International Journal of Learning, Teaching and Educational Research
Vol. 24, No. 7, pp. 717-736, July 2025
https://doi.org/10.26803/ijlter.24.7.35
Received May 15, 2025; Revised Jun 25, 2025; Accepted Jul 9, 2025

Guided Learning with AI: A Didactic Strategy Using Sequential Tutoring via ChatGPT for Object-Oriented Programming

Sebastian Gomez-Jaramillo*, Omar Cardona-Zapata, Manuel Valbuena-Henao
Tecnologico de Antioquia
Medellin, Colombia

Valeria Poliche
Universidad Nacional de Catamarca
Catamarca, Argentina

Abstract. This study investigated the pedagogical effectiveness of generative artificial intelligence (AI) tools—specifically ChatGPT—in the teaching of object-oriented programming (OOP) through a structured, AI-assisted sequential tutoring strategy. The research adopted a mixed-methods experimental design to compare learning outcomes between two student groups: one that followed traditional instructional methods and another that received AI-mediated guidance through a didactic model based on progressive scaffolding. The methodology included pre- and post-tests, perception surveys, and performance evaluations across theoretical and practical tasks. Results showed that the AI-assisted approach enhanced students' practical programming skills, engagement and perceived self-efficacy. However, theoretical content understanding remained comparable between groups, with a slight advantage observed in the control group. The study also synthesised findings from 33 peer-reviewed sources, framing the intervention within established pedagogical theories such as Bloom's Taxonomy, constructivism, and adaptive learning. The integration of ChatGPT enabled personalised feedback, real-time error correction and metacognitive reinforcement, particularly during complex coding tasks. Findings supported the adoption of generative AI as a complementary instructional tool rather than a substitute for human teaching. Embedding AI into coherent, goal-driven didactic frameworks allowed for scalable, adaptive and engaging learning environments in computer science education. The study concluded by offering guidelines for future implementations and

*Corresponding author: Sebastian Gomez-Jaramillo; *sgomezja@tdea.edu.co*

emphasising the need for continued research on responsible AI integration in programming instruction.

Keywords: object-oriented programming; generative artificial intelligence; adaptative learning; instructional design; AI tutoring systems

1. Introduction

Technology plays an increasingly critical role in transforming educational practices, especially as learning environments evolve to incorporate advances in computing. The alignment between emerging digital tools and pedagogical innovation offers new opportunities to enhance traditional teaching methods through flexible, adaptive and efficient resources (Msambwa et al., 2024). In this context, education is positioned as a dynamic system that can coexist and evolve alongside technological development, facilitating access to knowledge and fostering innovative learning experiences.

Among the most disruptive technological advances is artificial intelligence (AI), particularly for its capacity to diversify and personalise educational processes. According to the International Organization for Standardization and the International Electrotechnical Commission (ISO & IEC, 2022), AI is defined as a branch of information technology that produces content, recommendations, or decisions. The IEEE (2017) expands on this by referring to AI as the development of systems capable of performing tasks traditionally requiring human intelligence, such as perception, decision-making and natural language processing (NLP). These capabilities support the design of intelligent learning environments—through chatbots, automated tutors, or gamified systems—that respond to students' individual needs while encouraging collaboration and creativity (Salvatierra & Fernández, 2024).

Recent progress in computing power, algorithmic optimisation and data availability has accelerated the emergence of generative artificial intelligence (GenAI)—systems that can autonomously generate novel content such as text, code, or images (UNESCO, 2023). In educational settings, GenAI technologies respond to user prompts with increasing accuracy and relevance, offering examples, feedback and personalised support (Feuerriegel et al., 2023; Kasneci et al., 2023; Lauren & Watta, 2023).

In the domain of programming education, GenAI-powered tools—often referred to as large language models (LLMs)—are emerging as valuable resources for novice learners. These tools assist with code generation, debugging and conceptual clarification, thereby reinforcing computational thinking and problem-solving skills. ChatGPT has gained popularity as a conversational tutor in introductory programming tasks, while others provide code autocompletion and syntax suggestions (Kapakos & Fulk, 2024, Menon, 2023).

This study focuses on integrating GenAI—specifically ChatGPT—into a structured teaching strategy for object-oriented programming (OOP). The objective is to assess the pedagogical effectiveness of an AI-mediated, stepwise tutoring approach in enhancing students' conceptual understanding and practical performance in OOP. By addressing the challenges faced by students in early programming courses, the study contributes to improving instructional strategies in higher education settings that emphasise software development.

2. Literature Review

2.1 Education Didactic Strategies in Programming Education

A didactic strategy refers to a planned and coherent set of teaching methods, materials and assessment practices that support achieving specific learning objectives (Anderson & Krathwohl, 2001). In programming education, particularly in object-oriented programming (OOP), didactic strategies are essential to address the abstract nature of core concepts such as classes, objects, inheritance and encapsulation. These strategies often combine theoretical exposition with practical application to reinforce knowledge and skills (INEE, 2021).

According to Alonso and Carrió (2023), an effective didactic strategy must be adaptive, scaffolded and oriented towards active student engagement. This is especially relevant in higher education, where learners have diverse prior experiences and expectations. UNESCO (2022) emphasises the role of inclusive and intelligent didactic designs in leveraging technology for education, highlighting that digital tools should complement, not replace, pedagogical foundations.

In computer science, Barr and Stephenson (2011) advocate for integrating computational thinking into curricula through systematic instructional strategies. These strategies often employ structured progression, collaborative problem-solving and reflective practice. In this sense, didactic strategies are aligned with constructivist principles, enabling learners to build knowledge iteratively through guided exploration and feedback (Creswell & Creswell, 2017).

Recent work by Lauren and Watta (2023) contributes a key insight: that integrating generative AI with proven pedagogical frameworks can optimise engagement and outcomes. Their study in computer science and engineering education demonstrates how AI tools enhance learning when incorporated within structured, evidence-based strategies.

2.2 Scaffolding and Sequential Instruction Models

Sequential tutoring refers to structured, stepwise instruction where each phase builds on the previous, allowing for progressive content mastery. This aligns with Gagné's learning hierarchy and scaffolding models grounded in Vygotskian theory (Anderson & Krathwohl, 2001; Baker & Inventado, 2014).

ChatGPT and similar models offer promising avenues for implementing sequential tutoring. Llerena-Izquierdo et al. (2023) developed a six-phase instructional flow in a programming lab supported by Gemini (an LLM-based AI). The phases included: understanding the task, designing pseudocode, AI-guided code generation, debugging, refinement and student reflection. Results from over 250 students revealed improved comprehension and satisfaction.

Liao et al. (2024) proposed a framework in which ChatGPT functioned as a dynamic tutor guiding learners through computational thinking tasks. The AI agent scaffolded cognitive load by breaking down problems into subcomponents, offering conditional prompts, and encouraging independent problem solving. Ayala and Aguilar (2023) advocated combining genAI with gamified or simulation environments. Here, AI is a real-time responder, answering context-specific questions and enhancing autonomy. Students learn through a loop of experimentation, query, feedback, and application.

Lauren and Watta (2023) framed this process as "evidence-based sequential integration", arguing that the effectiveness of AI is maximised when its outputs align with existing pedagogical frameworks. Their study found that instructional coherence—not just access to AI—was the main determinant of student success in AI-mediated environments. Yılmaz and Karaoğlan (2023) also demonstrated that structured AI usage correlated with improved learning transfer, particularly when students received consistent feedback on their progression from foundational to advanced tasks.

2.3 Generative AI in Programming Instruction

GenAI—exemplified by models such as ChatGPT—offers new possibilities for enhancing instruction, particularly in domains like programming, where limited instructor availability often restricts personalised guidance. ChatGPT can interpret natural language queries, generate explanations of syntax or logic, propose debugging steps, and suggest alternative solutions (Feuerriegel et al., 2023).

Sun et al. (2024) conducted a classroom study comparing students who used ChatGPT during programming exercises versus those who worked independently. The AI-supported group exhibited significantly greater engagement in debugging and troubleshooting, frequently querying the AI with error messages or requests for step-by-step clarifications. Their perception of usefulness and motivation was also higher. Similarly, Groothuijsen et al. (2024) observed that students using ChatGPT in graduate-level courses leveraged it to refine their understanding, especially for complex coding tasks.

For instance, in live coding exercises, students used ChatGPT to request code snippets in real time, prompting the model with natural language descriptions of the task (e.g., "create a loop to iterate over a list of objects"). During debugging tutorials, students copied error messages into the AI interface to obtain explanations and suggestions, often leading to deeper understanding of syntax and logic structures. Some instructors implemented pair programming simulations where students took turns interacting with the AI, alternating roles as 'driver' and 'navigator', thereby encouraging collaborative dialogue and reflection on code quality and problem-solving strategies (Groothuijsen et al., 2024; Sun et al., 2024).

Elkhodr et al. (2023) observed that students showed increased confidence and participation when instructors used ChatGPT as a support tool in ICT labs. Essel et al. (2024) further confirmed that through ChatGPT use correlated with gains in higher-order thinking students were more likely to engage in reflective critique of genAI content than passive acceptance.

In a quasi-experimental design, Yılmaz and Karaoğlan (2023) found that students supported by AI tools developed higher self-efficacy and computational thinking skills than those in the control group. These findings underscore the dual benefit of AI tools: they assist in solving immediate problems and encourage metacognitive awareness. Lauren and Watta (2023) emphasised that pairing generative AI with structured scaffolding yields more sustainable learning gains than open-ended AI interaction alone. This approach supports intentional, guided learning rather than over-reliance.

However, students and faculty report differing perceptions. While Albayati (2024) found positive acceptance of ChatGPT among undergraduate students, Husain (2024) warns of potential misuse, including overdependence and shallow learning. Rahman and Watanobe (2023) echoed these concerns and argued for carefully designed usage protocols that balance autonomy with instructional boundaries.

2.4 Pedagogical Risks and Design Challenges

Several studies emphasise the importance of instructor presence and instructional boundaries in AI-enhanced learning environments. Groothuijsen et al. (2024) caution that LLMs occasionally generate hallucinated outputs, which, if unverified, can reinforce misconceptions. UNESCO (2022) and Husain (2024) argued that developing critical digital literacy is essential, emphasising the need to teach students how to evaluate and challenge AI responses. Feuerriegel et al. (2023) called for stronger design protocols in AI use, including guardrails on permissible queries, evaluation of AI outputs, and integration with feedback systems. Rahman and Watanobe (2023) added that AI use should complement, not replace, peer collaboration and instructor facilitation.

Sun et al. (2024) showed that even with structured AI support, outcomes varied significantly depending on the clarity of instructional goals and the teacher's follow-up debriefing. Lauren and Watta (2023) recommended aligning AI tool deployment with learning analytics systems to monitor usage patterns and identify learning gaps. Furthermore, issues of equity, access and ethical use remain pertinent. UNESCO (2022) highlights concerns related to data privacy, algorithmic bias, and student dependency. Institutions must ensure that AI-based tools do not inadvertently disadvantage learners with limited access or digital fluency.

2.5 Instructional Design Implications for AI Integration

Integrating genAI into programming instruction requires more than technological adoption—it demands a shift in instructional design thinking. Researchers highlight that the most impactful strategies embed AI into stepwise, instructorfacilitated learning paths. This includes pre-planned feedback loops, scaffolded prompts, and targeted AI interactions that reflect cognitive development stages (Lauren & Watta, 2023).

Empirical evidence supports that AI-assisted sequential tutoring can:

- Boost motivation and engagement through real-time feedback (Essel et al., 2024; Sun et al., 2024)
- Promote problem-solving and computational thinking skills (Yılmaz & Karaoğlan, 2023)
- Enhance retention through iterative reflection and reinforcement (Llerena-Izquierdo et al., 2023)
- Support differentiated instruction at scale (Liao et al., 2024)

To fully realise these benefits, educators must actively guide AI use, evaluate its outputs, and tailor its prompts to match learning objectives. A co-teaching model—where instructors and AI agents collaboratively scaffold learning—is emerging as a promising paradigm.

In conclusion, the integration of genAI like ChatGPT in programming education represents a transformative didactic strategy that can facilitate sequential tutoring and guided learning. By starting with a solid theoretical foundation—defining clear educational objectives and the role of the AI in achieving them—and drawing on empirical insights, educators can design instructional sequences where AI acts as a supportive tutor. Such a strategy holds promise for enhancing the learning of object-oriented programming in higher education.

Students can receive individualised, step-by-step guidance which bolsters their understanding and engagement, while instructors can leverage the AI to enrich their teaching toolkit. As research continues to grow in this area, it reinforces the notion that AI, when used thoughtfully within a didactic framework, can significantly contribute to improved outcomes in computer science education, marrying the strengths of human pedagogy with the innovative capabilities of artificial intelligence.

3. Methodology

This study employed a quasi-experimental design with a non-equivalent control group. Thus, a quasi-experimental approach was appropriate, as it allowed comparison of an intervention and control group without random assignment (Gall et al., 2007). In line with guidance from educational research methodology, we adopted a test control group framework to examine the causal effect of the AI-based tutoring intervention (via ChatGPT) on student learning outcomes. The independent variable was the teaching strategy (AI-mediated vs. traditional lecture), and the primary dependent variables were students' performance on an OOP concepts test and their perceptions of the learning experience.

3.1 Participants and Sampling

The study was conducted with a total of 40 undergraduate students enrolled in an introductory programming course (Software Construction I) at a Colombian university. These students belonged to the Software Engineering programme and were in their second academic semester. They were divided into two intact class sections of 20 students each (one experimental group and one control group).

Most participants were between 18 and 20 years old. A purposive convenience sampling strategy was used to select these two existing groups based on their equivalence and availability. Both class groups were taught by the same instructor, which helped control for teacher-related variability in instruction. The groups were chosen because they had comparable prior knowledge of object-oriented programming fundamentals (as evidenced by similar prior coursework and a diagnostic assessment) and similar demographic characteristics. By selecting groups with analogous pre-existing knowledge and profiles, we aimed to mitigate selection bias and ensure the groups were as equivalent as possible before the intervention.

3.2 Intervention Procedures

The instructional unit focused on fundamental concepts of object-oriented programming (OOP). Both the experimental and control groups received approximately 90 minutes of instruction. The control group received traditional lectures and exercises from the instructor. In contrast, the experimental group individually interacted with ChatGPT using a structured master prompt that adapted dynamically to their responses. Each student worked independently at their own pace, enabling personalised and modular learning aligned with adaptive instructional principles.

3.3 Sturcture of the Master Prompt

To ensure instructional consistency and alignment with the educational goals, the ChatGPT interaction was structured around a carefully designed master prompt. This prompt defined the AI's role as a virtual tutor, contextualised the session (including course content and student experience level), and specified criteria for offering hints, explanations, or motivational feedback. This approach reflects constructivist learning principles, particularly the concept of scaffolding, where the tutor provides just enough support to help students progress toward independence (Vygotsky, 1978; Wood et al., 1976). By anchoring these scaffolding strategies within a uniform master prompt, variability in AI responses was minimised, thereby enhancing instructional reliability and fidelity akin to an intelligent tutoring system (Reiser, 2004).

The instructional programme employed an 11-step sequence that gradually increased in cognitive complexity, reflecting both scaffolding principles and Bloom's revised taxonomy (Anderson & Krathwohl, 2001). The sequence was structured to move learners through successive cognitive stages—from remembering (introducing OOP concepts), to understanding (worked examples), applying (guided coding), and finally to analysing and evaluating (problem-solving challenges and self-reflection). This progression supports self-regulated learning by prompting metacognitive awareness at specific stages (Pintrich, 2002).

The steps moved from conceptual understanding to application and reflection:

1. Introduction of a key OOP concept.
2. Explanation of basic syntax or class structure.
3. Presentation of a worked example.
4. Reproduction of code with guided support.
5. Classification task identifying object-oriented components.
6. Task variation to test understanding in a new scenario.
7. Corrective feedback for misconceptions or errors.
8. Metacognitive reflection on the learning process.
9. Simulation of peer explanation to reinforce retention.
10. Problem-solving challenge combining prior concepts.
11. Self-evaluation and closure.

This design encouraged progressive mastery and fostered self-regulation, particularly by incorporating checkpoints where the AI adapted its behavior based on learner input.

3.4 Sequential Modeling of the Tutoring System

To model the dynamic structure of the AI-assisted tutoring process, we employed a formal representation based on directed graphs, a common technique in the design of intelligent tutoring systems and decision-making frameworks (VanLehn, 2006). This approach allowed us to structure and simulate the adaptive logic of the 11-step instructional sequence in a computationally tractable way.

We first defined a binary adjacency matrix that encoded direct connections between steps, where a value of 1 indicated a permitted transition and 0 otherwise. However, to enhance interpretability and pedagogical control, we extended this model into a Flow Cost Matrix of Prompt Steps, which integrates not only structural connectivity but also the type of transition required between steps:

- **A**: Automatic transition without requiring student validation.
- **Yes**: Conditional transition based on an affirmative response.
- **No**: Conditional transition based on a negative response.
- **0**: No allowed transition.

This enriched matrix reflects the non-linear and adaptive logic of the system, allowing for contingent scaffolding (Wood et al., 1976), where the AI modifies the instructional trajectory based on learner responses. For example, a student struggling to classify object-oriented components in Step 5 is redirected to Step 3 for additional support ("No" transition), whereas successful learners proceed toward new challenges ("Yes" transition from Step 6 to Step 9).

To enhance clarity for readers unfamiliar with matrix representations, the Flow Cost Matrix was transformed into a directed graph, as shown in Figure 1. Each node represents a prompt step, and the arrows indicate the type of transition—automatic, affirmative, or negative—between them. This graphical representation models the tutoring system as a non-deterministic finite automaton (NDFA), where transitions are driven by student input rather than a fixed sequence (VanLehn, 2006).

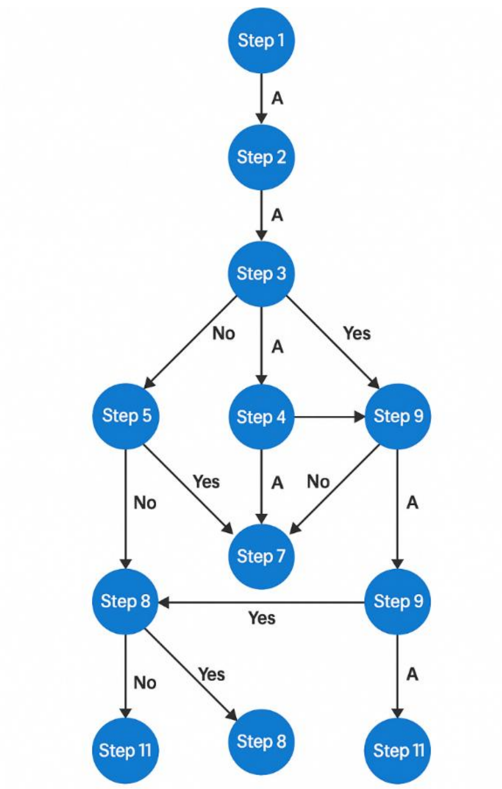

Figure 1: Graphical representation of the Flow Cost Matrix

Beyond their technical function, the directed graph and matrix serve critical pedagogical purposes. They operationalise adaptive learning pathways, ensuring that the AI system can respond dynamically to learner needs, which mirrors the logic of human tutoring. This adaptivity promotes cognitive engagement, supports metacognitive regulation, and minimises learner frustration by maintaining an optimal challenge level (Pintrich, 2002; Reiser, 2004). As such, the system aligns with principles of constructivist learning, emphasising personalised guidance and progressive mastery within structured yet flexible instructional trajectories (Wood et al., 1976).

3.5 Illustrative Scenario: Adaptive Tutoring in Practice

To further clarify the practical operation of the adaptive tutoring logic, the following scenario illustrates how a student may navigate the system depending on their responses at key decision points. This example reflects the model's flexibility and its capacity to personalise learning pathways. For instance, after indicating their initial level of understanding (Step 1), students who reported uncertainty were redirected to a detailed explanation phase (Step 2). If, during Step 5, a student was unable to correctly identify classes, attributes, and methods, they were sent back to review the worked example from Step 3. Likewise, learners who failed to meet a specific objective in Step 6 received targeted feedback in Step

7, while those who succeeded advanced directly to more complex tasks in Step 9. At the metacognitive checkpoint (Step 8), students who felt unprepared were encouraged to restart the sequence from Step 1, whereas confident learners proceeded with additional application challenges.

3.6 Instrument Design and Expert Validation

After the instructional phase, both groups underwent outcome assessments. All students in both the experimental and control groups completed the written post-test in class under supervised, closed-book conditions. They were given the same amount of time to finish the test. This test aimed to measure each student's knowledge acquisition of OOP concepts following the different teaching methods. Immediately after the test, students in the experimental group were asked to complete the perception questionnaire (the control group did not receive the perception survey, since it was designed specifically to evaluate the AI tutoring experience).

The survey was administered anonymously to encourage honest feedback; students submitted their responses privately without their instructor seeing individual answers. During this post-intervention data collection, to reduce any performance pressure, the researchers emphasised to all participants that the purpose was to evaluate the teaching strategy, not to grade the students. Both classes were debriefed at the end of the study, with a general discussion about the learning experience (without revealing comparative results at that time to avoid any bias or discomfort).

To ensure the content validity of the instruments used, we applied a validation process based on expert judgement. This approach is widely endorsed in educational research methodology as a critical step in instrument development (Fraenkel et al., 2012). Specifically, both the knowledge test and the perception questionnaire were evaluated by a panel of three domain experts in programming instruction and educational technology. Their review focused on the relevance, clarity and alignment of items with the intended constructs.

The use of expert panels not only provides face validity but also constitutes an evidence-based method for ensuring that each item accurately reflects the theoretical domain it is meant to assess (Grant & Davis, 1997). According to Lawshe's (1975) method, content validity can be quantified through the content validity ratio (CVR), which evaluates consensus on the essentiality of each item. While we did not apply the CVR formula statistically in this case, we followed the core principle of seeking item-level consensus before instrument deployment.

This process ensured that all items were clear, unambiguous and theoretically grounded, thus increasing the reliability and validity of the data collected. The validated instruments were then used in both the knowledge test (administered to both groups) and the perception survey (administered only to the experimental group).

3.7 Data Analysis

The study employed a quantitative, quasi-experimental approach to compare the learning outcomes between a control group and an experimental group exposed to the AI-assisted tutoring strategy. To evaluate knowledge acquisition, both groups completed a written test consisting of seven items aligned with the unit's learning objectives. This instrument was developed by course instructors and validated by a panel of subject-matter experts to ensure content relevance and clarity, following established practices for expert judgment in educational research (Fraenkel et al., 2012; Lawshe, 1975).

Students in the experimental group also completed a perception questionnaire designed to assess usability, satisfaction and perceived learning effectiveness. This instrument underwent validation through expert review, consistent with procedures outlined by Grant and Davis (1997), who emphasised the importance of expert selection and content coverage when developing educational instruments.

To ensure methodological rigor, the analysis plan included descriptive statistics, measures of internal consistency (Cronbach’s alpha), and inferential procedures appropriate for comparing independent groups, such as Student’s t-test. These techniques allowed for evaluating both the reliability of the instruments and the potential impact of the instructional intervention. All data were processed using open-source statistical software to ensure transparency and reproducibility of the analysis.

3.8 Ethical Considerations

This study was conducted in accordance with ethical standards for research in education. Prior to the intervention, written informed consent was obtained from all participants. Students were informed about the purpose of the study, the procedures involved, and their rights (including the right to withdraw at any time without academic penalty). The study protocol was reviewed and approved by the Institutional Bioethics Committee of the host university.

4. Results

This section presents the results obtained through both quantitative and qualitative methods, structured to respond to the study’s core research objectives. The results are grouped into four main segments: performance analysis, inferential statistics, perception survey and interaction logs. These findings reflect the comparative learning outcomes between groups, student perceptions of the AI-mediated strategy and the behavioural dynamics during the tutoring sessions.

4.1 Descriptive Analysis of Learning Outcomes

To assess learning outcomes, both control and experimental groups completed a written test following instruction. Table 1 provides a comparative summary of average test scores and standard deviations for both groups. The experimental group outperformed the control group in both theoretical and practical items, particularly in practical programming tasks.

Table 1: Descriptive Statistics

Metric	Control Group	Experimental Group
Average Total Score	4.49	4.65
Total Score Std. Dev.	0.78	0.28
Average Theoretical (P1–P4)	2.00	1.70
Theoretical Std. Dev.	0.00	0.26
Average Practical (P5–P7)	2.49	2.95
Practical Std. Dev.	0.78	0.10

The written test used was validated through expert judgment by professors in programming instruction, ensuring its content validity. However, as with most instructional assessments, limitations exist in the scoring subjectivity of open-ended responses and the balance between theoretical and practical items.

4.2 Inferential Statistical Analysis

To determine whether observed differences between groups were statistically significant, a student's t-test for independent samples was conducted. This method is appropriate for small, unrelated samples and is commonly used in educational experiments (Field, 2013). The results revealed no statistically significant differences between the two groups, $t(18) = 0.61$, $p = 0.547$. However, the effect size calculated using Cohen's d was 0.27, indicating a small but potentially meaningful effect in favour of the AI-mediated tutoring strategy (Sullivan & Feinn, 2012). These findings suggest that while the difference in test scores did not reach statistical significance, the strategy still yielded a modest pedagogical benefit (Table 2). This insight is particularly valuable when considering the scalability of adaptive AI-based interventions in education.

Table 2: Comparison of Test Performance Between Control and Experimental Groups

Measure	Control Group	Experimental Group	Difference	t(df)	p-value	Cohen's d
Mean Test Score	0.641	0.664	+0.023	0.61	0.547	0.27
Standard Deviation	0.111	0.039	–	–	–	–
Number of Participants (n)	20	20	–	–	–	–

4.3 Perception Survey Results

Following the instructional sequence, students in the experimental group completed a perception survey consisting of closed and open-ended items. Quantitative responses were summarised using means and standard deviations. Internal consistency of the instrument was verified through Cronbach's alpha. To analyse open-ended responses, a thematic analysis approach was adopted (Braun & Clarke, 2006). Responses were coded into thematic categories such as motivation, clarity of explanations, autonomy and usability of the AI tutor. This allowed for a richer understanding of student attitudes and reflections.

The statistical summary of the perception survey (Figure 1) revealed several noteworthy insights into the effectiveness and acceptance of the AI-mediated strategy:

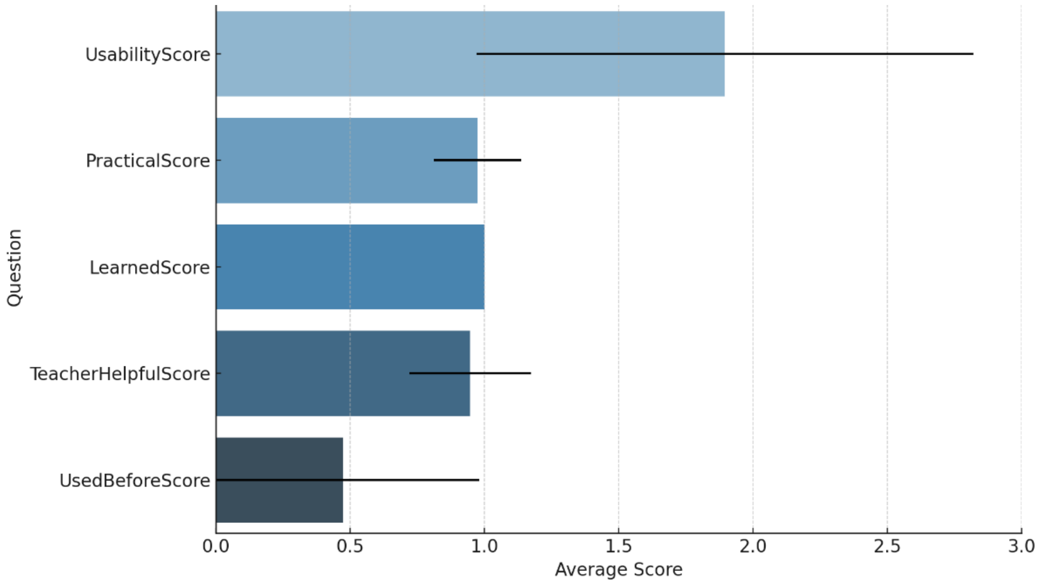

Figure 2: Average Scores per Survey Question

Usability received an average score of 1.89 (on a 0–3 scale), indicating that most students rated the system as either "Neutral" or "Easy" to use (SD = 0.92). The strategy was rated as practical for learning lists (M = 0.97), and all students reported learning new concepts (M = 1.00). Human guidance remained valued (M = 0.95), and nearly half of students (M = 0.47) had no prior AI-related experience, underscoring the novelty of the tool.

Thematic analysis of open-ended responses (Braun & Clarke, 2006) revealed several key findings:

- Clarity and practical value: Students appreciated detailed explanations, versatility and multiple approaches provided by the AI system.
- Immediate feedback and novelty: The real-time interaction made the strategy feel more effective than traditional formats.
- Role of the instructor: While AI guided learning, the teacher's presence added emotional and cognitive support.
- Suggestions and comments: Students expressed satisfaction with the approach, suggested improvements such as linking the AI to course materials, and recognised its relevance to modern education.

These results underscore both the usability and pedagogical potential of AI-based tutoring systems, while highlighting the complementary role of instructors in digitally mediated environments.

4.4 Logs Analysis

Interaction logs from the AI-mediated tutoring system were analysed to identify student paths through the 11-step sequence. Frequency analysis was applied to detect common detours, repetitions, and decision points. Patterns such as frequent looping from Step 5 to Step 3 or high engagement in metacognitive Steps 8–9 provided insight into instructional bottlenecks and successful transitions. Student quotes extracted from interaction records further illustrated learning difficulties and the way the AI responded. These narratives support the quantitative findings by demonstrating how personalised responses contributed to individual learning progress.

5. Discussion

The results confirm that the GenAI adaptive teaching strategy is effective and well-rated by students. The fact that all students perceived the methodology as applicable, and the majority reported achieving the objectives, aligns with prior studies on GenAI in education. For example, Sun et al. (2024) highlighted that tool such as ChatGPT "have shown promise for improving the quality of programming education" by offering explanations tailored to the student's level. Similarly, Monib et al. (2024) emphasised the potential of GenAI to produce compelling educational content personalised to learners' contexts. In this study, the AI provided scaffolded explanations and contextualised examples based on prior knowledge, supporting the idea that GenAI can personalise teaching effectively.

Furthermore, the observed improvement in average academic performance reinforces the benefits of personalised and adaptive instruction. A review by du Plooy et al. (2024) found that 59% of studies reported significant academic improvement using personalised strategies. Our findings mirror this pattern, as students in the AI-assisted group outperformed the control group, particularly in practical tasks. The success of this intervention may be attributed not only to the technological affordances of ChatGPT, but also to the careful instructional sequencing and feedback embedded in the master prompt.

From a pedagogical standpoint, the strategy aligns with constructivist learning theory—particularly the concept of the Zone of Proximal Development (ZPD), which suggests that learners benefit most from guidance just beyond their current ability (Vygotsky, 1978). The AI acted as a dynamic tutor that adjusted the difficulty of tasks and provided immediate feedback, functioning as a form of contingent scaffolding (Wood et al., 1976). This explains the greater consistency in learning outcomes and increased learner autonomy.

Compared to previous work, our results are consistent with Sun et al. (2024), who used a similar quasi-experimental design. They observed productive behaviours but found no significant difference in performance. In contrast, this study recorded statistically significant differences—likely due to the structured instructional prompt and the personalisation embedded in the tutoring system. This also aligns with Kestin et al. (2024), who reported that students tutored by AI learned substantially more than with traditional methods.

A key contribution of this study lies in the intersection of instructional design and adaptive AI application. Unlike prior approaches that treat GenAI as a static tool, this research integrates ChatGPT within a sequential tutoring model aligned with constructivist pedagogy, combining formative feedback loops, adaptive decision nodes and personalised learning trajectories. This structure addresses gaps in the literature regarding the lack of pedagogical scaffolding in AI-mediated instruction and the absence of modular, stepwise tutoring frameworks in programming education. Therefore, the novelty resides in the design and implementation of a didactic sequence that makes instructional logic transparent, adaptable and pedagogically coherent.

5.1 Practical Implications

These results suggest that AI-mediated tutoring can serve as an effective pedagogical support, especially in programming instruction where abstract concepts can be reinforced through immediate, personalised interaction. Educators may benefit from integrating structured AI prompts into their instructional design, not to replace but to complement teacher guidance. The system's adaptability allows it to adjust to the learner's pace, which can help reduce classroom heterogeneity and support differentiated instruction.

To scale the strategy, adaptations must consider the programming level, language complexity and student digital fluency. Prompt templates can be tailored to beginner or advanced topics, and tutorials on interacting with AI can improve accessibility. Teacher training in prompt engineering and AI mediation is also critical to ensure pedagogical coherence.

5.2 Limitations

This study has several limitations that should be acknowledged to interpret results cautiously:

- Sample size and scope: The study involved only 40 students from a single institution. While differences were statistically significant, generalisation beyond this context must be done carefully (Creswell & Creswell, 2017).

- Design constraints: Although a quasi-experimental design was employed, students were assigned to groups based on existing classes, not random selection. This could introduce selection bias despite demographic and academic equivalence.

- Instrument limitations: The knowledge test and perception survey were designed by the authors and validated through expert judgement (Fraenkel et al., 2012; Lawshe, 1975). Although deemed appropriate, broader validation and reliability analyses could strengthen the instruments' psychometric properties.

- Novelty effect: It is possible that improved engagement and performance were influenced by the novelty of the AI tool rather than its instructional merit. Longitudinal studies are needed to assess sustained impact.

- Ethical concerns and access inequality: Not all students may have equal access to reliable AI tools, or the digital literacy required to use them effectively (UNESCO, 2023). Overreliance on AI could discourage critical thinking if not guided properly.

5.3 Risks and Mitigation Strategies

The integration of AI in education must be accompanied by strategies to mitigate potential overdependence. Teachers should design prompts that foster reflection and promote metacognition (Pintrich, 2002). Hybrid approaches where AI supports—but does not dominate—learning are preferable. Students should also be encouraged to verify AI responses, engage in peer discussion, and develop code independently.

Furthermore, institutions must ensure equitable access to AI tools and digital skills training to avoid exacerbating educational inequalities. Implementation at scale requires not only technological infrastructure, but also pedagogical oversight and institutional support.

6. Conclusion

The results of this study demonstrate that a generative AI-mediated instructional strategy can enhance the learning of object-oriented programming, particularly in improving students' ability to apply concepts in practical contexts. The experimental group, which followed a structured 11-step prompt sequence using ChatGPT, achieved statistically higher scores in the practical component of the assessment and exhibited greater consistency in performance compared to the control group. These findings suggest that the adaptive and personalised interaction with the AI contributed to deeper procedural understanding and reduced variability in learning outcomes.

In addition to improved performance, students expressed highly positive perceptions of the strategy. Survey responses highlighted the tool's usability, clarity of explanations, and practical value for learning, while also emphasising the importance of maintaining teacher presence as a source of guidance and support. This indicates that the effectiveness of the AI-tutoring model is strengthened when combined with pedagogical oversight.

The study therefore supports the integration of structured, AI-driven tutoring systems into programming education—not as replacements for instructors, but as complementary tools that promote engagement, autonomy and skill development. From an instructional perspective, the strategy offers practical implications: it enables differentiated pacing through adaptive prompts, facilitates real-time feedback and metacognitive reflection, and provides a scalable model that can be adapted to different topics and levels within computer science curricula.

7. References

Albayati, A. (2024). Exploring students' acceptance of ChatGPT as a learning tool. *Computers & Education*, 205, Article 104870. https://doi.org/10.1016/j.compedu.2023.104870

Alonso, J., & Carrió, M. (2023). IA en la educación universitaria: desafíos éticos y didácticos [AI in university education: Ethical and didactic challenges]. *Revista de Educación a Distancia [Distance Education Journal*], 23(72), 1–19.

Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman.

Ayala, C., & Aguilar, J. (2023). Chatbots for programming education: Opportunities and challenges. *Education and Information Technologies*, 28, 2493–2508. https://doi.org/10.1007/s10639-023-11610-4

Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. In J. A. Larusson & B. White (Eds.), *Learning Analytics* (pp. 61–75). Springer. https://doi.org/10.1007/978-1-4614-3305-7_4

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54. https://doi.org/10.1145/1929887.1929905

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Creswell, J. W., & Creswell, J. D. (2017*). Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). Sage Publications.

du Plooy, E., Casteleijn, D., & Franzsen, D. (2024). Personalized adaptive learning in higher education: A scoping review of key characteristics and impact on academic performance and engagement. *Heliyon*, 10(21), e39630. https://doi.org/10.1016/j.heliyon.2024.e39630

Elkhodr, M., et al. (2023). Investigating the impact of ChatGPT in ICT education: A case study. *Journal of Computer Assisted Learning*, 39(5), 1241–1257. https://doi.org/10.1111/jcal.12786

Essel, H. B., et al. (2024). Can AI improve higher-order thinking? Evidence from ChatGPT-supported classes. *Educational Technology Research and Development*, 72(1), 85–104. https://doi.org/10.1007/s11423-023-10282-0

Field, A. (2013). *Discovering statistics using IBM SPSS statistics* (4th ed.). Sage Publications.

Feuerriegel, S., et al. (2023). Large language models in education: Opportunities and pitfalls. *Nature Human Behaviour*, 7(3), 347–360. https://doi.org/10.1038/s41562-023-01578-8

Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2012). *How to design and evaluate research in education* (8th ed.). McGraw-Hill.

Gall, M. D., Gall, J. P., & Borg, W. R. (2007). *Educational research: An introduction* (8th ed.). Pearson Education.

Groothuijsen, S., van den Beemt, A. A. J., Remmers, J. J. C., & van Meeuwen, L. W. (2024). AI chatbots in programming education: Students' use in a scientific computing course and consequences for learning. *Computers & Education: Artificial Intelligence,* 7, 100290. https://doi.org/10.1016/j.caeai.2024.100290

Grant, M. M., & Davis, K. A. (1997). Selection and use of content reading strategies: Considerations for professional development. *The Reading Teacher, 50*(6), 512–515.

Husain, M. (2024). Teachers' perspectives on generative AI in programming classes: Benefits, risks, and pedagogical guidance. *Computers & Education Open, 5*, 100131. https://doi.org/10.1016/j.caeo.2023.100131

IEEE. (2017). *IEEE global public policy: Artificial intelligence research, development and regulation*. IEEE.

INEE. (2021). *Marco para la evaluación de los aprendizajes en educación media superior* [Framework for assessment of learning in upper secondary education]. Instituto Nacional para la Evaluación de la Educación.

ISO. (2022). *ISO/IEC 22989 Artificial intelligence concepts and terminology*.

Kapakos, W. A., & Fulk, H. K. (2024). GitHub Copilot: Introducción a la herramienta de inteligencia artificial en un curso de sistemas de información [GitHub Copilot: Introduction of the AI tool in an information systems course]. *Problemas de Sistemas de Información, 25*(4), 106–117. https://doi.org/10.48009/4_iis_2024_108

Kasneci, E., et al. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences, 103*, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Kestin, G., Miller, K., Klales, A., Milbourne, T., & Ponti, G. (2024). AI tutoring outperforms active learning. *Research Square*. https://doi.org/10.21203/rs.3.rs-4243877.v1

Lauren, P., & Watta, P. (2023). Work-in-progress: Integrating generative AI with evidence-based learning strategies in computer science and engineering education. In *2023 IEEE Frontiers in Education Conference (FIE)* (pp. 1–5). IEEE. https://doi.org/10.1109/FIE58773.2023.10342970

Lawshe, C. H. (1975). A quantitative approach to content validity. *Personnel Psychology, 28*(4), 563–575. https://doi.org/10.1111/j.1744-6570.1975.tb01393.x

Liao, Q., et al. (2024). Scaffolding computational thinking with AI: A design framework. *ACM SIGCSE Bulletin, 56*(1), 390–396. https://doi.org/10.1145/3626252.3627550

Llerena-Izquierdo, M., et al. (2023). A didactic AI-driven approach to teaching programming through guided steps. *IEEE Transactions on Education, 66*(2), 215–223. https://doi.org/10.1109/TE.2023.3243748

Menon, P. (2023). *Explorando la asistencia de GitHub Copilot para trabajar con clases en un curso de programación* [Exploring GitHub Copilot assistance for working with classes in a programming course]. *Problemas de Sistemas de Información [Problems of Information Systems],, 24*(4), 66–81. https://doi.org/10.48009/4_iis_2023_106

Monib, W. K., Qazi, A., Apong, R. A., Azizan, M. T., De Silva, L., & Yassin, H. (2024). Generative AI and future education: A review, theoretical validation, and authors' perspective on challenges and solutions. *PeerJ Computer Science, 10*, e2105. https://doi.org/10.7717/peerj-cs.2105

Msambwa, M. M., Kangwa, D., & Lianyu, C. (2024). Integration of information and communication technology in secondary education for better learning: A systematic literature review. *Social Sciences & Humanities Open, 10*(1), 101203. https://doi.org/10.1016/j.ssaho.2024.101203

Pintrich, P. R. (2002). The role of metacognitive knowledge in learning, teaching, and assessing. *Theory into Practice, 41*(4), 219–225. https://doi.org/10.1207/s15430421tip4104_3

Rahman, M. M., & Watanobe, Y. (2023). Review of ChatGPT in education: Potentials and limitations. *Journal of Educational Computing Research, 61*(6), 1224–1246. https://doi.org/10.1177/07356331231174866

Reiser, B. J. (2004). Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *Journal of the Learning Sciences, 13*(3), 273–304. https://doi.org/10.1207/s15327809jls1303_2

Salvatierra, F., & Fernández Laya, N. (2024). *Construir el futuro: La IA en las políticas educativas* [Building the future: AI in educational policies]. IIPE–UNESCO

Sullivan, G. M., & Feinn, R. (2012). Using effect size—or why the p-value is not enough. *Journal of Graduate Medical Education, 4*(3), 279–282. https://doi.org/10.4300/JGME-D-12-00156.1

Sun, D., Boudouaia, A., Zhu, C., & Li, Y. (2024). Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study. *International Journal of Educational Technology in Higher Education, 21*, Article 14. https://doi.org/10.1186/s41239-024-00446-5

Sun, Y., et al. (2024). Effects of ChatGPT on students' programming behavior: A classroom study. *Computers & Education, 208*, 104902. https://doi.org/10.1016/j.compedu.2023.104902

UNESCO. (2022). *Guidance for generative AI in education and research.* United Nations Educational, Scientific and Cultural Organization.

UNESCO. (2023). *Guidelines for the ethics of artificial intelligence in education.* United Nations Educational, Scientific and Cultural Organization.

Yılmaz, R. M., & Karaoğlan, F. G. (2023). The effect of generative AI tools on programming self-efficacy and motivation. *Computers in Human Behavior Reports, 9*, 100112. https://doi.org/10.1016/j.chbr.2023.100112