

**DOI 10.36074/logos-06.02.2026.047**

## **USING CHATGPT IN IT COURSES: PEDAGOGICAL APPROACHES AND IMPLEMENTATION EXPERIENCE**

**Bazhan Stanislav<sup>1</sup>**

---

1. PhD in Applied Mathematics  
Teacher of computer and mathematical disciplines  
*Separate structural unit "Technological Professional College of Dniprovsky State Technical University", UKRAINE*  
**ORCID ID: 0000-0003-2228-9389**

---

The integration of intelligent assistants, particularly ChatGPT, into the programming learning process opens new perspectives for the individualization of education and the provision of real-time student support. Recent studies increasingly emphasize that such systems can create an environment of continuous access to knowledge, facilitate the rapid correction of errors, and even foster self-directed learning skills. However, the effectiveness of this tool largely depends on how pedagogically well it is integrated into the curriculum.

Empirical experience gained during practical classes in the discipline Programming (using the C# language) demonstrates both the advantages and the potential risks of using ChatGPT. The learning process followed a progressive structure: from creating simple console applications to developing Windows Forms applications incorporating elements such as buttons, text fields, tables, charts, and timers. Students were encouraged to use ChatGPT to address challenges arising during code writing, to search for optimal solutions, and to perform code refactoring. This approach enabled the rapid correction of syntactic and logical errors, facilitated the creation of more structured and concise code, and increased the pace of task completion.

At the same time, without proper pedagogical guidance, the use of ChatGPT may lead to undesirable effects. Most commonly, this was manifested in solutions that extended beyond the studied material. Students who had not yet mastered certain concepts or principles were sometimes offered by the AI to use constructs or libraries that were either not included in the current syllabus or not recommended for use at this stage. This created the risk of superficial learning, as

the completion of assignments became more a matter of technically copying existing code rather than engaging in a conscious problem-solving process.

Another challenge arose when students received excessively large and complex code snippets. Due to their limited experience, they were often unable to understand the program's logical structure or the interactions between its components, which reduced the intended learning effect of solving the task. This supports the findings of contemporary research suggesting that uncontrolled use of generative models can contribute to the so-called "illusion of knowledge," where learners are able to present correct answers or working code but cannot explain the underlying logic.

In the course of implementation, a number of pedagogical interventions proved highly effective in improving learning outcomes. A key element was the prior specification of the subject context of the query: before addressing ChatGPT, students formulated explicit constraints, emphasizing that the solution must be based solely on the syntax already studied and must not include external libraries or technologies that had not been covered in class. This significantly reduced the risk of deviating from the curriculum and helped maintain focus on the intended educational objectives.

Another effective strategy involved framing queries as requests for step-by-step hints rather than for complete code solutions. This method stimulated reflection, activated higher-order thinking, and encouraged independent problem solving. A mandatory element after interacting with ChatGPT was written reflection, in which students explained the logic of each functional block of the received code. This allowed for a clearer assessment of the actual level of understanding and helped to prevent mechanical copying.

For more complex tasks, it proved beneficial to break them into smaller subproblems, allowing students to seek AI assistance for specific aspects only, such as event-handling logic for button clicks, table population, or input validation. The final stage of the process involved a mandatory code review with the instructor, who asked clarifying questions and required oral explanations of the solution's logic. This significantly reduced the illusion of knowledge and promoted deeper retention of the material.

The results of the practice confirmed that when these rules were followed, improvements were observed in code quality, conciseness, and efficiency, along with increased task completion speed and the development of students' metacognitive skills. Conversely, when students engaged in uncontrolled copying of large code fragments from the assistant, their understanding dropped considerably. These findings lead to the conclusion that ChatGPT is an effective tool only when used under pedagogical constraints and supervision, and that its

**ABSCHNITT 21.**  
PÄDAGOGIK UND BILDUNG

optimal role lies in functioning as a mentor—providing hints, explanations, and refactoring support—rather than as a substitute for the learning process itself.

The next steps in developing this methodology include creating a set of subject-specific prompt templates for the C# language, developing guidelines for the ethical and methodical use of AI in education, and conducting controlled studies to assess the long-term impact of these approaches on knowledge retention. Integrating these principles into the educational process can not only improve the effectiveness of programming education but also foster strong critical thinking skills essential in the modern IT industry.