B. A., Denny P., Finnie-Ansley J. та ін. Programming is hard – or at least it used to be: educational opportunities and challenges of AI code generation [Електронний ресурс] // Proceedings of the 54th ACM Technical Symposium on Computer Science Education (SIGCSE 2023). – 2023. – Режим доступу: <https://arxiv.org/abs/2212.01020> (дата звернення: 30.03.2026).
 2. Prather J., Denny P., Leinonen J. та ін. The robots are here: navigating the generative AI revolution in computing education [Електронний ресурс] // Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education (ITCSE-WGR 2023). – 2023. – Режим доступу: <https://arxiv.org/abs/2310.00658> (дата звернення: 30.03.2026).
 3. Kazemitabaar M., Chow J., Ma C. K.. Studying the effect of AI code generators on supporting novice learners in introductory programming [Електронний ресурс] // Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI 2023). – 2023. – Режим доступу: <https://arxiv.org/abs/2302.07427> (дата звернення: 30.03.2026).
- Lau S., Guo P. J. From “banned” to “required”: shaping computer science education in the era of generative AI [Електронний ресурс] // Proceedings of the 2023 ACM Conference on International Computing Education Research (ICER 2023). – 2023. – Режим доступу: <https://doi.org/10.1145/3568813.3600138> (дата звернення: 30.03.2026).