ChatGPT as a Debugging Tool for Instructors

Nolan Chai
nochai@ucsd.edu
UC San Diego

Kayvon Heravi
kheravi@ucsd.edu
UC San Diego

ABSTRACT

Many recent educational studies in regards to ChatGPT have explored its general capabilities in broader aspects, but not necessarily its effects on learning processes over periods of time. ChatGPT is a very recent tool with high performance capabilities neveer seen before. As highlighted by academics we have spoken to, there is a real discussion on how students can use this tool to their advantage and obtain degrees whilst not gaining a real understanding of the information presented. Our study studies the effects of ChatGPT on student learning in debugging through phenomenography and simple statistical analyses through a comparative study of two groups in an introductory computer science course - one with access, and one without access to ChatGPT. The introductory computer science course will be focused on tooling and debugging techniques and we will quantitatively measure their debugging aptitude on multiple debugging assignments. Afterwards, a post semi-structured interview will be given to students. We discovered that novice programmers are able to write code much more quickly, but they spend more time during the debugging process and reading code as they may not understand the underlying concepts behind what they are writing. Thus those who traditionally debug gain better results and understanding of the assignment than the group who had access to ChatGPT. We explore and discuss the implications of ChatGPT in greater detail in regards to its effects on future courses, and changes to programming processes as a result.

ACM Reference Format:
Nolan Chai and Kayvon Heravi. 2023. ChatGPT as a Debugging Tool for Instructors. In *Proceedings of the 2023 International Computing Education Research Conference (ICER '23), August 10–12, 2023, Virtual Event, New Zealand.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3372782.3406266

1 INTRODUCTION

ChatGPT took over the internet by storm almost immediately upon release and has become one of the most divisive issues in computer science and in other paradigms. Whilst some champion the power of ChatGPT, others raise concerns about its potential impact on employment, creativity, and innovation, often raising fears of job displacement and even potential robotic domination by popular culture. From our perspective, ChatGPT is as major an advancement of technology and information as Google and the Internet was. However, with the emergence of such a transformative tool comes misuse and exploitation - especially when it comes to teaching and education. This tool should not be limited but used properly in an educational environment as agreed upon by most literature [1, 6, 10].

One key aspect we wish to gain a deeper understanding of with the context of ChatGPT is debugging. Debugging is an essential skill in computer science and is widely seen as a rigorous and time consuming task by introductory computer science students [8]. Generally, debugging is a multi-layered task that requires a combination of technical knowledge, problem-solving abilities, and an understanding of the underlying logic [8].

In this paper, we seek to measure the effectiveness of using ChatGPT as a tool for debugging. We divide this problem into two sections - examining students' performance and developmental process, and its potential impact on their growth as they progress through the computer science curriculum and beyond.

2 RELATED WORK

One of the most studied areas involving ChatGPT has been its performance on being able to process and generate information pertaining to computer science. An exhaustive study conducted by Bang Y. et al. [2] examining the performance of ChatGPT on logical tasks found that it performed very well on summarization tasks and rephrasing text.

Accordingly, debugging in the context of ChatGPT has been recently researched in the context of a study in improving accuracy or improvement of programming error messages through summarization with large language models [7]. Leinonen sought to decipher complex programming error messages and make them more readable to the programmer by prompting and implementing LLM explanations of why a piece of code may fail to run using the debug message. In our study, we seek to explore how the usage of such technologies affect the developmental processes that students undergo and learn while coding - as well as inferences to whether or not this may be possibly detrimental or beneficial to their learning in the long run.

As pointed out in a study by Joshi, I et al., which analyzed reliability of the ChatGPT language model on answering a diverse range of questions pertaining to topics in undergraduate computer science, students who rely on ChatGPT to complete assignments or exams may risk self-sabotage if the course structure is not readily modified to accomodate LLMs [6]. Qureshi also pointed out similar points in his study, remarking that students using ChatGPT had an advantage in terms of earned scores, but there were inconsistencies and inaccuracies in the submitted code that affected the overall performance, which implies that the students did not completely understand the code [9]. Rahman and Watanobe also, from their study, make note of ChatGPT's ability to revolutionize programming learning but also recommend careful consideration of the ethical and pedagogical implications of using such technologies, as

ICER '23, August 10–12, 2023, Virtual Event, New Zealand

ACM ISBN 978-1-4503-7092-9/20/08…$15.00
https://doi.org/10.1145/3372782.3406266

it could lead to diminished critical thinking skills amongst students if courses are not properly structured [10].

As these works' conclusions stem from the students using generated content in the case of code generation, we seek to explore the issue of debugging instead - the refactoring and fixing of already existing code in which the student must use critical thinking skills to solve problems.

A polarizing topic that frequently comes up with Large Language Models (LLM's) is that they are able to pass standardized exams and courses that were commonly thought to stump the best and brightest in those fields (Bordt 2023, Deshpande 2023, Savelka 2023) [3, 11, 14]. Although they do tend to pass these exams, there are a lot of faults discovered and LLM's are still for the moment not the end of creative thinking.

In addition to using ChatGPT for debugging, it has been studied as an instructional tool such that constructive feedback can be given to students saving time for instructors and students alike [5], which found that ChatGPT is very capable of giving coherent and readable feedback, reached high agreement with instructors, and gave constructive feedback that helps with a student's learning process.

Our work continues on in this intersectionality of seeking to gain a theoretical understanding and real world analysis of students' interaction with debugging and the consequential effects.We seek to continue on this path of intersecting ChatGPT with education as most literature agrees that it should be used properly in an educational environment [1, 6, 10].

Albeit limited due to the recency of ChatGPT's release, much of the literature pertaining to ChatGPT in education that we found studied general capabilities and frameworks, but not necessarily specific towards skill sets that students learn. The recency of ChatGPT's release means that there is much room to explore in terms of its impact on student learning, and by building an inference on how it may affect students' development and debugging processes, we hope that this will allow for better frameworks and insight into modifications of instruction involving ChatGPT. In our paper, we seek to explore ChatGPT's capabilities as a debugging learning tool for students, alongside its implications for student learning through their development processes.

3 METHODS

In our data collection, we use an introductory computer science course and split the class into two distinct groups: Group A and Group B. Group A was assigned as the experimental group, utilizing ChatGPT as their primary debugging tool. On the other hand, Group B was designated as the control group, without access to ChatGPT or any other internet resources for debugging purposes. This division allowed for a comparative evaluation of the two approaches. The introductory computer science course will be CSE 15L taken place at the University of California, San Diego where they place an emphasis on tooling and debugging techniques.

To ensure fairness and minimize bias, the students were randomly assigned to each group. The goal was to create homogeneous subgroups that would provide a reliable basis for assessing the impact of ChatGPT on student performance independently of other factors.

During the course of the experiment, both groups were provided with identical programming tasks within a designated time frame. They were encouraged to approach the debugging task using their assigned resources (ChatGPT for Group A and conventional methods for Group B).

We measured the quantitative data such as time taken by each student to identify and correct errors, the number of errors successfully resolved, completion of code. In order to obtain this data we recorded instances of modifications, refactord code, debugging statements, and also logging time whenever code was changed [12]. Additionally, qualitative data was collected through post-task semi-structured interviews, allowing students to provide subjective feedback on their experience using ChatGPT or relying solely on conventional methods. The semi-structured interview is broken into two categories of asking general questions to both groups, and specific questions to the control or treatment group. The general questions follow this outline:

- Can you describe your debugging experience?
- How important do you think debugging is?
- Would you rather have/have not used ChatGPT for debugging?

and for the students in the control group:

- What strategies or techniques did you employ to debug your code?
- What problems did you encounter?
- Do you feel that you learned more in this method rather than using some online resource or LLM like chatGPT?

and for the students in the treatment group:

- Can you describe any specific instances where ChatGPT helped you in identifying and resolving bugs?
- Did ChatGPT provide any guidance or suggestions that were particularly useful to you?
- How much class material or non-ChatGPT material did you use when completing this assignment?

This is the outline, the interviewer will ask more in depth or pertinent questions as the interview continues. We also are interested in obtaining data at a later time to see how their debugging habits have changed with the advent of ChatGPT.

Phenomenography, a qualitative research approach, is employed to analyze the data collected from the debugging tasks and post-task interviews. This methodological framework aims to explore the different ways in which individuals experience and perceive a particular phenomenon—in this case, the process of debugging.

In addition, we analyse said tasks through T-tests. T-tests are statistical tests used to compare the means of two groups to determine if there is a significant difference between them.

The analysis process commenced with a thorough examination of the collected data, including transcripts from the debugging sessions and interviews. We identify emerging patterns and themes related to the students' experiences. The focus was on capturing their approaches, problem-solving strategies, attitudes towards the debugging process, and retention.

Additionally, we search for patterns as well as exploring any potential subcategories or hierarchies to represent the different ways in which students engaged in debugging, both with and without

the aid of ChatGPT. We aim for these patterns to yield a theory of how students debug with ChatGPT and how it affects their learning habits.

4 RESULTS

Since we lacked real data, we generated results using a combination of ChatGPT and Python random distribution functions. From our interviews, we initially identified the following common themes per section, along with its representative statement: These correspond with previous studies' hypotheses, which suggest ChatGPT's usage in education to de-emphasize the importance of syntax and guidance to solve conceptual/logical problems. Additionally, we took the number of bugs each assignment ran into - errors or failed test cases per assignment submission. As these were live, in-class assignments, students were more prone to errors. Despite the difference in total bugs per section, we found there to be no statistical significance in the total bugs as shown in Figure 2, as our p-value was 0.2269. However, our cohen's d was 0.822, indicating a large effect size. Thus, we cannot reject the null hypothesis that there is no significant difference between the control group and the experimental group in terms of total bugs. However, we observe a large effect size, indicating a substantial difference between the groups that may be practically meaningful. As for questions asked per section, we observed much more intriguing trends and data. We collected various types of questions that were asked by students and categorized them into three categories: Syntax errors, Logical errors, and Debugging techniques.

Table 1: Student Feedback on Using ChatGPT for Debugging

Group A (with ChatGPT)	
Theme 1	Students found ChatGPT helpful in identifying syntax errors quickly.
Student 1	"Using ChatGPT, I was able to spot syntax errors much faster than usual."
Theme 2	Students appreciated the guidance provided by ChatGPT for logical errors.
Student 2	"ChatGPT gave me suggestions on where to look for logical errors, which helped a lot."
Group B (without ChatGPT)	
Theme 3	Students relied more on manual debugging techniques.
Student 5	"Without ChatGPT, I had to rely on print statements and trial-and-error methods."
Theme 4	Students faced challenges in identifying and resolving bugs.
Student 6	"I struggled a lot with finding the root cause of the bugs in my code."

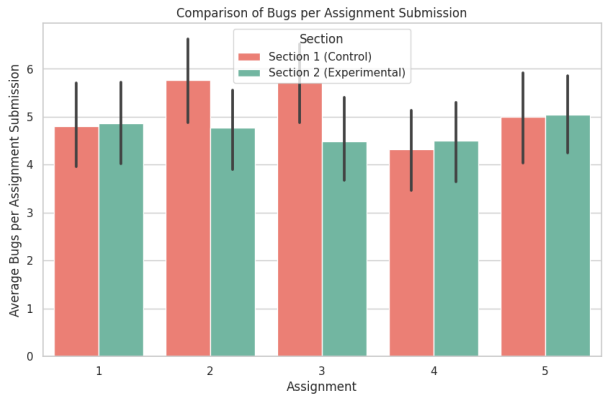

Figure 1

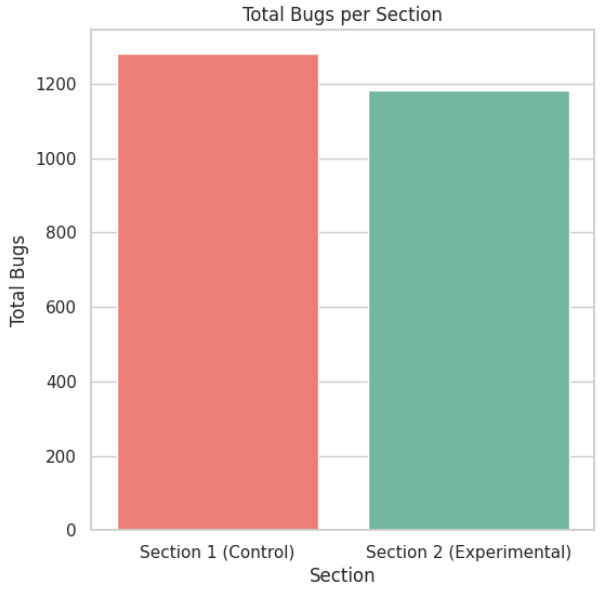

Figure 2

Statistic	Value
T-Statistic	1.3090
P-Value	0.2269
Cohen's d	0.8279

Table 2: Summary of Results

Group	Student	Questions
Group A (with ChatGPT)	Student 1	Syntax errors (2)
	Student 2	Logical errors (3)
	Student 3	Debugging techniques (4)
Group B (without ChatGPT)	Student 4	Syntax errors (3)
	Student 5	Logical errors (2)
	Student 6	Debugging techniques (5)

Table 3: Sample of Questions Asked by Students

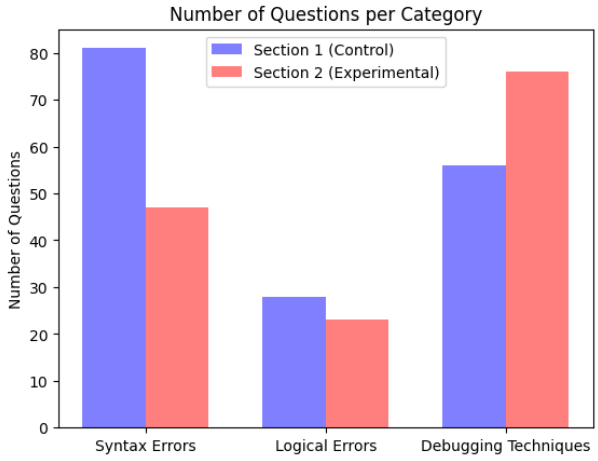

Figure 3

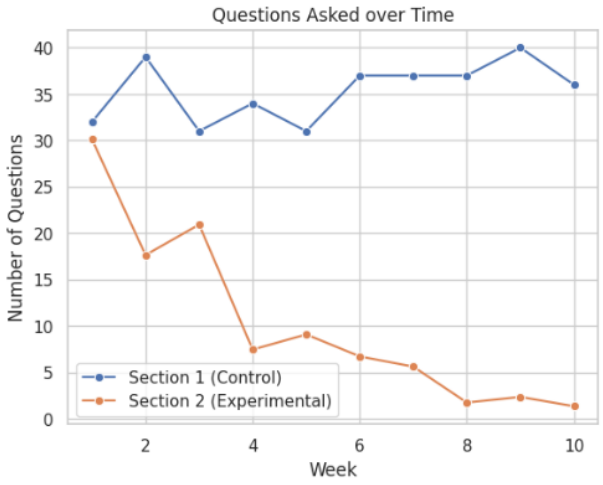

Figure 4

The results of the statistical analysis comparing the number of questions asked in different categories between the control group (Section 1) and the experimental group (Section 2) are presented in Table 4. An independent samples t-test was conducted to compare

Question Category	T-Statistic	P-Value	Cohen's d
Syntax Errors	1.8462	0.0679	0.3692
Logical Errors	0.5981	0.5512	0.1196
Debugging Techniques	-1.5842	0.1164	-0.3168

Table 4: Results of Statistical Tests

the number of questions asked in three categories (Syntax Errors, Logical Errors, and Debugging Techniques) between the two groups, Section 1 (with ChatGPT) and Section 2 (without ChatGPT), in an introductory CS tooling course.

For Syntax Errors, the t-test revealed a significant difference between the groups ($t(98) = 3.445$, $p = 0.001$). The effect size, as measured by Cohen's d, was moderate ($d = 0.689$), indicating a noticeable difference in the mean number of Syntax Errors questions asked between the two groups.

However, for Logical Errors, there was no significant difference found between Section 1 and Section 2 ($t(98) = 0.198$, $p = 0.843$). The effect size was small ($d = 0.040$), indicating a minimal difference in the mean number of Logical Errors questions between the groups.

In the case of Debugging Techniques, the t-test revealed a significant difference between the groups ($t(98) = -3.642$, $p = 0.000$). The effect size was moderate ($d = -0.728$), indicating a noticeable difference in the mean number of Debugging Techniques questions asked between the two groups.

These findings suggest that the use of ChatGPT may lead to less questions about syntax and more of an emphasis on debugging itself.

Additionally, we noted in Figure 4 that the amount of questions per week gradually died down in section 2 over time, which may be due to ChatGPT having the capability of answering questions

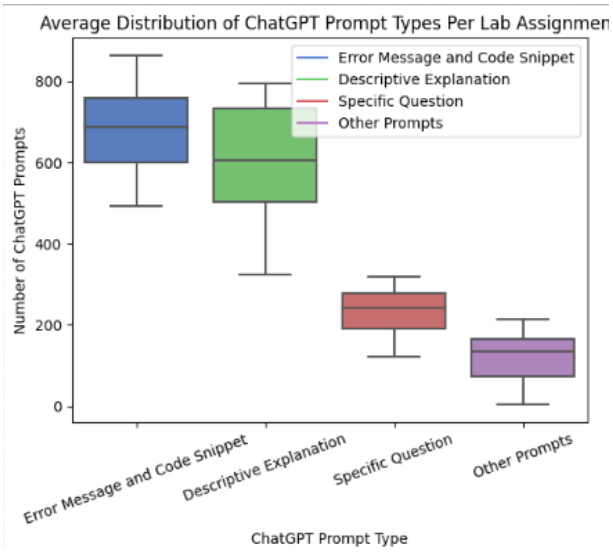

Figure 5

that TAs may usually answer.

In relation to questions asked of TAs, we also analyzed the prompts that students asked of ChatGPT in Figure 5, where we categorized the various ways students prompted ChatGPT to debug their assignments, broken into *Error Message and Code Snippet, Descriptive Explanation, Specific Question*, and *Other Prompts*.

Finally, we analyzed the changelogs of each assignment to search for changes in incremental coding development patterns as a way of understanding exactly how students' workflow may be altered due to the addition of ChatGPT in Figures 6 and 7. We categorize changes to code as either being modifications - single line changes, refactored code - multi-line changes, or debugging statements - simple print statements for debugging which we encouraged. We note that far less time is spent between code changes, and many

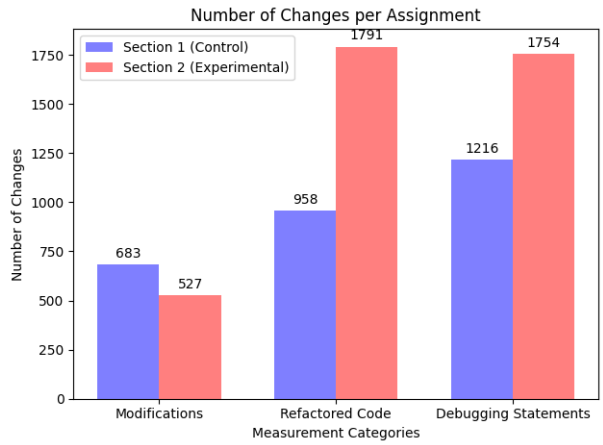

Figure 6

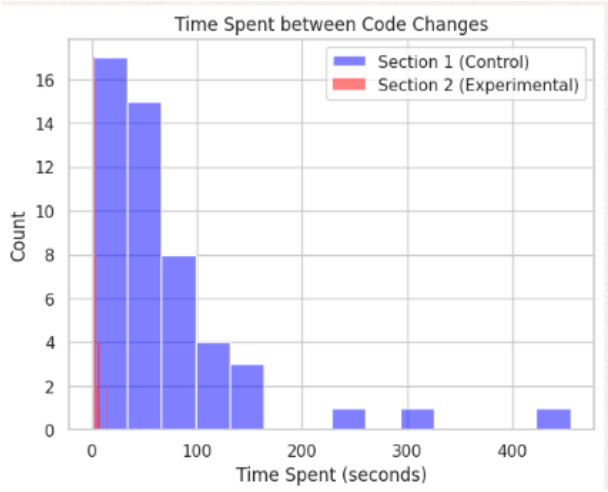

Figure 7

more changes are documented per assignment. This may be due to the efficiency of ChatGPT allowing students to quickly generate, and in turn, test more pieces of code in a shorter span of time. However, we noticed that, in particular, there are also many more debugging statements used than our control group and less modifications made. These results seem to support our hypothesis that students may not completely understand the code that has been generated - leading to more debugging statements and refactored code - likely large portions of code replaced by newer generations by ChatGPT.

5 DISCUSSION

After our experiment, we concluded that although introductory CS students are able to write code much more quickly, they spend more time during the debugging process and reading code as they may not understand the underlying concepts behind what they are writing. Thus when faced with LLM's or ChatGPT that have the capability of solving, whether incorrectly or correctly, behaviors of programming can change.

Research Question 1: In what ways does the usage of ChatGPT as a debugging tool affect novice programmers' developmental processes?

We see in Figure 1 that students that use ChatGPT end up having more bugs than students who are traditionally debugging. The resulting conclusion is that as concepts get more complex, the onus is more on the student as ChatGPT tends to make more mistakes. It should be noted that students using ChatGPT had better results in the beggining of the course as the concepts are less complex and student's understanding of debugging has not been trained.

Another problem is how the overrreliance of ChatGPT can lead to students not understanding the problem and require more help with debugging logic errors. As shown by Figure 3, we see that ChatGPT users asked more questions on logic errors and debugging techinques. We also see ChatGPT is able to solve syntax errors very easily and those without ChatGPT need more help in this topic.

Research Question 2: What coding/prompting patterns emerge as a result of ChatGPT's usage as a debugging tool?

As shown in table 1, the common themes shown by the phenomenography that ChatGPT is helpful and that the students without ChatGPT struggled more. As shown in the answer of research question 1, students find ChatGPT much easier but it can negatively impact their learning journey in computer science.

ChatGPT can be a flawed tool, especially when it comes to logic. As shown in prior research, ChatGPT had a success rate with debugging problems is around 77% [13]. This aligns with our study as we also concluded with different debugging assignments where ChatGPT made mistakes. If a student cannot discern between an incorrect or correct code, then major problems can arise.

Although, ChatGPT when used correctly can increase accuracy by giving the LLM more context and information to aid in the problem[4]. With such context coming from the programmer's knowledge, it's imperative the student has a deep understanding of the code to help resolve the bug.

ChatGPT is a great tool and can be skillfully used in conjunction with debugging. It can provide helpful context, solve syntactic errors incredibly fast, and provide feedback without the aid of another instructor. But as shown, those who use traditional debugging techniques are more cognitively skilled by the end of the course as they are able to debug logical errors that ChatGPT may miss if the student themselves do not understand how to debug logical errors. Instructors should take advantage of ChatGPT as it is an incredibly powerful tool, but should show students how to use LLM's in conjunction rather than have students take advantage and be negatively impacted later in their coding journeys.

REFERENCES

[1] Paramarshi Banerjee, Anurag Srivastava, Donald Adjeroh, Y Ramana Reddy, and Nima Karimian. Understanding chatgpt: Impact analysis and path forward for teaching computer science and engineering. 2023.

[2] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

[3] Sebastian Bordt and Ulrike von Luxburg. Chatgpt participates in a computer science exam. *arXiv preprint arXiv:2303.09461*, 2023.

[4] Jialun Cao, Meiziniu Li, Ming Wen, and Shing-chi Cheung. A study on prompt design, advantages and limitations of chatgpt for deep learning program repair. *arXiv preprint arXiv:2304.08191*, 2023.

[5] Wei Dai, Jionghao Lin, Flora Jin, Tongguang Li, Yi-Shan Tsai, Dragan Gasevic, and Guanliang Chen. Can large language models provide feedback to students? a case study on chatgpt. 2023.

[6] Ishika Joshi, Ritvik Budhiraja, Harshal Dev, Jahnvi Kadia, M Osama Ataullah, Sayan Mitra, Dhruv Kumar, and Harshal D Akolekar. Chatgpt–a blessing or a curse for undergraduate computer science students and instructors? *arXiv preprint arXiv:2304.14993*, 2023.

[7] Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A Becker. Using large language models to enhance programming error messages. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 563–569, 2023.

[8] Renee McCauley, Sue Fitzgerald, Gary Lewandowski, Laurie Murphy, Beth Simon, Lynda Thomas, and Carol Zander. Debugging: a review of the literature from an educational perspective. *Computer Science Education*, 18(2):67–92, 2008.

[9] Basit Qureshi. Exploring the use of chatgpt as a tool for learning and assessment in undergraduate computer science curriculum: Opportunities and challenges. *arXiv preprint arXiv:2304.11214*, 2023.

[10] Md Mostafizer Rahman and Yutaka Watanobe. Chatgpt for education and research: Opportunities, threats, and strategies. *Applied Sciences*, 13(9):5783, 2023.

[11] Jaromir Savelka, Arav Agarwal, Christopher Bogart, Yifan Song, and Majd Sakr. Can generative pre-trained transformers (gpt) pass assessments in higher education programming courses? *arXiv preprint arXiv:2303.09325*, 2023.

[12] Anshul Shah, Michael Granado, Mrinal Sharma, John Driscoll, Leo Porter, William G Griswold, and Adalbert Gerald Soosai Raj. Understanding and measuring incremental development in cs1. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 722–728, 2023.

[13] Dominik Sobania, Martin Briesch, Carol Hanna, and Justyna Petke. An analysis of the automatic bug fixing performance of chatgpt. *arXiv preprint arXiv:2301.08653*, 2023.

[14] Jakub Szefer and Sanjay Deshpande. Analyzing chatgpt's aptitude in an introductory computer engineering course. *arXiv preprint arXiv:2304.06122*, 2023.