The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center ASEE
Portland, OR . June 23 - 26, 2024
Paper ID #41661

Board 43: AP-CS, ChatGPT and Me: a High School Student Perspective

Dr. Zoe Wood, California Polytechnic State University, San Luis Obispo

Whether it is creating computer graphics models of underwater shipwrecks or using art and creativity to help students learn computational thinking, Professor Zoe Wood's projects unite visual arts, mathematics and computer science.

Miguel Manoah Refugio Greenberg

AP-CS, ChatGPT and Me: a high school student perspective

Abstract

With the creation of openAI's ChatGPT system, a problem has arisen in introductory computer programming courses. Students now have the power to prompt ChatGPT with any computer programming question or problem they have been assigned and ChatGPT will generate a quick response for the student for free. Some computer scientists have predicted that by 2040 programming will become obsolete and the need for humans to know how to code will become useless because generative Artificial Intelligence (AI) will be able to do better than what humans do and faster. This study addresses this issue from the perspective of a high school student learning computer science through the Advanced Placement (AP) Computer Science curriculum at a public high school in California. After testing fifteen AP CS computer science course programming prompts and an open-ended project assignment, we conclude that in its current free version, ChatGPT was able to provide correct solutions about 66% of the time with the prompt as given 'as-is' in the assignment. However, the solutions to the AP course assignments were not correct all of the time, and occasionally the solution includes a fatal flaw that someone who does not know basic coding would not be able to identify or correct. This poster includes conclusions and recommendations from a high school student's perspective.

1 Introduction

A big problem has appeared in the world of computer science education and that is the use of ChatGPT in introductory computer programming courses. ChatGPT can quickly generate a response to almost any computer programming prompt you have and it is an easy, free way for students to finish their schoolwork in an instant without having to use their computer programming knowledge. The obvious flaw with using ChatGPT as a student beginning to learn computer programming is that the student is not learning anything by getting all their answers from generative AI.

I am a junior in high school and have taken two introductory computer programming classes offered at my school that cover various coding languages such as HTML and Java. For myself, while learning to code, I started to wonder about the following questions:

1. Can ChatGPT do as well as a high school student on AP Computer Science Java assignments?
2. Is it still useful to learn computer programming? Or will generative AI like ChatGPT take over and nullify the skills computer scientists have obtained and the teachings of computer

science to beginners?

This study addresses some of these questions.

Since ChatGPT was introduced in November, 2022, many scholars have begun to study its effect on education. A study of high school students found that 58% of high school students used ChatGPT on a daily basis in their everyday routines and for school-related tasks [1]. Several studies have focused on ChatGPT and computer science education, exploring whether ChatGPT can generate answers to introductory CS assignments and tests. For example, in one experiment, ChatGPT generated incorrect programming codes and also could not identify or solve its own errors [2]. In another exam experiment, ChatGPT only got a 20.4 out of 40 points [3]. In a study of generating answers to assignment questions about CS logic and theory, ChatGPT exhibited a "high degree of unreliability in answering a diverse range of questions pertaining to topics in undergraduate computer science" [4]. In another study, ChatGPT was used to complete assignments and tests for an introductory-level functional language programming course, and it only got a B- grade [5].

In another set of relevant studies, researchers investigated how ChatGPT could be used to aid students in computer science courses instead of how well ChatGPT itself performed in the courses. One study investigated the effect of using ChatGPT on undergraduate students' computational thinking skills, programming self-efficacy, and motivation toward the lesson. Half of the students in the study used ChatGPT during weekly programming assignments. Compared to the group that did not use ChatGPT, these students' "computational thinking skills, programming self-efficacy, and motivation for the lesson were significantly higher than the control group students" [6]. In a similar study, a group of students in a Data Structures and Algorithms college course were encouraged to use ChatGPT to solve programming challenges within a short period of time. Compared to students who only had textbooks and notes, the ChatGPT group earned higher scores [7]. Qureshi stated that knowing the variables that influence the results of ChatGPT is "crucial" to prompting ChatGPT to generate correct answers [7]. Similarly, many of the studies cited above mentioned that ChatGPT code generation becomes more accurate when initial answers from ChatGPT are improved through rephrased prompts.

The authors of these studies tend to agree that easy access to ChatGPT and other AI programs have both positive and negative consequences on computer science education. On the one hand, researchers predict that code reading and evaluating will become more important than code generation. In addition, if AI can produce basic code generation, then students can move on to more advanced computer science assignments more quickly [8]. According to Welsh, "The bulk of the intellectual work of getting the machine to do what one wants will be about coming up with the right examples, the right training data, and the right ways to evaluate the training process" [9]. On the other hand, these researchers think that ethics should become a more important aspect of teaching computer science. Two studies mentioned potential bias in the data that led to inaccurate coding. For example, ChatGPT was unable to generate accurate answers for an examination specific to the country of India. The authors suspect this is because ChatGPT training data includes less information about countries and contexts that are less represented on the internet [4].

My work departs from these studies summarized above because most of them were conducted in college courses and not high school courses. Also, the studies were conducted by professors and not by a high school student. A lot of people doing research like this already have a lot of experience in computer programming so I think I have a different perspective than others because I am still a beginner in computer programming. Finally, because I am new to coding, my advisor and I decided to use a rubric to help me evaluate the quality of the code we studied, and not just whether or not it could generate a correct answer. The studies above did not use rubrics to evaluate the performance of ChatGPT and focused more on how well ChatGPT did in generating correct answers.

2 Methods

This study involved:

- creating a rubric to evaluate functionality and quality of code in general
- comparing a students' responses and ChatGPT's responses to 15 AP Computer Science assignment prompts, given 'as-is' (i.e., how they were written in the assignment - no modified prompts)
- analyzing each ChatGPT response according to the rubric
- comparing three program outputs from a more open ended 'final project' style program (again from an AP CS high school course).

First, I identified four existing rubrics for evaluating code [10–13]. I chose these rubrics primarily because they were publicly available and from college computer science courses which is a good indicator that they are truthful and complete. Rubrics are a set of criteria for judging the functionality and quality of code. Different companies, engineers, and teachers have their own ideas for what constitutes high quality code [14, 15], but in comparing these four rubrics, there appears to be a set of common criteria that are widely accepted.

The four rubrics contain similar aspects that I deemed to be important rules for evaluating code such as whitespace and indentation. All four rubrics were published on the internet at the time of this study and included: Google (coding standards for source code in the Java Programming Language) [10], and three from Computer Science courses taught at: the United States Naval Academy [11], Illinois Wesleyan University [12], and Texas State University, San Marcos [13]. The merged final rubric I created from these four initial ones included these key/common criteria:

- White space
- Indentation
- Keep lines pretty short
- Good variable names
- Commenting
- Code efficiency

- Code works as expected

Second, I chose fifteen AP Computer Science assignments for this study which I had completed in my AP Computer Science course in spring, 2023. An assignment consisted of an assignment prompt (written description) and typically included uncompleted code. The prompt included an overview of the assignment telling the student what type of code they needed to implement. Typically, this consisted of adding new code into some uncompleted code to make it function correctly given the specifications. The fifteen assignments I chose vary in difficulty and also cover a wide variety of java concepts such as arrays, Boolean expressions, and iteration.

After creating the assessment rubric, I analyzed my answers and the ChatGPT answers against my merged rubric criteria to understand differences in how I wrote the code, how ChatGPT wrote the code, and how the assignments were graded in my class. For each assignment, each rubric criteria is given a score of low, medium, or high. A score of low meant it did not meet the criteria; medium meant it partially met the criteria; and high meant it completely met the criteria.

In a second component of this study, we compared ChatGPT's output for an open-ended project assignment (similar to the required 'final project' required in my AP CS course). The program was a final program created for the AP-CS course, which was a Pokemon-style turn-based text game. There was no existing prompt for this assignment as it was open ended. To create a prompt in which to use with ChatGPT, two different high school students played the working game and then generated text descriptions to prompt ChatGPT to write similar code. This portion of the study examined ChatGPT's capabilities when given student generated (not teacher generated) programming prompts.

3 Results and Discussion

Comparison of ChatGPT and Student Code According to Quality Rubric

	white space	indent	short lines	variable names	comments	code efficiency	works as expected
ChatGPT results							
low	0	0	0	1	12	3	3
medium	0	0	0	1	0	0	2
high	15	15	15	13	3	12	10
Student results							
low	0	0	0	0	10	1	0
medium	2	0	0	2	0	3	0
high	13	15	15	13	5	11	15

Based on the rubric, my analysis showed that overall, the student (me) performed slightly better in the AP course assignments than ChatGPT. ChatGPT scored highly for all quality criteria on only two assignments, whereas the student scored highly for all quality criteria on four assignments. Chat GPT performed better than the student in only two areas; whitespace and code efficiency. ChatGPT created especially high quality code related to the whitespace, indentation, and line length criteria, whereas the student created the highest quality code related to the indentation, line length, and code works as expected criteria. Importantly, apart from the quality of the code, ChatGPT generated code did not work as expected 33% of the time. We note that this finding is with respect to the prompts for the assignment ‘as-is’ - likely ChatGPT could be prompted to fix the code in many cases, but this is beyond the scope of this current study. Another important finding in the comparison between ChatGPT and student answers to assignment prompts is that ChatGPT almost never included comments about the code even though commenting on code is a fundamental standard practice of high quality code. Again, we note that likely ChatGPT could be prompted to add comments, but we find it interesting that it did not add comments as a default when generating code, although this is a known best practice.

The results for the second portion of this study, examining ChatGPT’s capabilities with respect to more open-ended prompts revealed further concerns about its use. Two different students ‘played’ working code and then created prompts for ChatGPT to program the same solution. While one prompt did produce a working version of a Pokemon-style game (it again did not contain comments and used very generic variable names), the other prompt resulted in ChatGPT producing code that included a fatal flaw. In brief, the game allowed the user to select a Pokemon character which played a turn-based attack game against the computer (i.e. another Pokemon object ‘controlled’ by the computer). In the fatally flawed coding solution produced by ChatGPT, prior to the user selecting a Pokemon type, the program allocated only two objects, one for each Pokemon type. One Pokemon-type was initially assigned to the computer and the other to the user. Then the program prompted the user for which Pokemon-type, the user wanted to select. If the user happened to select the same Pokemon-type as the one initially allocated for the computer (AI), the game played ‘against’ itself, with only one object battling itself and decrementing its own health score for every attack, both the computer’s turn and the players turn. This resulted in very short and inaccurate game play, with the one object battling itself. While the solution was ‘sometimes’ correct (i.e. if the user chose the type not initially selected and assigned to the computer in the initialization code), this kind of fatal flaw could prove detrimental for anyone unfamiliar with coding and unable to detect this kind of error.

4 Conclusion

Similar to the studies reviewed above, we conclude that this research project shows that ChatGPT would not have been an effective way for a student to cheat in the AP Computer Science course. To answer my second research question, I think it is still useful to learn how to code because currently, ChatGPT does not always generate correct answers. Even if ChatGPT becomes more advanced, it is still important for people to learn how to code because we need to understand what ChatGPT is doing. Also, no matter how advanced ChatGPT gets, it is still only getting its information from the internet, yet the internet does not contain equal amounts of information from every part of the world. Teachers should continue to teach coding and include ways that ChatGPT can improve learning instead of replace learning.

5 Acknowledgments

Thank you to Ms. Ashley Ong, AP-CS high school teacher for teaching me CS and to our paper editor.

References

[1] N. Forman, J. Udvaros, and M. S. Avornicului, “Chatgpt: A new study tool shaping the future for high school students,” *International Journal of Advanced Natural Sciences and Engineering Researches*, vol. 7, no. 4, p. 95–102, May 2023. [Online]. Available: https://as-proceeding.com/index.php/ijanser/article/view/562

[2] F. M. Megahed, Y.-J. Chen, J. A. Ferris, S. Knoth, and L. A. Jones-Farmer, “How generative AI models such as ChatGPT can be (mis)used in SPC practice, education, and research? an exploratory study,” *Quality Engineering*, pp. 1–29, jun 2023. [Online]. Available: https://doi.org/10.1080%2F08982112.2023.2206479

[3] S. Bordt and U. von Luxburg. (2023) Chatgpt participates in a computer science exam. [Online]. Available: arXivpreprintarXiv:2303.09461

[4] I. Joshi, R. Budhiraja, H. Dev, J. Kadia, M. O. Ataullah, S. Mitra, D. Kumar, and H. D. Akolekar, “Chatgpt in the classroom: An analysis of its strengths and weaknesses for solving undergraduate computer science questions,” 2023.

[5] C. Geng, Y. Zhang, B. Pientka, and X. Si, “Can chatgpt pass an introductory level functional language programming course?” 2023.

[6] R. Yılmaz and F. G. Karaoğlan Yılmaz, “The effect of generative artificial intelligence (ai)-based tool use on students’ computational thinking skills, programming self-efficacy and motivation,” *Computers and Education: Artificial Intelligence*, vol. 4, p. 100147, 06 2023.

[7] B. Qureshi, “Exploring the use of chatgpt as a tool for learning and assessment in undergraduate computer science curriculum: Opportunities and challenges,” 2023.

[8] B. A. Becker, P. Denny, J. Finnie-Ansley, A. Luxton-Reilly, J. Prather, and E. A. Santos, “Programming is hard - or at least it used to be: Educational opportunities and challenges of ai code generation,” in *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, ser. SIGCSE 2023. New York, NY, USA: Association for Computing Machinery, 2023, p. 500–506. [Online]. Available: https://doi.org/10.1145/3545945.3569759

[9] M. Welsh, “The end of programming,” *Commun. ACM*, vol. 66, no. 1, p. 34–35, dec 2022. [Online]. Available: https://doi.org/10.1145/3570220

[10] I. Google, “Google java style guide,” july 2023. [Online]. Available: https://google.github.io/styleguide/javaguide.html

[11] U. N. Academy, “Us nvaal academy rubric,” july 2023. [Online]. Available: https://usna.edu/Users/cs/norine/ic312f23/resources/rubrics.html

[12] M. Liffiton, “Illinois weslyan university,” july 2023. [Online]. Available: https://sun.iwu.edu/~mliffito/class/2011f/cs127/rubric.phpl

[13] V. Metsis, “Texas state rubric,” july 2023. [Online]. Available: https://userweb.cs.txstate.edu/~v_m137/cs3354_fall2016/Grading%20Rubric.pdf

[14] M. Stegeman, E. Barendsen, and S. Smetsers, “Towards an empirically validated model for assessment of code quality,” in *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 99–108. [Online]. Available: https://doi.org/10.1145/2674683.2674702

[15] H. Keuning, J. Jeuring, and B. Heeren, “A systematic mapping study of code quality in education,” in *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, ser. ITiCSE 2023. New York, NY, USA: Association for Computing Machinery, 2023, p. 5–11. [Online]. Available: https://doi.org/10.1145/3587102.3588777