A friend or a foe? Evaluating ChatGPT's Impact on Students' Computational Thinking Skills

May Mahmoud
New York University Abu Dhabi
Abu Dhabi, United Arab Emirates
m.mahmoud@nyu.edu

Eric Asare
New York University Abu Dhabi
Abu Dhabi, United Arab Emirates
eric.asare@nyu.edu

Nourhan Sakr
The American University in Cairo
Cairo, Egypt
n.sakr@columbia.edu

Sarah Nadi
New York University Abu Dhab
Abu Dhabi, United Arab Emirates
sarah.nadi@nyu.edu

Abstract

With the easy accessibility of generative AI tools, software engineering students are increasingly using these tools in their coursework. However, the impact of these tools on the students' development of core computational thinking skills remains unclear. In this paper, we present our ongoing research that investigates whether access to ChatGPT during the learning process influences students' understanding and retention of essential computational skills. We specifically look at their ability to decompose complex problems (decomposition) and abstract key elements into classes and objects (abstraction). In a controlled, between-subject experiment, we divide students into two groups: one with ChatGPT assistance during the learning phase and a control group relying solely on traditional resources. Participants first undergo a study session to learn and apply decomposition and abstraction skills through various tasks, followed by a retention test session within a week from their study session. We measure the time taken to complete the tasks as well as the correctness of the solutions submitted by participants for each task. Participants also complete a post-study survey to provide insights into their experiences and to gauge their confidence in their solutions and acquired skills. We also analyze screen recordings and ChatGPT interaction logs to better understand the participants' usage strategies and interaction patterns. Our study aims to contribute to the ongoing discussion regarding the pedagogical role of generative AI in Software Engineering education and to offer guidance on effectively integrating these tools into teaching and learning practices.

ACM Reference Format:
May Mahmoud, Eric Asare, Nourhan Sakr, and Sarah Nadi. 2025. A friend or a foe? Evaluating ChatGPT's Impact on Students' Computational Thinking Skills. In *Proceedings of the 2025 Evaluation and Assessment in Software Engineering Companion (EASE Companion '25), June 17–20, 2025, Istanbul, Turkiye.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3727967.3756830

1 Introduction

When considering the skills needed to be an efficient software engineer, we often consider programming languages and tools. However, essential problem-solving skills, such as computational thinking, are just as critical [8]. Computational thinking involves formulating solutions in structured, executable ways, including decomposition of complex problems, pattern recognition, abstraction, and algorithmic thinking—all vital for designing and implementing effective systems [8].

With the rise of generative AI tools like ChatGPT [6], students increasingly use them to complete coursework efficiently. Yet, it remains unclear whether such reliance supports or hinders the learning and retention of computational thinking skills. There is a growing concern that students may develop an over-reliance on generative AI, which could hinder the development of the skills required to become efficient software engineers[4]. While existing studies largely focus on AI's impact on programming, particularly in introductory courses [4, 7], little empirical work explores its effect on computational thinking.

To address this gap, we design a controlled experiment involving Object-Oriented design and Programming (OOP) tasks, a natural context for computational thinking, as it demands abstraction and decomposition in modeling systems. Our study goes beyond task completion and correctness to examine how generative AI affects the retention of these core skills.

Our main research question is *does having access to ChatGPT when solving object-oriented design tasks help or hinder students' ability to learn abstraction and decomposition skills?*. We break down this main research question into the following sub-questions:

- What is the difference—if any—between the performance of the ChatGPT-assisted group and the control group when working on different design tasks that require decomposition and abstraction skills?
- How well do students retain their knowledge gained from learning abstraction and decomposition when they have access to ChatGPT as opposed to no assistance when manually working on similar tasks on a later date (one week)?

This paper presents our ongoing efforts in conducting this study. We present our pilot as well as some preliminary results with four participants. From our preliminary results, we observe that students in the experimental group with access to ChatGPT were able to complete the tasks faster than the control group, with better performance in the learning phase of the study. However, they do have a slightly worse performance in the testing phase. We also observed that the students in the experimental group were more confident in their solution and perceived the task to be less complex than the control group. However, they do report less confidence in their skills when applying them to future tasks.

2 Related Work

Hanifi et al. [3] surveyed 113 computer science and software engineering students on their use and perceptions of ChatGPT. Most students reported using it in projects and found it helpful, though only about half understood how it works. Notably, around 90% of the students reported experiencing hallucinations - situations where ChatGPT generated incorrect or nonsensical information. Similarly, Hammer et al. [2] found that students frequently use ChatGPT in academic work, view it as a helpful "multi-task solver," and often assume it is highly accurate [2].

With this growing reliance of students on generative AI tools like ChatGPT, researchers began to study their effect on students' performance and learning. Kazemitabaar et al. [4] found that code generation tools improve novice programmers' task performance and speed without hindering manual coding abilities. Xue et al. [9] studied students using ChatGPT for a UML task and observed no significant effect on learning performance, though students often stopped using traditional resources once they started interacting with ChatGPT. However, this study focuses primarily on how students interact with ChatGPT and the performance on their assigned tasks, rather than explicitly examining the learning and retention of computational thinking skills such as abstraction and decomposition.

While the above studies suggest that ChatGPT has become "a student's best friend", there are other studies that also reveal potential concerns. Rahe and Maalej [7] found that students often fall into a loop of resubmitting flawed AI-generated code and asking for fixes, rather than using ChatGPT to understand the code or their mistake. This may suggest that students over-rely on the tool without critically thinking about the task at hand.

In their literature survey on the effects of ChatGPT in teaching and learning software engineering, Zönnchen et al. [10] report a common observation that generative AI tools are highly effective for solving simpler programming tasks but less efficient for complex ones. They observed that such tools tend to produce more semantic errors (logical errors leading to incorrect results) than syntax errors (language-related mistakes causing compilation issues). They highlight that critical evaluation of generated code may become more important than writing syntactically correct code [10].

Both these studies further motivate our goal of investigating the effect of using generative AI on students' ability to develop *and* retain more complex skills like computational thinking. In this study, we specifically focus on decomposition and abstraction as two key components of computational thinking.

3 Experiment Overview

3.1 Experiment Purpose

The purpose of our experiment is to investigate the effect of using generative AI tools such as ChatGPT when students learn to decompose and abstract a given natural language problem statement into classes, as opposed to using traditional online resources. In addition to correctness, we consider solution confidence and perceived complexity of the task. We also investigate how well students retrain the knowledge when later working on a task with no assistance.

3.2 Recruitment and Participants

3.2.1 Inclusion and Exclusion Criteria. Our target participants are students in our university. We consider students with sufficient programming skills and limited knowledge of advanced topics like software design and engineering in our inclusion criteria. We define our inclusion criteria to be limited to students who have taken the introductory programming course (either from the Computer Science or Computer Engineering programs) and have not yet taken more advanced courses that go into depth about topics like object-oriented programming and software engineering.

3.2.2 Recruitment. We advertised the study through campus flyers and classroom announcements in introductory courses. Interested students completed an online screening survey covering their programming background, course performance, programming language proficiency, and use of generative AI. We verified eligibility through the survey and contacted selected participants to schedule sessions. Participants received a 100 AED Amazon gift card as compensation, approved by the ethics board.

3.2.3 Experimental Block Assignment. We use a *matched-pair design* [1, 5] to ensure that the students in the two groups are similar in terms of their programming experience and background. We balance students based on their years of programming, Python proficiency, and grade in the Introduction to Programming course. As of now, we have recruited six students. We assign two students to the study's pilot to test both treatments (one experimental group and one control group). We match the other four students and randomly assign them to each group (two in the control group and two in the experimental group). However, only three of the four students showed up for the retention test session.

3.2.4 Ethical Considerations. We applied and obtained our university's ethics board's approval. The study is considered to pose minimal risk to participants. We obtain informed consent from each participant at the beginning of the study. We inform the students of the purpose of the study, and that they can withdraw at any time and take a break at any time during the session. We ensure that the data collected is anonymized and that the screen recording will not include any audio or video of the participants. We store the data securely and with sole access to the research team. We also used the pilot study to confirm that the session was not too long and the tasks were not too complex to avoid fatigue.

3.3 Experiment Setup

We design our experiment as a between-subject controlled experiment where each participant receives only one treatment. We divide the students into two groups. One group has access to AI-assistance functionality (*experimental group*) while the other does not (*control group*). We illustrate the workflow of our experiment in Figure 1. The experiment consists of a study session and a post-study retention test session. Each scheduled session has only one participant.

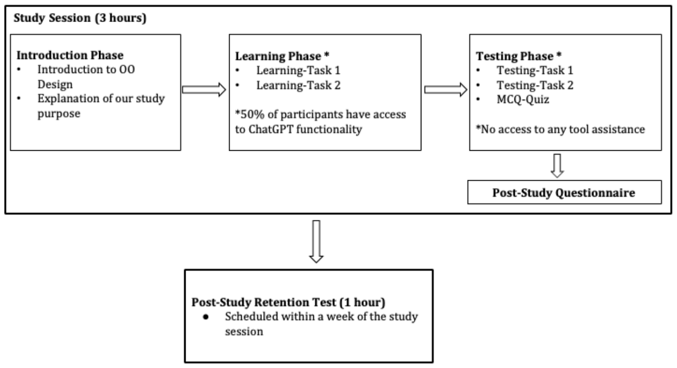
The study session lasts for three hours. At the beginning of the study session (*introduction phase*), the researcher gives the participant a brief lecture on decomposition and thinking in abstraction, including an example, and then the participant watches one educational video on abstraction and another video on writing classes in Python. The participant then moves on to the *learning phase*, where they solve two learning tasks. We share the tasks with the participants as a Google Colab[1] notebook, since it facilitates online sharing and storage. For participants in the experiment group, we have ChatGPT open in a split screen. We use our own ChatGPT account and clear the account data before each study session. At the end of the learning phase, we export and store the chat logs from the ChatGPT session. Once done, participants move on to the *testing phase* where they attempt two testing tasks on their own (regardless of which treatment they receive) and answer a Multiple-Choice Question Quiz (MCQ-Quiz) without any assistance. At the end of the study session, participants fill out a post-session questionnaire where we collect their confidence in their solution as well as information about their experience solving the tasks.

Within a week, participants sit through a *post-study retention test*, where they work on a task similar to the one they solved in the study session but without access to any assisting tool. We chose the week time frame for the retention test instead of longer periods for feasibility of implementation and similar to previous retention studies [4]. Participants can sign up for the retention test anytime during the week following their initial study session. Sign-ups open on the Monday immediately after their session week, ensuring at least a two-day gap between the original session and the retention test (e.g., participants attending a session on Friday will have the weekend as a minimum gap before the test). We provide more details on the tasks below.

3.3.1 Independent and Dependent Variables. Our *independent variable* in the experiment is access to ChatGPT during the learning phase of the study. Our *dependent variables* are the student's solution correctness based on an answer key and a predefined rubric (scored out of a maximum of 116 points), time to complete the tasks, their confidence in their solution, and their skills retention (graded retention test based on an answer key and a pre-defined rubric with a maximum of 60 points). To avoid researcher bias affecting the grades, we had two researchers grade the tasks and discuss any agreement before assigning final grade.

3.3.2 Hypothesis. Our H_o (null hypothesis) states that there is no significant difference in task completion time or scores between the experimental and control groups. The H_A (alternative hypothesis) is that the AI-assisted group will differ significantly in these measures. However, as this is ongoing research, we do not present hypothesis testing in this paper.

[1]https://colab.research.google.com/



Study Session (3 hours)
Introduction Phase
• Introduction to OO Design
• Explanation of our study purpose
Learning Phase *
• Learning-Task 1
• Learning-Task 2
*50% of participants have access to ChatGPT functionality
Testing Phase *
• Testing-Task 1
• Testing-Task 2
• MCQ-Quiz
*No access to any tool assistance
Post-Study Questionnaire
Post-Study Retention Test (1 hour)
• Scheduled within a week of the study session

Figure 1: Experiment Workflow

3.3.3 Tasks. We design the tasks to teach and test participants' decomposition and abstraction skills. In the learning phase, the participant first solves a simple task, **Learning-Task1**, with the aim of learning the syntax of creating a class in Python and adding attributes and methods. In this first task, we provide the students with the description of a simple Car class and ask them to implement it in Python as per the description. We allocate a maximum of 15 minutes for the first task. Then, the participant works on **Learning-Task2**, which is the main design task. The participant works on decomposing and abstracting a problem statement of a university student information system into classes with attributes and methods, and implementing the skeleton of each class in Python, including the attributes and methods. We allocate a maximum of 45 minutes for the second task.

In the testing phase, both groups work on **Testing-Task1**, first modifying the Car class from Learning-Task1 to add an extra attribute and a new method. In the final task **Testing-Task2**, participants work on modifying a basic implementation of the university information system from Learning-Task2 to add new requirements. Participants also answer (**MCQs-Quiz**), which focuses on object design. Some of the questions ask about the general concepts of objects and classes, while some questions provide students with a short system description and then ask them to choose from the list of classes or attributes required to best abstract the system.

Finally, participants fill out a post-session questionnaire that helps us understand their confidence in their solution and gain more insights into their experience solving the tasks. For the **Retention-Test**, we ask participants to attempt a similar design task to Learning-Task2, where they design an event management system, but without access to any assistance. We provide all details regarding the tasks, MCQs, and study workflow in our online replication package: https://figshare.com/s/d2ff01679da3c4880a2b.

3.3.4 Data Collection. We collect the following data from each participant: (1) *Time taken* to complete each task (Each task from the study session and post-study retention test session). (2) The *correctness* of the solution based on a pre-defined rubric (study session and post-study retention test). (3) The *answers to MCQs-Quiz.* (4) Answers to the *post-study session questionnaire.* (5) *Screen recording* of each session with no audio or camera capture of the participants. (6) *Chat log history* with ChatGPT if the student was in the experimental group.

Table 1: Comparison of Mean Task Completion Time and Solution Correctness for the Preliminary Results of the Main Study

Task	Completion Time (minutes, %)		Correctness (%)	
	Control Group	Experimental Group	Control Group	Experimental Group
Learning-Task 1	8 (53%)	5 (33%)	95%	100%
Learning-Task 2	45 (100%)	17.5 (39%)	50%	73%
Testing-Task 1	9.50 (63%)	2.5 (17%)	83%	75%
Testing-Task 2	19.5 (65%)	7.5 (25%)	60%	57%

4 Preliminary Results

We ran two study sessions as a pilot for our study to test the study session's duration and the complexity of the tasks. The pilot helped us confirm that the study workflow and data collection process were effective. We used our observations and findings from the pilot study to add more instructions to clarify the tasks' instructions.

Following the pilot, we ran the study with four participants. Two participants were in the control group, and the other two participants were in the experimental group. Only three of the participants completed the retention test. We report on the task completion time, solution correctness, interaction with ChatGPT, and the post-study questionnaire analysis.

Task Completion Time and Solution Correction. We summarize the mean time for each task and the percentage of the allocated time taken to complete each task as well as the average of correctness score in Table 1. While we cannot conduct a proper statistical analysis at this point to compare both groups, the preliminary results show that the experimental group completed the tasks faster than the control group. Table 2 summarizes the time reduction, which we find ranges from 20% to 60%. We calculate the time reduction as the amount of saved time over the maximum allocated time. We find that the experimental group achieved higher scores in the learning tasks but slightly lower scores in the testing tasks, with Table 2 summarizing the correctness change. To summarize, we observe that the experimental group completed the tasks faster than the control group, without having a noticeable difference in the correctness of the solutions in the learning phase, but they had a slightly worse performance in the testing phase tasks.

We report retention scores for the three participants who completed the test: control group scored 60% and 9%, and the experimental group 27%..

4.0.1 Interaction with ChatGPT and Prompting Patterns. We analyze the screen recording of the learning tasks for the experimental group to discover any prompting patterns. We summarize the prompting patterns we observed in Table 3, including how often we observe each pattern across both participants of the experimental group. Overall, we mostly observe prompting patterns that are similar to previous studies [7, 9] with some minor differences. We notice that the most common prompting pattern is the direct solution pattern, where participants ask for a complete solution or a ready-to-use code snippet by directly pasting the problem statement, followed by the modification pattern, where participants ask for modification to a given code or solution.

4.0.2 Post-study Questionnaire Analysis. We find that the experimental group reported more confidence in their solution and found the tasks less challenging than the control group. Interestingly, the experimental group reported less confidence in their skill when applying them in the future than the control group. Figure 2 summarizes these results. For Figures 2c, and 2b, the figures show the aggregated answers for two questions regarding the participants' solutions, one regarding the solution classes, and the other is regarding the solution attributes and methods.

Table 2: Percentage Change in Task Completion Time and Correctness for Experimental Group vs. Control Group.

Task	Time Reduction (%)	Correctness Difference (%)
Learning-Task 1	20%	+5%
Learning-Task 2 (Learning)	61%	+23%
Testing-Task 1 (Testing)	47%	-8%
Testing-Task 2 (Testing)	38%	-3%

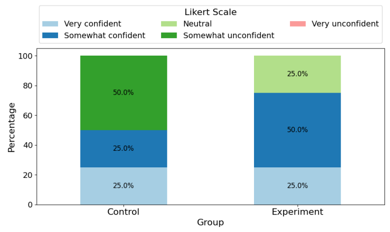
5 Discussion and Next Steps

Although this paper only reports our preliminary results based on a very small sample size, we already find interesting observations. We observe that students in the experimental group with access to ChatGPT were able to complete the tasks faster than the control group, with better performance in the learning phase of the study. However, they do have a slightly worse performance in the testing phase. We are recruiting more participants to increase our sample size and enable more insightful, statistically significant conclusions. It would be interesting to see if this observation still holds in a larger sample size, and across universities.

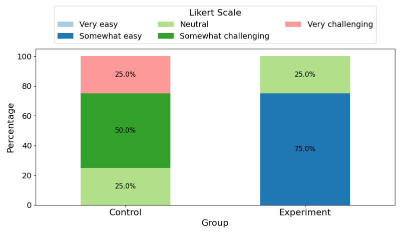
We also observe that the students in the experimental group were more confident in their solution and perceived the tasks to be less complex than the control group. However, they do report less confidence in their skills when applying them in future tasks. With more participants, we can compare the confidence in future skill application to retention test scores.

We plan to do more qualitative analysis of the screen recordings to consider resource consultation order and task context, enabling us to identify patterns in how students engage with ChatGPT or alternative resources during problem-solving. We initially focus on abstraction and decomposition as these two skills are present in object-oriented design; future research can adapt the study to consider other aspects of computational thinking.

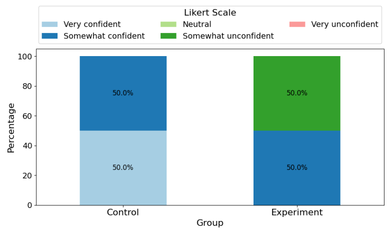
Once completed, our study can inform how to integrate generative AI into computer science and software engineering education. Improved performance and retention using GPT may suggest that generative AI can complement traditional learning, encouraging educators to incorporate these tools in their teaching. On the other hand, if performance or retention is worse, it may indicate that such tools hinder skill development, requiring educators to adapt their



(a) Participants' Confidence in Their Solutions



(b) Perceived Complexity of the Tasks



(c) Participants' Confidence in Future Skill

Figure 2: Survey Results: Participants' Confidence and Perceived Task Complexity

Table 3: ChatGPT prompting pattern categories, including percent of prompts belonging to each category (frequency)

Pattern Category	Description	Frequency
Direct Solution (Solve)	Prompts where users request a complete solution or a ready-to-use code snippet by directly pasting the problem statement.	33%
Modification	Prompts where users ask for modification to a given code or solution.	25%
Code Correction (Fix)	Prompts in which users provide an error message or problematic code and ask ChatGPT to generate a corrected version.	8%
Problem Statement Clarification	Prompts where users ask for clarification of ambiguous or complex parts of the problem statement.	8%
Explanatory (Assumption) Inquiry	Prompts that ask for detailed explanations regarding some aspects of the problem statement.	8%
Miscellaneous (Others)	Any prompts that do not clearly fit into the above categories, such as incomplete queries.	17%

strategies, emphasize critical analysis of AI-generated solutions, and revise assessments to better test retention.

6 Threats to Validity

Construct Validity. We use decomposition and abstraction as proxies for computational thinking. Though not the only components, they are key aspects. While computational thinking is hard to measure objectively in a study setting, we believe object-oriented design tasks exemplify these skills and offer a concrete, feasible way to assess students' understanding and application. Future studies can look at other aspects of computational thinking to augment our study and provide a more comprehensive understanding of the impact of generative AI tools on students' computational thinking skills.

Internal Validity. Variations in the students' prior knowledge can impact the results. We mitigate this confounding factor using a matched-pair design. With this design, along with a higher number of participants, the effect of such variations should be minimized.

External Validity. We conduct the study in a controlled environment, which may not reflect real-world scenarios where students have broader resource access. The tasks are limited in scope and may not capture full system complexity, but such constraints ensure feasibility without overburdening students during the term.

The early results reported in this paper are based only on four participants, only three of which have completed the retention test. To draw more generalizable and statistically significant conclusions, we are currently recruiting more participants in our university and planning to replicate the study in another university. We also encourage others to reach out to us to collaboratively replicate the study in more universities.

7 Conclusion

This paper presents our ongoing research investigating whether access to ChatGPT during the learning process influences students' understanding and retention of computational skills. We designed a controlled experiment that focuses on decomposition and abstraction. We presented preliminary results from four participants, divided evenly across the experimental and control groups. Our early results suggest that students with access to ChatGPT perform learning and testing tasks more quickly. However, while they score higher on the correctness of the learning tasks, they score slightly lower on testing tasks. Our full experiment results will confirm whether these observations continue to hold. Our findings will shape how educators can effectively integrate generative AI in teaching and learning computational thinking skills within software engineering.

Acknowledgments

The authors acknowledge using ChatGPT for editorial purposes only and to occasionally summarize original text to fit space limitations.

References

[1] Miriam Bruhn and David McKenzie. 2009. In pursuit of balance: Randomization in practice in development field experiments. *American economic journal: applied economics* 1, 4 (2009), 200–232.
[2] Sabine Hammer, Sarah Ottinger, Benedikt Zönnchen, Michel Hohendanner, Martin Hobelsberger, and Veronika Thurner. 2024. ChatGPT in Higher Education: Perceptions of Computer Science-Related Students. In *2024 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 01–08.
[3] Khadija Hanifi, Orcun Cetin, and Cemal Yilmaz. 2023. On chatgpt: Perspectives from software engineering students. In *2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security (QRS)*. IEEE, 196–205.
[4] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J Ericson, David Weintrop, and Tovi Grossman. 2023. Studying the effect of AI code generators on supporting novice learners in introductory programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–23.
[5] I Scott MacKenzie. 2024. Human-computer interaction: An empirical research perspective. (2024).
[6] OpenAI. [n. d.]. ChatGPT. https://chat.openai.com. Accessed: 2025-04-16.
[7] Christian Rahe and Walid Maalej. 2025. How Do Programming Students Use Generative AI? *arXiv preprint arXiv:2501.10091* (2025).

[8] Jeannette M Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.
[9] Yuankai Xue, Hanlin Chen, Gina R Bai, Robert Tairas, and Yu Huang. 2024. Does chatgpt help with introductory programming? an experiment of students using chatgpt in cs1. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training*. 331–341.
[10] Benedikt Zönnchen, Veronika Thurner, and Axel Böttcher. 2024. On the Impact of ChatGPT on Teaching and Studying Software Engineering. In *2024 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 1–10.