

## Naučni članci

Vrsta rada: Originalni naučni članak

Primljen: 21.05.2023.

Prihvaćen: 02.06.2023.

UDK: 004.891:378.014.3

37.091.312::004

## ChatGPT in programming education: ChatGPT as a programming assistant

Milić Vukojičić<sup>1</sup>, Jovana Krstić<sup>2</sup>

<sup>1</sup> International School, Belgrade, Serbia; [milic.vukojicic@iss.edi.rs](mailto:milic.vukojicic@iss.edi.rs)

<sup>2</sup> "Dragan Kovačević" Elementary School, Belgrade, Serbia; [jovanakrstic20@yahoo.com](mailto:jovanakrstic20@yahoo.com)

**Abstract:** This study explores the use of ChatGPT as a programming assistant in the context of programming education. The aim is to compare student work with and without the assistance of ChatGPT in explaining standard methods of solution, including linear search, bubble sort, totaling, counting, finding maximum and minimum values, and calculating the average. The primary objective is to evaluate the impact of ChatGPT on students' understanding and performance in these fundamental problem-solving techniques. The study will focus on students proficient in Python programming language version 3.10. The research will employ a controlled experimental design, with two groups of students: Group A and Group B. Group A will receive guidance and support from ChatGPT, whereas Group B will work without any external assistance. By comparing the outcomes and performance of both groups, the study aims to assess the effectiveness of ChatGPT as a programming assistant. The evaluation will consider various factors, including the quality of code written by students, the clarity of their explanations, and their overall understanding of the standard methods of solution. Additionally, students' perceptions and experiences regarding the use of ChatGPT as a programming assistant will be collected through surveys or interviews. The results indicate that the use of ChatGPT as a programming assistant yielded better outcomes compared to working without any external assistance. The assistance provided by ChatGPT enhanced students' coding proficiency, improved the quality of their explanations, and deepened their understanding of standard methods of solution.

**Keywords:** ChatGPT, large language models, education, computer science, programming

## 1. Introduction

In today's rapidly evolving digital landscape, programming education has become increasingly vital for individuals seeking to succeed in various professional domains. As technology continues to advance, traditional methods of teaching programming have faced challenges in engaging students and catering to their diverse learning needs. This has led to a growing interest in exploring innovative approaches to enhance programming education. One promising avenue for improving programming education lies in the integration of ChatGPT, a state-of-the-art language model developed by OpenAI. ChatGPT has demonstrated remarkable abilities in generating human-like responses and providing personalized feedback, making it a potential tool for transforming the learning experience in programming education. By harnessing the power of ChatGPT, educators have the opportunity to create interactive and adaptive learning environments that can better engage students and enhance their understanding of programming concepts.

The aim of this paper is to delve into the strategies for implementing ChatGPT in programming education, with a focus on exploring the potential benefits and challenges associated with its integration. By analyzing the current landscape of programming education and the limitations of conventional approaches, this study aims to shed light on the transformative potential of ChatGPT. The research will provide a comprehensive analysis of various implementation strategies, including leveraging ChatGPT as a virtual programming tutor, enabling collaborative coding experiences, and harnessing its real-time problem-solving capabilities.

By embracing ChatGPT in programming education, educators have the opportunity to revolutionize the way programming is taught, fostering personalized learning experiences and equipping students with the skills needed to thrive in the digital era. This exploration of ChatGPT's implementation strategies aims to contribute to the ongoing discourse on innovative approaches to programming education and inspire educators to harness the potential of AI-powered tools for enhanced learning outcomes.

## 2. Related work

Using ChatGPT in programming education can provide valuable support to learners by serving as a virtual programming tutor, offering real-time code explanations, and assisting with problem-solving. It can also facilitate collaborative coding experiences and provide personalized feedback, fostering interactive and adaptive learning environments.

Paper [1] highlights the importance of providing training and awareness to faculty members and students regarding the use of ChatGPT in programming education. Educators should educate students on how to evaluate the accuracy, track queries, and distinguish between text generation and idea generation. Allowing students to use ChatGPT can foster creativity, provide equal opportunities for idea development, and identify students who put in more effort. As ChatGPT continues to evolve, universities may integrate it into learning management systems and OpenAI could develop a specialized version tailored to academia. The authors emphasize the need for critical thinking, presentation skills, and assessment methods like viva and defending one's work to adapt to the changing educational landscape. Practical examples of using ChatGPT for academic writing are provided, suggesting its potential use in academic research and publication. The suggested policy can be customized and implemented to meet the specific needs of institutions and courses.

In this comprehensive review of 50 articles [2], the performance of ChatGPT in different subject domains is examined, revealing its diverse outcomes. The study highlights the potential advantages of ChatGPT as an assistant for instructors and a virtual tutor for students. However, the review also raises concerns regarding the tool's tendency to generate inaccurate or fabricated information, posing a threat to academic integrity.

The findings of this review stress the urgent need for schools and universities to take proactive measures by revising their existing guidelines and policies related to academic integrity and plagiarism prevention. It is crucial for institutions to update their frameworks to address the specific challenges posed by the use of ChatGPT. Furthermore, it is recommended that instructors undergo training to effectively utilize ChatGPT and develop skills to identify instances of student plagiarism that may arise from its use. Equally important, students should receive education regarding the appropriate utilization of ChatGPT, understanding its limitations, and the potential impact it may have on academic integrity. By implementing these recommendations, educational institutions can foster responsible and ethical use of ChatGPT, ensuring that it enhances the learning experience while upholding the values of academic integrity and authenticity.

Several recommendations are proposed to mitigate the risks associated with ChatGPT [3]. Firstly, educational institutions should consider incorporating ChatGPT into their practices, encouraging students, teachers, and researchers to explore its potential benefits while ensuring inclusive, equitable, transparent, and ethical use. Secondly, assessment standards should be revised to prevent unfair evaluation, emphasizing the utilization of analytical, critical, communicative, and problem-solving abilities. Implementing formative assessment and engaging activities that foster debate, group discussions, presentations, and teamwork are advised.

Paper [4] explores ChatGPT and other AI language models (LLMs). ChatGPT and other AI language models (LLMs) have emerged as valuable tools for educational and research purposes. ChatGPT, in particular, stands out as a revolutionary LLM capable of engaging in human-like conversations and generating text that closely resembles human writing. Its applications span a wide range, including answering questions, writing essays, solving problems, explaining complex topics, providing virtual tutoring, language learning, programming education, teaching, and supporting research. Surveys and experimental results indicate that ChatGPT is useful not only in programming education but also in other educational and research contexts.

The study utilized proposed in the paper [5] a three-stage instrumental case study to explore the concerns surrounding the use of chatbots in education, focusing specifically on ChatGPT. The findings highlight the power of ChatGPT as an educational tool but emphasize the need for caution and the establishment of guidelines for its safe use in educational settings. The study identifies several research directions and questions that researchers and practitioners should investigate to ensure better and safer adoption of chatbots, particularly ChatGPT. The implications of this study are both theoretical and practical. Theoretical implications contribute to the ongoing debate on chatbot use in education, providing insights into relationship formation theories and the need for new teaching philosophies to align with chatbot-driven educational reforms. From a practical standpoint, the study emphasizes the importance of upskilling competencies by developing curricula that enable teachers and students to effectively navigate and understand chatbot advancements. Additionally, the study emphasizes the development of responsible chatbots in education by considering human values beyond privacy concerns.

### **3. Proposed methods**

In this study, we propose a method to compare student work with and without the assistance of ChatGPT in explaining standard methods of solution, such as linear search, bubble sort, totaling, counting, finding maximum and minimum values, and calculating the average. The objective is to assess the impact of ChatGPT on students' understanding and performance in these fundamental problem-solving techniques. Students will use Python programming language version 3.10.

The study will employ a controlled experimental design, involving two groups of students: Group A and Group B. Group A will receive assistance from ChatGPT, while Group B will work without any external assistance. The aim is to compare the outcomes and performances of the two groups to evaluate the effectiveness of ChatGPT in supporting students' understanding of the standard methods of solution.

Participants will be selected from a pool of students who have a basic understanding of programming concepts and are familiar with the standard methods of solution. It is important to ensure that both groups have a similar level of prior knowledge and skills to minimize potential bias. Group A and Group B will take the test after the exploration of the standard method of solution.

Group A had access to ChatGPT as an additional resource. They used ChatGPT to seek clarification, get hints or suggestions, and receive guidance in implementing the standard methods of solution. The students interacted with ChatGPT through a text-based interface, posing questions and receiving responses. Group B worked on the programming problems without any external assistance. They will rely solely on their knowledge and understanding of the standard methods of solution to complete the tasks. They did not have access to ChatGPT or any other external resources except the subject book. Students' performance will be assessed based on the correctness and efficiency of their solutions, as well as the clarity and coherence of their explanations for the standard methods of solution. By comparing student work with and without ChatGPT assistance, this study aims to provide insights into the potential benefits and drawbacks of using AI models like ChatGPT to enhance students' understanding of standard methods of solution in programming.

#### 4. Experiments

The aim of this experiment is to compare the performance of students when explaining standard methods of solution, such as linear search, bubble sort, totaling, counting, and finding maximum, minimum, and average values, with and without the assistance of ChatGPT. ChatGPT is an advanced language model that can generate human-like text and provide explanations and solutions to various queries. By comparing student work with and without ChatGPT assistance, we can assess the impact of this AI tool on students' understanding and presentation of these standard methods of solution. The findings of this experiment will contribute to our understanding of the potential benefits and limitations of integrating ChatGPT in educational settings, particularly in programming and algorithmic problem-solving.

<b>Linear Search:</b>	<b>Bubble Sort:</b>	<b>Totaling (Summing) Values:</b>
<pre>def linear_search(arr, target):     # Iterate over each element in the     array     for i in range(len(arr)):         # Check if the current element         matches the target value         if arr[i] == target:             # Return the index of the             matching element             return i         # If target value is not found,         return -1     return -1</pre>	<pre>def bubble_sort(arr):     n = len(arr)     # Traverse the array from the     first element to the second-to-last     element     for i in range(n):         # Compare each pair of         adjacent elements         for j in range(n - i - 1):             # Swap elements if they are             in the wrong order             if arr[j] &gt; arr[j + 1]:                 arr[j], arr[j + 1] = arr[j + 1],                 arr[j]</pre>	<pre>def total_values(arr):     total = 0     # Iterate over each element in the     array     for num in arr:         # Add the current element to         the total         total += num     # Return the total sum     return total</pre>
<b>Counting:</b>	<b>Finding Maximum Value:</b>	<b>Finding Minimum Value:</b>
<pre>def count (arr):     count = 0     # Iterate over each element in the     array</pre>	<pre>def find_maximum(arr):     maximum = arr[0]     # Iterate over each element in the     array</pre>	<pre>def find_minimum(arr):     minimum = arr[0]     # Iterate over each element in the     array</pre>

```

for num in arr:
    # Increment the count if the
    element matches the target value
    count += 1
# Return the count of
occurrences
return count

```

```

for num in arr:
    # Update the maximum value
    if a larger element is found
    if num > maximum:
        maximum = num
# Return the maximum value
return maximum

```

```

for num in arr:
    # Update the minimum value
    if a smaller element is found
    if num < minimum:
        minimum = num
# Return the minimum value
return minimum

```

#### Finding Average Value:

```

def find_average(arr):
    total = 0
    # Iterate over each element in the
    array
    for num in arr:
        # Add the current element to
        the total sum
        total += num
    # Calculate the average by
    dividing the sum by the number of
    elements
    average = total / len(arr)
    # Return the average value
    return average

```

**Figure 1.** Example of standard methods of solution with the assistance of ChatGPT

We notice a clear improvement in the use of comments. Students who utilized ChatGPT support have incorporated detailed comments, elucidating their code's logic and offering clarity to readers. It appears that ChatGPT's assistance has encouraged students to express their thoughts more effectively, resulting in a more comprehensible and coherent code structure. Furthermore, the examples in Figure 1 also demonstrate a greater uniformity in the writing style of student work with ChatGPT. The code snippets exhibit consistent indentation, spacing, and naming conventions, reflecting a more standardized approach to code writing. This suggests that ChatGPT may have provided guidance or suggestions to students regarding coding best practices, resulting in a more uniform and professional coding style.

#### Linear Search:

```

def linear_search(array, x):
    for i in range(len(array)):
        if array[i] == x:
            return x
    return 0

```

#### Bubble Sort:

```

def bubblesort(array):
    length = len(array)
    for i in range(length):
        for j in range(length - 1):
            # swap function
            if array [j] > array [j + 1]:
                array [j], array [j + 1] =
                array [j + 1], array [j]

```

#### Totaling (Summing) Values:

```

def totalling (array):
    sum = 0
    for number in arr:
        sum += number
    return sum

```

Counting:	Finding Maximum Value:	Finding Minimum Value:
<pre>def count (arr):   count = 0   for elm in range(0, len(arr)):     count += 1   return count</pre>	<pre>def max (arr):   max = arr[0]   for element in arr:     if num &gt; max:       element = num   return max</pre>	<pre>def min(array):   min = array [0]   # iterate over each elements   for element in array:     if element &lt; min:       min = element   # return min   return min</pre>
<p><b>Finding Average Value:</b></p> <pre>def average(array):   sum = 0   for element in array:     sum = sum + element   return sum/len(array)</pre>		

**Figure 2.** Example of standard methods of solution without the assistance of ChatGPT

Figure 2 shows the lack of uniformity in writing style among students when they do not have access to ChatGPT. The code snippets demonstrate inconsistent indentation, spacing, and naming conventions. This inconsistency can make it difficult for others to read and interpret the code, hindering collaboration and comprehension. From the examples absence of ChatGPT's assistance has led to a decrease in the use of comments and a less uniform writing style.

The observations made in Figure 2 highlight the potential benefits of incorporating ChatGPT as a supportive tool in the learning process. By leveraging ChatGPT's capabilities, students can improve their commenting practices, providing comprehensive explanations of their code and facilitating understanding for others. Additionally, ChatGPT can guide students towards adopting a more uniform and professional writing style, enhancing code readability and maintainability.

## 5. Conclusion

In conclusion this research provides evidence of the positive influence of ChatGPT on student work. The integration of ChatGPT as a supportive tool has led to an increase in the use of comments, enhancing code understanding and facilitating collaboration among peers and instructors. Moreover, the uniformity in writing style promotes code readability and maintainability, making it easier for others to comprehend and work with the code.

These improvements not only benefit the individual students by honing their coding skills but also contribute to a more efficient and effective learning environment. By encouraging students to adopt good commenting practices and adhere to a uniform writing style, ChatGPT empowers students to produce high-quality code that is easier to understand, modify, and build upon.

## References

1. Halaweh, M. (2023). ChatGPT in education: Strategies for responsible implementation. *Contemporary Educational Technology*, 2023, 15(2), ep421.
2. Lo, C. K. (2023). What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature. *Education Sciences*, 13(4), 410.
3. Sok, S., & Heng, K. (2023). ChatGPT for education and research: A review of benefits and risks. Available at SSRN 4378735.
4. Rahman, M. M., & Watanobe, Y. (2023). ChatGPT for education and research: Opportunities, threats, and strategies. *Applied Sciences*, 13(9), 5783.
5. Tlili, A., Shehata, B., Adarkwah, M. A., Bozkurt, A., Hickey, D. T., Huang, R., & Agyemang, B. (2023). What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education. *Smart Learning Environments*, 10(1), 15.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License.