Using ChatGPT During Implementation of Programs in Education

Norbert Baláž
Department of Computers and Informatics, Technical University of Košice, Slovakia

Jaroslav Porubän[1]
Department of Computers and Informatics, Technical University of Košice, Slovakia

Marek Horváth
Department of Computers and Informatics, Technical University of Košice, Slovakia

Tomáš Kormaník
Department of Computers and Informatics, Technical University of Košice, Slovakia

Abstract

This paper examines the impact of ChatGPT on programming education by conducting an empirical study with computer science students at the Department of Computers and Informatics at the Technical University in Košice. The study involves an experiment where students in a Component Programming course use ChatGPT to solve a programming task involving linked lists, comparing their performance and understanding with a control group that does not use the AI (artificial intelligence) tool. The task necessitated the implementation of a function to add two numbers represented as linked lists in reverse order. Our findings indicate that while ChatGPT significantly enhances the speed of task completion – students using it were nearly three times quicker on average – it may also detract from deep understanding and critical thinking, as evidenced by the uniformity and superficial engagement in solutions among the ChatGPT group. On the other hand, the group working independently displayed a broader variety of solutions and deeper interaction with the problem, despite slower completion times and occasional inaccuracies. The results highlight a dual-edged impact of AI tools in education: while they enhance efficiency, they may undermine the development of critical thinking and problem-solving skills. We discuss the implications of these findings for educational practices, emphasizing the need for a balanced approach that integrates AI tools without compromising the depth of learning and understanding in students.

2012 ACM Subject Classification Software and its engineering → Software creation and management

Keywords and phrases generative artificial intelligence, chatbot, ChatGPT, prompt engineering, source code generation

Digital Object Identifier 10.4230/OASIcs.ICPEC.2024.18

Funding This work was supported by project VEGA No. 1/0630/22 Lowering Programmers Cognitive Load Using Context-Dependent Dialogs.

1 Introduction

In recent years, AI (artificial intelligence) has begun to develop at a tremendous speed, while its applications have become an integral part of our daily lives. Among its newest and most exciting areas is its use for code generation, where machine learning models such as OpenAI's ChatGPT show significant potential for automating and streamlining software development. This technology impacts both students [1] and educators [2] in the realm of software-related education.

[1] Corresponding author

Since its creation, ChatGPT has been used by students around the world to help with a variety of tasks, from writing school papers to implementing term projects [16]. Multiple research papers are exploring ChatGPT, its benefits, and its drawbacks when used in education. The paper [9] highlighted ChatGPT's varied performance across different subject domains and its potential benefits when serving as an assistant for instructors and as a virtual tutor for students. The paper [6] provides an examination of issues and explores the potential use of ChatGPT in educational contexts. The paper [5] explores the potential and problems associated with applying advanced AI models in education. A systematic review of the literature and an analysis of the impact of the application of the ChatGPT tool in education are presented in [10]. Authors in the [11] assess the efficacy of employing the ChatGPT language model to generate solutions for coding exercises within an undergraduate Java programming course. In this paper, we will explore the impact of ChatGPT on education, specifically focusing on its role in learning programming.

A study detailed in [16] involved 41 university students (33 males and 8 females) aged 19–25, examining the impact of ChatGPT on programming education. Throughout an eight-week period, students utilized ChatGPT for their weekly object-oriented programming projects. The study concluded with a questionnaire where students shared their insights, providing an evaluation of ChatGPT's advantages and disadvantages based on their experiences.

The study highlights the benefits of using ChatGPT in programming education, including its ability to provide quick and largely accurate responses and boost confidence in coding. However, it also points out some significant drawbacks, such as promoting laziness, producing occasional inaccuracies, and raising concerns about future job security. The findings suggest that while ChatGPT can be an effective tool in programming education, its utilization needs careful moderation to address both its positive aspects and potential pitfalls.

The reliability of ChatGPT's responses remains a contentious issue, especially in educational settings. The impact of ChatGPT on programming education continues to be an area of active research, focusing on its practical uses, benefits, drawbacks, and ethical considerations [1]. Initially, some educational bodies responded to the rise of AI tools like ChatGPT by banning them, as occurred in early 2023 [14]. As attitudes changed within months, bans were lifted and AI resources were recommended for educational use.

For the correct implementation of ChatGPT in education, it is important to choose the right ways of using it. AI can support creative thinking when solving problems, but it is also simple to use it just to generate the necessary code for a given task, often without any deeper understanding of the problem.

In our study conducted at the Department of Computers and Informatics at the Technical University in Košice, we designed an experiment involving computer science students. The experiment required students enrolled in the Component Programming course to develop and code a solution to a specified task using Java, followed by completing a questionnaire about their understanding of the task, their evaluation, and their opinions on using ChatGPT. We divided the students into two groups, allowing one to use ChatGPT and prohibiting the other.

We hypothesize that while ChatGPT might improve task completion speed and productivity, it may detract from students' understanding of the task, critical thinking, and problem-solving creativity. The questionnaire aimed to gather students' perspectives on the use of ChatGPT and how they personally utilized the tool. We anticipated that the group using ChatGPT, despite potentially completing tasks correctly, might exhibit a poorer grasp of the task. Conversely, the group without ChatGPT might demonstrate a deeper understanding of the problem or proposed solutions, even if their implementations were not fully functional.

2 Experiment

In this section we present the experiment of using ChatGPT by students in a Component Programming course to solve the programming task. The task concerns the addition of two numbers, which are represented as a linked list, where each node represents one digit of the whole number. This task complies with our department's standards [4].

Text of the task: We have entered 2 integers represented as the linked list format so that each node represents a digit in the number. Nodes form a number in reverse order, e.g. `1 -> 2 -> 3 -> 4 -> 5` is the number `54321`. Complete the provided code so that it returns the sum of the two provided numbers in this format. For example: `9 -> 9` and `5 -> 2` returns `124` (99 + 25) represented as `4 -> 2 -> 1`. Provided code:

```
class ListNode {
    int val;
    ListNode next;

    ListNode(int val) {
        this.val = val;
    }
}

class AddLinkedLists {

    public ListNode addTwoNumbers(ListNode l1, ListNode l2) {
        // TODO: implement this function for adding 2 numbers
    }

    public static void main(String[] args) {
        ListNode l1 = new ListNode(0);
        l1.next = new ListNode(1);

        ListNode l2 = new ListNode(3);
        l2.next = new ListNode(2);

        AddLinkedLists solution = new AddLinkedLists();
        ListNode result = solution.addTwoNumbers(l1, l2);

        while (result != null) {
            System.out.print(result.val + " ");
            result = result.next;
        }
    }
}
```

In the designated coding exercise, numbers are encapsulated within linked lists with the least significant digit at the head of the list. Students are tasked with implementing the `addTwoNumbers` function, which takes two such linked lists, $l1$ and $l2$, as inputs. This function adds the numbers represented by these lists and returns a new linked list that encapsulates the sum, with digits again in reverse order, starting from the least to the most significant.

The challenge lies in correctly iterating through both lists, summing corresponding digits, and managing any carry-over that occurs when digits sum to 10 or more. This process begins with the head of each list, ensuring that the addition mirrors the operation of adding numbers from their least significant digits upwards.

This task, while straightforward for ChatGPT with its ability to quickly generate the correct solution, presents a significant challenge for introductory programming course students without AI assistance. These students must not only develop the solution independently, but they must also deeply understand the underlying problem and manage the complexities of linked list manipulation and digit-wise addition. The experiment familiarized the students

not only with the task but also with the problem, laying the groundwork for a comparative analysis of problem-solving approaches and the depth of understanding between the two groups.

A total of 23 students worked on the assignment. There were 14 students in the first group who worked with ChatGPT, and 9 in the second group who worked without ChatGPT support. Students were in different study groups, which caused a significant disproportion in group size due to absences. Students solved the task and filled out the questionnaire during the exercise from the Component programming course. Their solutions can be further used in analysis or comparison with previously created relevant datasets [15].

2.1 ChatGPT Group

In the group where students used ChatGPT to solve the task, all participants successfully implemented the correct solution. The times taken to complete the task varied, with an average duration of 8 minutes and 17 seconds. The quickest completion time recorded was 1 minute, and the slowest was 25 minutes, as depicted in the Fig. 1.

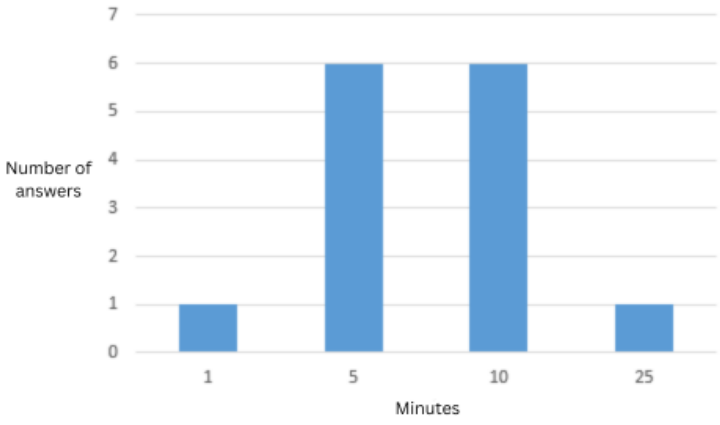

Figure 1 ChatGPT group task solving time.

Analysis of the interactions between students and ChatGPT revealed that the average number of prompts (i.e., questions or commands given to ChatGPT) per student was 3.7, with the range being from 1 to 9 prompts. The transcripts of these interactions revealed highly uniform solutions, primarily generated by ChatGPT with minimal modification by the students.

This uniformity in the solutions and reliance on ChatGPT's outputs could explain why students perceived the task as relatively easy, rating its difficulty as 4 out of 10 on average. This rating is visualized in Fig. 2, which illustrates the students' perceived challenge of the task. This perception highlights the impact of AI tools like ChatGPT in simplifying complex tasks, but also raises concerns about the depth of understanding and engagement in problem-solving when using such tools.

2.1.1 Understanding the problem

The questionnaire, which included specific questions like "Why are numbers in a linked list represented in reverse?", assessed students' understanding of problem-solving. The first group had access to ChatGPT for assistance. Responses varied, with some students clearly recognizing benefits like simplified manipulation and more efficient operations, while many answers were vague or off-topic, indicating a lack of engagement with the material.

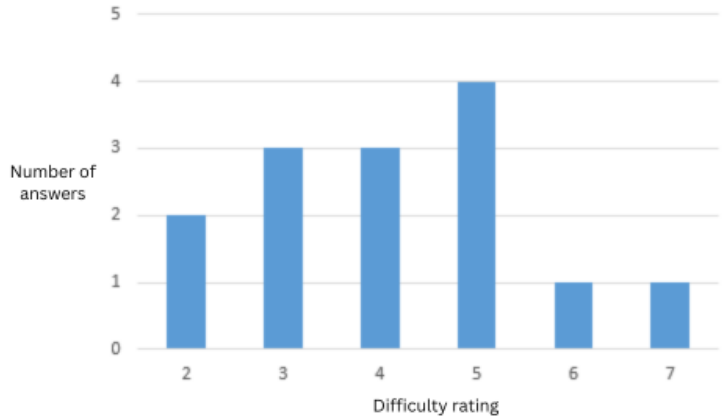

Figure 2 ChatGPT group difficulty rating.

Further questions asked students to explain the logic behind their solutions, even if incorrect. Many demonstrated an understanding of adding numbers via linked lists, with responses varying from detailed technical explanations involving carry transfers and pointer movements to concise summaries. However, some responses were unclear or irrelevant, indicating confusion or disinterest.

The final question probed whether students gained new insights or improved their mastery of the data structure. Responses were split, with half reporting new knowledge or enhanced understanding and the other half noting no significant learning. This variation underscores the differing levels of comprehension and engagement with the task.

2.2 Independent Group

In the independent group, 9 students attempted to implement the task without ChatGPT assistance, and 6 of them managed to solve it correctly. However, a closer look at the code revealed some discrepancies. Although one student implemented the solution correctly, they failed to convert the result back into a list format. Another student's code only outputs the tenth digit of the result. This latter error, though minor and quickly rectifiable, seemed to stem from a lack of careful code testing before submission, especially since this student took the longest to complete the task. Despite these issues, since the primary focus of this experiment was not on code perfection but on solving the problem, these solutions were deemed correct.

The average time taken by this group to complete the task was 22 minutes and 26 seconds, as illustrated in Fig. 3. Students who solved the problem correctly had an average completion time of 23 minutes and 40 seconds, with the quickest solution taking 15 minutes and the slowest taking 45 minutes. Those who did not solve the task spent an average of 20 minutes, with times ranging from 10 to 30 minutes.

Overall, this group rated the task's difficulty as 5 out of 10, with individual evaluations depicted in the Fig. 4. This suggests a moderate level of challenge perceived by the students who tackled the task independently.

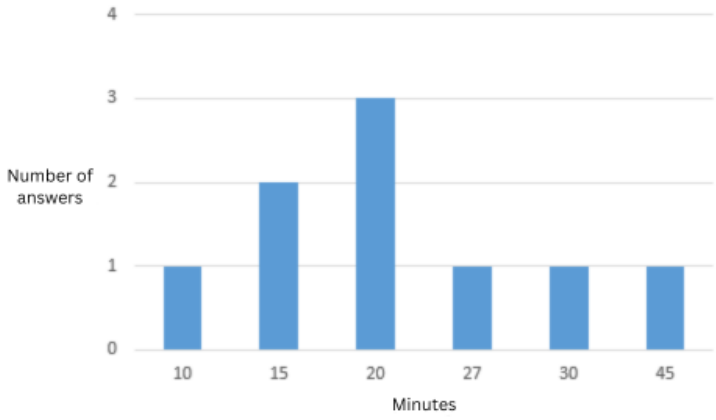

Figure 3 Independent group task solving time.

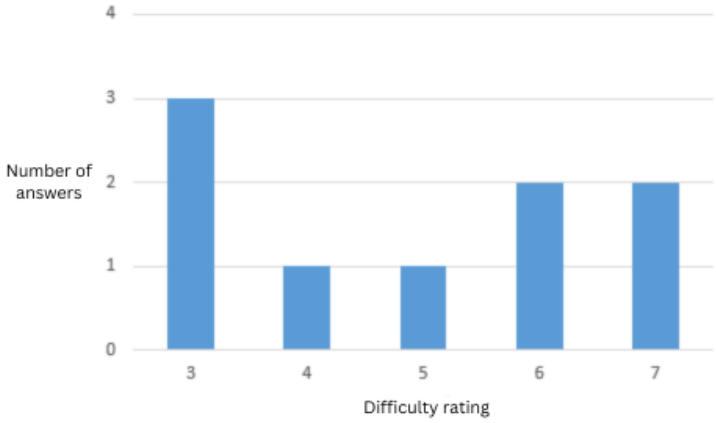

Figure 4 Independent group difficulty rating.

2.2.1 Correct solutions

The analysis of correct solutions using verified methods [12] has shown several diverse approaches. Some students converted linked lists into integers, summed them, and converted the sum back into a linked list. This method demonstrates a straightforward application of mathematical operations and transitions between number formats, but it poses a risk of precision issues or memory overflow with large numbers. Meanwhile, a different approach involved the use of *StringBuilder* to reverse and manipulate the numbers, showcasing creativity and originality, even if this method might be less efficient in some scenarios.

The most accurate solution to the task involved simultaneously traversing both lists, summing corresponding nodes, and properly managing carryover from one digit to the next. This approach was particularly effective as it minimized type conversions and directly manipulated the nodes of the linked lists, aligning closely with the task's objectives.

2.2.2 Understanding the problem

Responses to the analysis of why linked lists often store numbers in reverse showed a wide range of understanding levels. Most were concise; some addressed the key point directly, others came close, and a few admitted complete ignorance.

The majority of students demonstrated at least a basic grasp of the rationale behind storing numbers in reverse order, particularly highlighting how this facilitates the addition process. Notably, even students who struggled with the task recognized the purpose of this data structure orientation.

The responses to questions regarding the logic behind their solutions revealed a diversity of approaches and depths of understanding. This variability provides insights into the students' strategic thinking and problem-solving skills, even though not all solutions were optimal or adhered strictly to best practices.

In their explanations, most students described converting the numbers represented by the lists into whole numbers, adding these, and then converting the result back into a linked list. This method aligns with the intuitive logic commonly used in everyday number manipulation and was the predominant approach even among those who did not complete the task successfully. Other students, who briefly described their working solutions, mentioned techniques like using *StringBuilder* for manipulating strings and numbers.

In response to the question about gaining new knowledge or a better understanding of the given data structure, the group's feedback was predominantly negative, with seven indicating no new insights and only two reporting positive learning outcomes. This suggests that most students were already quite familiar with the data structure in question.

3 Conclusion

In this paper, we investigate the effect of using ChatGPT on programming education by conducting a limited-scale empirical study with introductory programming course students. After analyzing both groups of students and their responses to the programming task, we can draw several conclusions about the impact of using ChatGPT versus working independently:

- Speed of Completion: Students who used ChatGPT finished their tasks much faster, nearly three times quicker on average. This efficiency demonstrates the capability of AI to streamline problem-solving processes.
- Depth of Understanding: The speed advantage for the ChatGPT group came at a cost. Most of their solutions were very similar, suggesting heavy reliance on AI without much alteration. Many students accepted the generated solutions without deeply engaging with their content or understanding their functionality, leading to a superficial grasp of the tasks. However, about a third of these students showed they might have the potential to solve the problems independently, indicating some retained problem-solving abilities.
- Diversity of Solutions and Critical Thinking: On the other hand, the independent group, while slower, displayed a wider variety of solutions and tended to describe their methods simply and effectively. This not only shows a broader scope of creativity but also suggests a deeper interaction with the task, which can enrich the learning experience.
- Approach to Problem Solving: Independent students often used approaches similar to traditional paper-and-pencil methods, reflecting an intuitive and straightforward way of thinking. This method suggests that they relied on fundamental problem-solving and mathematical reasoning skills rather than automated processes.

These findings underscore the need for a balanced approach in educational settings that thoughtfully integrates the use of AI tools like ChatGPT with traditional learning techniques. This strategy ensures that students not only achieve quick solutions but also deeply understand the processes and principles involved, thereby cultivating their critical thinking and problem-solving skills. Their solutions should be examined carefully, since overuse of ChatGPT without actual understanding of the created solution can be considered

plagiarism, which can create issues for the evaluator [3] and students of its own. Naturally, source codes created by ChatGPT can be somewhat different in individual instances, but their overall similarity can be easily detected, as is confirmed in previous research conducted on our department [8]. We also observed that multiple results that utilized ChatGPT were not the best possible ones but rather the most common ones, which is also proven by related work [7].

One notable observation is that students who worked without ChatGPT frequently gravitated towards more traditional methods of solving the task. This choice suggests a deeper level of engagement with the problem and a more inventive approach to finding solutions. Even though not all these attempts were successful, the students' ability to devise potential strategies on their own demonstrates their capacity for critical thinking and independent analysis. This contrast with the AI-assisted group highlights how reliance on technology can sometimes bypass the deeper learning processes involved in problem-solving.

When interpreting the results and conclusions of this experiment, it is necessary to consider validation risks that may affect the generality and accuracy of our findings:

- Sample size – With a total of 23 participants, the sample is relatively small, which may limit the statistical significance of the findings and their applicability to a broader student population.
- Group heterogeneity – Dividing students into two groups may introduce hidden differences beyond just access to ChatGPT (such as prior programming experience, motivation, or personal preferences), which could bias the results.
- Limited scope of tasks – The experiment focuses on only one specific programming task, which may limit the ability to apply the findings to other types of tasks or subjects.

Using ChatGPT and similar tools to solve school assignments and projects can significantly increase the speed and productivity of students, but at the same time, it can have a potentially negative effect on their understanding of the subject matter, their ability to think critically, and their creativity. Conducted experiment thus confirmed our assumption. The findings support the results presented by Savelka et al. [13] about students and by Balse et al. [2] regarding mentors.

A suitable approach could be the combined use of ChatGPT as a tool for obtaining quick information or solution proposals while simultaneously ensuring a deeper study of the material and independent problem solving.

References

1 Christos-Nikolaos Anagnostopoulos. Anagnostopoulos CN. ChatGPT impacts in programming education: A recent literature overview that debates ChatGPT responses [version 1; peer review: 1 approved with reservations]. *F1000Research*, 2023. doi:10.12688/f1000research.141958.1.

2 Rishabh Balse, Prajish Prasad, and Jayakrishnan Madathil Warriem. Exploring the potential of GPT-4 in automated mentoring for programming courses. In *ACM Conference on Global Computing Education Vol 2*, page 191, 2023. doi:10.1145/3617650.3624946.

3 Miroslav Binas. Affecting students behavior with plagiarism detection in evaluation process. In *International Conference on Overcoming the Challenges and Barriers in Open Education*, pages 441–441, 2018.

4 Miroslav Binas and Emilia Pietrikova. Useful recommendations for successful implementation of programming courses. In *IEEE International Conference of Emerging Elearning Technologies and Applications*, pages 397–401, 2014. doi:10.1109/ICETA.2014.7107618.

5 Simone Grassini. Shaping the future of education: exploring the potential and consequences of AI and ChatGPT in educational settings. *Education Sciences*, 13(7):692, 2023. doi:10.3390/educsci13070692.

6 Mohanad Halaweh. ChatGPT in education: Strategies for responsible implementation. *CONT ED TECHNOLOGY*, 2023. doi:10.30935/cedtech/13036.

7 Arto Hellas, Juho Leinonen, Sami Sarsa, Charles Koutcheme, Lilja Kujanpää, and Juha Sorva. Exploring the responses of large language models to beginner programmers' help requests. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*, ICER '23, pages 93–105, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3568813.3600139.

8 Marek Horvath and Emilia Pietrikova. An experimental comparison of three code similarity tools on over 1,000 student projects. In *IEEE World Symposium on Applied Machine Intelligence and Informatics*, pages 423–428, 2024. doi:10.1109/SAMI60510.2024.10432863.

9 Chung Kwan Lo. What is the impact of ChatGPT on education? a rapid review of the literature. *Education Sciences*, 13(4):410, 2023. doi:10.3390/educsci13040410.

10 Marta Montenegro-Rueda, José Fernández-Cerero, José María Fernández-Batanero, and Eloy López-Meneses. Impact of the implementation of ChatGPT in education: A systematic review. *Computers*, 12(8):153, 2023. doi:10.3390/computers12080153.

11 Eng Lieh Ouh, Benjamin Kok Siew Gan, Kyong Jin Shim, and Swavek Wlodkowski. ChatGPT, can you generate solutions for my coding exercises? an evaluation on its effectiveness in an undergraduate java programming course. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, pages 54–60, 2023. doi:10.1145/3587102.3588794.

12 Emilia Pietrikova and Sergej Chodarev. Profile-driven source code exploration. In *IEEE Federated Conference on Computer Science and Information Systems*, pages 929–934, 2015. doi:10.15439/2015F238.

13 Jaromir Savelka, Arav Agarwal, Christopher Bogart, Yifan Song, and Majd Sakr. Can generative pre-trained transformers (GPT) pass assessments in higher education programming courses? In *Conference on Innovation and Technology in Computer Science Education V. 1*, pages 117–123. ACM, 2023. doi:10.1145/3587102.3588792.

14 Jody Serrano. New york city schools lift ban on ChatGPT, say initial fear "overlooked the potential" of AI. URL: https://gizmodo.com/new-york-city-public-schools-lift-ban-chatgpt-ai-1850453424.

15 Matus Sulir, Michaela Bacikova, Matej Madeja, Sergej Chodarev, and Jan Juhar. Large-scale dataset of local java software build results. *Data*, 5(3):86:1–86:11, 2020. doi:10.3390/data5030086.

16 Ramazan Yilmaz and Fatma Gizem Karaoglan Yilmaz. Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Computers in Human Behavior: Artificial Humans*, 1(2):100005, 2023. doi:10.1016/j.chbah.2023.100005.