Does ChatGPT Help With Introductory Programming? An Experiment of Students Using ChatGPT in CS1

Yuankai Xue*
yuankai.xue@vanderbilt.edu
Vanderbilt University
Tennessee, USA

Hanlin Chen*
hanlin.chen@vanderbilt.edu
Vanderbilt University
Tennessee, USA

Gina R. Bai
rui.bai@vanderbilt.edu
Vanderbilt University
Tennessee, USA

Robert Tairas
robert.tairas@vanderbilt.edu
Vanderbilt University
Tennessee, USA

Yu Huang
yu.huang@vanderbilt.edu
Vanderbilt University
Tennessee, USA

ABSTRACT

Generative AI, notably ChatGPT, has garnered attention in computer science education. This paper presents a controlled experiment that explores ChatGPT's role in CS1 in a classroom setting. Specifically, we aim to investigate the impact of ChatGPT on student learning outcomes and their behaviors when working on programming assignments. Participants were tasked with creating a UML diagram and subsequently implementing its design through programming, followed by a closed-book post-evaluation and a post-survey. All the participants were required to screen-record the whole process. In total, 56 participants were recruited, with 48 successful screen recordings. Participants in the Experimental Group can access ChatGPT 3.5 and other online resources, such as Google and Stack Overflow when creating the UML diagram and programming; however, participants in the Control Group can access all online resources except for ChatGPT (i.e., the only design variable is the access to ChatGPT). Finally, we measured and analyzed participants' learning outcomes through their UML diagram, programming, and post-evaluation scores. We also analyzed the time participants took to complete the tasks and their interactions with ChatGPT and other resources from the screen recordings. After finishing the tasks, student participants also provided their perceptions of using ChatGPT in CS1 through a post-survey.

With rigorous quantitative and qualitative analysis, we found that (1) using ChatGPT **does not present a significant impact** on students' learning performance in the CS1 assignment-style tasks; (2) once using ChatGPT, students' tendency to explore other traditional educational resources is **largely reduced** (though available) and they tend to rely solely on ChatGPT, and this reliance on ChatGPT did not guarantee enhanced learning performance; (3) the majority of students hold neutral views on ChatGPT's role in CS1 programming but most of them raised concerns about its potential ethical issues and inconsistent performance across different tasks.

We hope this study can help educators and students better understand the impact of ChatGPT in CS1 and inspire future work to provide proper guidelines for using ChatGPT in introductory programming classes.

*Both authors contributed equally to this paper.

ICSE-SEET '24, April 14–20, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0498-7/24/04...$15.00
https://doi.org/10.1145/3639474.3640076

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

CS education, CS1, Generative AI, ChatGPT, OOP

ACM Reference Format:
Yuankai Xue, Hanlin Chen, Gina R. Bai, Robert Tairas, and Yu Huang. 2024. Does ChatGPT Help With Introductory Programming? An Experiment of Students Using ChatGPT in CS1. In *46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '24), April 14–20, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3639474.3640076

1 INTRODUCTION

The rapid advancements in machine learning and natural language processing (NLP) have given rise to groundbreaking innovations, among which generative AI stands out [1]. Generative AI harnesses deep learning models to generate a variety of human-like responses, through diverse and intricate prompts [21]. Driven by generative AI, ChatGPT has attained widespread recognition and has been used as a leading conversational AI model [1, 31]. ChatGPT has led to increasing discussion and working style changes in society and various industries [14, 24, 37]. Among them, education is shown as a domain where ChatGPT may have both negative and positive impacts on [1, 16, 28]. For instance, it might help teachers create and design educational content or assist students to learn vocabulary and create texts [11, 20]. However, it might also negatively affect education, such as fostering misunderstandings of the material, diminishing cognitive engagement, and plagiarism [6, 10, 34, 35].

In computer science (CS), research has leveraged ChatGPT (and other Large Language Models such asBERT, Transformer-XL, and RoBERTa) to effectively assist various programming tasks including code completion [12], program repair [33], code explanations [9], and text-to-code conversion [8]. The output quality of ChatGPT in these tasks can highly depend on the training data and prompt design [33]. Thus, researchers recommend using ChatGPT under guidance as a supplementary tool [4, 33]. In contrast to non-STEM education and computational research, albeit to a lesser extent currently, researchers have also initiated investigations into the influence of ChatGPT on CS education, especially considering the unique characteristics, such as practical programming skills, functionality requirement, and expertise-based judgment [2]. For example, Qureshi investigated ChatGPT's capability to support students in algorithmic problems [30]. Jalil et al. evaluated the performance of ChatGPT in answering practice questions in a popular software testing curriculum [15].

However, these studies mainly focused on ChatGPT's assistance to more experienced programmers and advanced tasks. CS education for novices also holds significant importance for the future of technology and research. Considering the limited programming experience that CS1 students have, there are unique challenges to investigate. For example, can ChatGPT help CS1 students achieve better learning performance? Will ChatGPT diminish students' opportunities to problem solve? Can CS1 students leverage the capability of ChatGPT properly? Is ChatGPT more helpful for one type of introductory-level programming task than another? What are CS1 students' attitudes toward ChatGPT? Few studies focus on how ChatGPT can support novice programmers. For instance, Kazemitabaar et al. investigated OpenAI Codex's influence on novice programmers' code-authoring skills. Their study, involving 69 novice programmers, revealed that Codex not only enhanced code generation capabilities but also maintained participants' proficiency in manual code alterations, showcasing Codex's balanced functionality in both aiding and preserving core coding skills [17].

Therefore, to build on current research, we conducted an experimental, controlled A/B study to evaluate the potential impact of ChatGPT 3.5 on students' learning outcomes in a CS1 classroom setting. Student participants were tasked with creating a UML diagram and implementing it by programming in Java. While UML concepts had been introduced in the lectures and students had previously interpreted them in past assignments, this was their first experience in generating a UML diagram. This assignment emphasized active learning and individual understanding of the topics, as the lectures prior to the experiment had only covered fundamental concepts without an in-depth exploration of UML diagrams. In the experiment, student participants were divided into two groups (see more details in Section 3.4). Participants in the Control Group were restricted from using ChatGPT during the UML diagram and programming tasks. However, they could access online resources, like course slides or fixed-content web pages. In contrast, the Experimental Group had the freedom to use ChatGPT and any other online resources for the UML diagram and programming tasks. The tasks for this experiment were presented to the participants as an optional extra credit assignment. Participants independently tackled these tasks. As they progressed through the UML diagram creation, programming, and post-evaluation phases, their screen activities were continuously recorded. These recordings were subsequently reviewed to ensure that participants adhered to the study's set guidelines and procedures. Following the UML and programming tasks, participants underwent a closed-book, closed-note post-evaluation in the form of a quiz. This evaluation, along with a subsequent survey, gauged participants' learning outcomes. The survey allowed participants who were allowed to use ChatGPT to provide feedback regarding interactions with the tool. The study procedure is shown in Figure 1.

With rigorous quantitative and qualitative analysis of a total of 65.5 hours of screen recordings from 48/56 participants (8 participants did not submit screen recordings or the submitted videos were not valid), learning performance measurement (i.e., UML diagram, codes, and post-evaluation), and post-survey data, we found that (1) employing ChatGPT doesn't substantially alter students' performance in assignment-like CS1 tasks; (2) engaging with ChatGPT results in students' diminished exploration of other available educational resources, leading them to predominantly depend on ChatGPT. However, this dependence doesn't necessarily enhance their learning outcomes; (3) most students remain ambivalent about the role of ChatGPT in CS1 programming, though many express concerns about potential ethical dilemmas and its variable effectiveness across tasks. Students suggest ChatGPT's capability may not be reliable across different programming tasks they encountered during their studies. The majority of students are more concerned about the potential ethical issues associated with ChatGPT.

The main contributions of this paper are as follows:

- A controlled study to investigate the impact of ChatGPT in CS1 programming;
- Presented analysis into the impact of ChatGPT on CS1 students' learning outcomes and resources utilization;
- Investigated students' perception of ChatGPT in the context of CS1;
- Provided discussions and suggestions for students and educators on leveraging ChatGPT to support CS1.

We hope our study can inspire future work to further explore the impact of ChatGPT on CS1, as well as to design more effective and efficient guidelines for integrating ChatGPT in CS1. All of the study materials including task design, protocols and instructions, rubrics, and deidentified participant data can be found in the anonymous supplementary package available online[1].

2 RELATED WORK

In this section, we discuss related work on ChatGPT's usage in education, computer science, computer science education, and CS1 education specifically.

ChatGPT in Education: Researchers have been devoted to investigating the use of ChatGPT in education. Sallam et al. showed that ChatGPT could be helpful in healthcare education by generating health reports, improving diagnostics, and increasing the accessibility of medical publishings [32]. Pardos et al. indicated that 70% of the hints provided by ChatGPT facilitated students' learning process in high school algebra [29]. Other studies found that ChatGPT and the revolutionary workflow around it were powerful and applicable for English language learners to learn writing and reading [16, 20, 38]. However, researchers also pointed out the limitations of ChatGPT, such as the increasing plagiarism in higher education, fabricating non-existing rules or equations to generate solutions, and diminishing students' innovative capacities and critical thinking [6, 10, 34, 35].

[1]https://zenodo.org/record/8265297

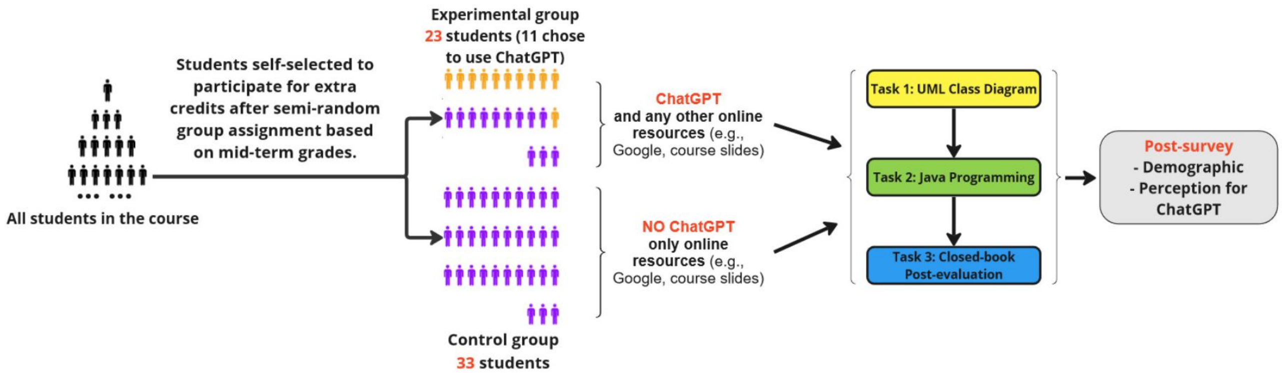

Figure 1: Study Procedure. Students were first semi-randomly assigned to the experimental and Control Group and then self-selected to participate the study. Both groups took the same three tasks. During the three tasks, with equal access to any other online resources, the Experimental Group was granted access to ChatGPT while the Control Group was not. Finally, all participants took a post-survey.

Generative AI in CS and CS education: Numerous researchers have been exploring the use of generative AI in CS and CS education. As one of the most advanced generative AI tools based on Large Language Models, ChatGPT has also been leveraged in various computational tasks [8, 9, 12, 33]. For example, Feng et al. indicated that ChatGPT excels at understanding code syntax; however, it has some difficulties comprehending code semantics, particularly dynamic semantics [12]. Researchers also compare ChatGPT with other generative AI. For instance, Chen et al. showed that GPTutor delivers more concise and accurate coding explanations than that provided by GitHub Copilot. ChatGPT is centered around natural language interactions, whereas GitHub Copilot is tailored for coding and software development tasks [9].

Meanwhile, researchers have also attempted to integrate ChatGPT into CS education and relevant scenarios. In a recent study by Banerjee et al., researchers suggested that students can enhance their learning by engaging in a collaborative analysis of programming tasks conducted by ChatGPT, focusing on factors including answer efficiency, code complexity, and readability [2]. Another investigation conducted by Macneil et al. showcased the potential of ChatGPT in explaining code concepts to CS students, such as analyzing and explaining time complexity and bug repair [25]. Also, it's worth acknowledging that certain limitations of ChatGPT have been identified by various researchers as well. Qureshi's study[30] compared two student groups: one with access to traditional programming material and another encouraged to use ChatGPT for programming help. Results showed ChatGPT's efficacy mainly in simpler data structure problems regarding test cases, time complexity, and memory. However, ChatGPT struggled with complex queries, requiring multiple prompts and increasing students' cognitive load [30]. Jalil et al. showed that ChatGPT could only provide correct or partially correct explanations to 53.0% of the software testing practice questions [15]. Their work indicated that ChatGPT's capabilities in facilitating CS education are unreliable in all circumstances.

ChatGPT in CS1 education: As the influence of generative AI like ChatGPT expands, several researchers began to explore their utility in CS1 Education. Kiesler et al. have underscored the potential of ChatGPT in providing formative feedback for introductory programming tasks. Their research offers insights into the efficacy of ChatGPT-3.5 and GPT-4 in handling 72 introductory Python programming tasks sourced from the CodingBat website. Their findings suggest that these LLMs can achieve high accuracy rates between 94.4% and 95.8% in task completion, consistently providing textual explanations and program code [18, 19]. Sam et al. interviewed 20 introductory programming instructors and their opinions on how to use ChatGPT in the long term diverged: while some instructors are willing to integrate them into courses, others prefer to ban them and continue teaching programming fundamentals [22]. Becker et al. discussed how learning should be adapted according to ChatGPT. For instance, students' learning experiences may be personalized with more individualized and relevant feedback from Generative AI [3].

As one of the few researchers who conducted a controlled experiment in this realm, Kazemitabaar et al. focused on the potential of OpenAI Codex to enhance the code-authoring abilities of novice programmers. Their experiment, which involved 69 novice participants, showed that Codex could significantly boost performance in code generation without undermining the participants' capability in manual code modification. This emphasizes the dual functionality of Codex: assisting with code generation while preserving the essence of manual coding [17].

However, as the adoption of these AI tools in education grows, so do concerns about the integrity of learning. Issues such as potential plagiarism and finding the right balance between enriching learning experiences and inadvertently fostering an over-reliance on AI-assisted tools are gaining prominence in discussions [17–19, 22].

The myriad benefits and emerging concerns associated with ChatGPT underscore the importance of conducting more in-depth research into its applications in CS education settings. In our study, we aim to contribute to this area, by focusing on ChatGPT's use in learning object-oriented programming. Building upon the work of Kiesler et al., we employ an experimental design to investigate how ChatGPT aids novices in learning object-oriented programming. We also extend the scope of research by examining novices' interactions with ChatGPT and their perspectives towards it, which can provide pedagogical implications for CS instructions and help instructors better leverage ChatGPT in their courses. Building upon the existing literature, our study introduces a novel perspective by examining the specific application of ChatGPT in learning object-oriented programming, an area not extensively covered in previous research. While previous studies have explored the general use of ChatGPT in various educational contexts, our research delves deeper into its role in object-oriented programming education. We not only investigate the effectiveness of ChatGPT in enhancing learning outcomes but also critically analyze its limitations and implications for pedagogy in computer science education.

3 STUDY

In this section, we discuss the experimental protocol, task design, participant recruitment, data collection, and analysis for the presented study. We aim to answer the following questions:

RQ1 How does ChatGPT impact students' learning performance in CS1 education? Do students learn better and complete the tasks more quickly with ChatGPT?

RQ2 What resources do students rely on to solve introductory programming problems? How do they use these resources?

RQ3 How do CS1 students perceive ChatGPT in the context of CS1?

3.1 Classroom Setting

We ran this study in Spring 2023 with students who were taking a 3-credit CS1 course at Blind University, a private research-intensive university in North America. This CS1 course is a Java-based course taken by all CS majors and minors and is open to non-majors. In this course, students learn the fundamentals of programming in Java and the concepts of object-oriented design. Students practice the concepts covered in the lecture in weekly programming assignments (11 programming assignments over the semester). The programming assignments were to be solved individually by each student, thus receiving aid (other than from the TAs), accessing Q&A sites, and generative AI, were prohibited.

In Spring 2023, there were 21 teaching assistants (TAs) in this course, each TA held three to four hours of office hours per week. Students could attend the TA office hours for questions, guidance, and feedback. Students were also encouraged to post questions on Piazza for peer, TA, and instructor support.

This study received an IRB exemption from Blind University.

3.2 Group Assignment & Experiment Protocol

Our study procedure is shown in Figure 1. To ensure a randomized assignment, all students in the CS1 Java-based course were initially semi-randomly divided into two groups based on their midterm scores. Students in the two initial groups have a similar distribution of their midterm scores. Then we designed an extra credit assignment for this study. The extra credit was based on the completion of the study's tasks and a post-survey (see Section 3.3). The study was announced during lectures by the authors of this paper.

We randomly assign the two initial groups to (1) Experimental Group and (2) Control Group. The only distinction between the settings for these two groups was the availability of ChatGPT:

- *Experimental Group*: Participants could use ChatGPT and other online resources (e.g., Google, course slides) while working on the UML diagram and programming.
- *Control Group*: Participants were not allowed to use ChatGPT but could use any other online resources (e.g., Google, course slides) for the UML diagram and programming tasks.

Accordingly, two versions of study instructions were presented to each group - the same tasks but with different constraints on using ChatGPT (ChatGPT-3.5) as stated above. The study instructions included details of the experimental setup. Then all students were self-selected to participate in the study (i.e., completed the study by the deadline, which was 48 hours after the release of the study) to earn surveys in the course. Participants were asked to complete the tasks in this study independently outside of school hours at their own convenience with a recommended duration of two hours at any location of their own choice. Participants were instructed to record their computer screens throughout the process, from UML diagram design to the implementation of the class skeleton, post-evaluation, and finally, the submission of the assignments.

In total, 23 students from the Experimental Group completed the study, while 33 students from the Control Group completed the study. After completing the tasks, participants were asked to fill out a post-survey. This survey gathered demographic details, gauged computer science skills, and collected feedback on participants' perceptions and opinions regarding ChatGPT's influence on CS1 programming.

3.3 Tasks

In this subsection, we discuss the design and metrics of each task.

3.3.1 UML class diagram.

Participants were instructed to design a UML diagram representing the very assessments used in the CS1 course the students were currently enrolled in. This diagram consists of classes that are different evaluative tools, such as TopHat questions, zyBooks activities, programming assignments, exams, and final exams. Participants need to include class names, inheritance relationships, overloaded constructors, fields with designated types, methods with associated parameters, and return types.

Metrics: Participants' UML designs are evaluated against a set of predefined criteria to ensure accuracy and completeness with a total score of 40 points. Key grading points include:

- *UML Syntax*:
  - Fields marked private using the minus sign and methods marked public with the plus sign.
  - Proper notation for fields and methods, ensuring names are followed by their types.
  - Correct inclusion of an inherited field of a derived class.
- *Top-Level Base Class and Derived Classes*:
  - Correct naming and inheritance representation.

- Precise inclusion of specified fields and their types.
- Presence of the toString method.

3.3.2 *Java Programming.*

Following the UML diagram, participants were tasked with defining each class with a class skeleton, which includes the instance variables and the headers of the instance methods. This task tested their proficiency in Java, understanding of core programming concepts, and ability to apply them in real-world scenarios.

Metrics: The grading for this task was based on the correct definition of class skeleton variables within each class and method headers according to the UML diagram. Key grading points include:

- *Class and Inheritance*:
  - Correct inclusion of a top-level base class and other classes to represent zyBooks activities, programming assignments, exams, final exams etc.
  - Correct specification of inheritance (e.g., derived from a base class or top-level base class).
- *Field Inclusion and Type Specification*:
  - Precise inclusion of specified fields with their correct types.
  - Mandatory fields for certain classes (e.g., hasTrueFalse, hasMultipleChoice for ZyBookActivity).

3.3.3 *Post-evaluation.*

Following the UML diagram and coding, participants completed a closed-book post-evaluation. The evaluation was designed to evaluate participants' comprehension and implementation of Object-Oriented Programming and it has four questions:

Q1 (brief answer) Please briefly describe the process you followed to design the hierarchical structure for the Study Assignment and elaborate on how these steps align with the major principles of Object-Oriented Programming?

Q2 (coding) Please implement the toString() method in EACH class given the program description in the Study Assignment (directly add to the existing skeleton Java programs). Use super() to reduce code redundancy if needed.

Q3 (multiple choices) Given the UML diagram, which of the following code snippets is correct? Select ALL that apply.

Q4 (multiple choices) Assume the code snippet is error-free, which of the following UML diagrams correctly represents the class structure? Select ALL that apply.

Metrics: The grading for this task was based on the correct answers to the four questions presented:

Q1 Responses that touched upon OOP cornerstones like Abstraction, Encapsulation, Inheritance, and Polymorphism were graded +3 points.

Q2 Participants were graded on their implementation of the toString() method and the efficient use of the super() function. Proper use was credited for making the code streamlined and non-redundant.

Q3 and Q4 Each right answer got +1 point. However, choosing any wrong options meant no points were given for that question.

3.4 Participants

We initially divided students into two academically balanced groups of the same size - the control and the experimental - based on their mid-term scores. This ensured that both groups had comparable average mid-term scores, with no statistically significant differences. Once the study was introduced as an optional opportunity for survey, students from both pre-assigned groups decided their participation level. Some were fully engaged, while others dropped out or didn't meet all requirements. Consequently, the final tally saw 23 students in the Experimental Group and 33 in the Control Group. The premise of the study remained consistent: the Control Group operated without ChatGPT access, while the Experimental Group had the option, but not an obligation, to utilize ChatGPT. Table 1 shows demographic information of the eligible participants assigned to two groups. One eligible participant did not report demographic information, so they are not included in the analysis of demographic information.

To examine whether the two different groups' student participants' background on the related programming tasks in the experiment might introduce biases, we checked normality using the Shapiro-Wilk test ($p < 0.01$) and thus used Mann-Whitney U-test on the mid-term scores of the two groups and did not find any significant difference ($p = 0.6$). Similarly, we conducted a Mann-Whitney U-test on the Java programming experience of the two groups and did not find any significant difference ($p = 0.38$).

3.5 Data Collection and Processing

We utilized various data sources to assess participants' behaviors and opinions:

Screen recordings: We collected around 65.5 hours of screen recordings from 48 participants (8 out of the 56 participants failed to submit screen recordings due to technical issues or submitted invalid videos): 20 participants from the Experimental Group and 28 from the Control Group. The duration of the recordings spanned a minimum of 12 minutes to a maximum of 154 minutes. The mean duration was approximately 81.83 minutes, with a median value of 80 minutes. Two of the authors manually annotated all the screen recordings together, creating a comprehensive log to capture every participant's actions during the tasks, such as *initiating tasks*, *referencing materials*, *seeking help from ChatGPT, Google, Stack Overflow*, and *specific code-related activities*. All the annotated actions were meticulously timestamped and described.

We also employed an inductive thematic analysis to dissect the questions posed to ChatGPT and other online search engines, utilizing ATLAS.ti. Such questions were finally categorized into 7 categories including *syntax*, *debug*, *concept* and *example*, etc. For both log annotation and thematic analysis, an initial codebook was created by the two authors from examining a subset of the data. Disagreements were resolved through extensive discussions, revisiting original transcripts, and iterative refinement with all authors, leading to the final codebook. A final codebook was developed after evaluating four participant recordings and can be found in the codebook provided in the supplementary package. This process follows the best practice of qualitative analysis [26].

Table 1: Demographic Data of Eligible Participants

Demographic Variables		ChatGPT	NonChatGPT
Gender	Male	11	10
	Female	12	21
	Others	0	1
Race	White or Caucasian	4	14
	Black or African American	0	4
	Asian	12	11
	Native Hawaiian or Other Pacific Islander	0	1
	Other	7	1
	Prefer not to say	0	1
Year in College	First Year	20	25
	Second Year	3	5
	Third Year	0	1
	Fourth Year	0	1

Among the 48 screen recordings, 26 participants recorded the UML diagram creation process (e.g., others opted for a handwritten UML diagram). Forty-five recorded the programming task and recorded the post-evaluation process.

Grading: A total of 56 UML diagrams and corresponding program files, and 55 post-evaluations were collected. The authors graded all three tasks using specified rubrics, which can be found in the supplementary package. The total scores for UML diagram, programming, and post-evaluation were 40, 40, and 8 points, respectively.

Furthermore, with the annotation of screen recordings, we also collected the time students spent on each task when applicable (i.e. when included in the screen recordings).

Post-survey: Fifty-five post-surveys were collected, including participants' demographic information, computer science background, usage, perceptions of ChatGPT, suggestions on how to use ChatGPT, etc. The questions for ChatGPT perception were designed as Likert scale questions ranging from 1 − 7 (1 represents *Strongly Disagree* and 7 represents *Strongly Agree*), as shown in Table 5. Participants also provided suggestions and opinions on the potential use of ChatGPT for CS1 students. Participants' suggestions were thematically coded into 6 categories following a similar qualitative analysis process for screen recording, as shown in Table 6. Their opinions towards chatGPT (i.e., negative, moderate, and positive) were also identified from the suggestions.

4 RESULTS

In this section, we present the results regarding the three proposed research questions. Based on the screen recording analysis and the post-survey results, participants in the Experimental Group may not use ChatGPT. To clarify the presentation of results, we redefined the participant groups as follows:

- ***ChatGPT*** ($n = 11$): Participants who actually used ChatGPT to solve the tasks (i.e., a subset of the Experimental Group).
- ***NonChatGPT*** ($n = 45$): Participants who did not use ChatGPT in the experiment.

The observation that over 50% of students in the Experimental Group (12/23) chose not to use ChatGPT might indicate that ChatGPT may not be considered a top-choice tool among CS1 students yet. Though this not the main purpose of this study, it can inspire future studies investigating CS1 students' acceptance of ChatGPT.

4.1 RQ1: ChatGPT for Learning Outcomes

In this study, we refer to "learning outcomes" as the measurable learning performance (i.e., correctness and time spent) from participants from the UML diagramming, Java programming, and the closed-book post-evaluation test. Therefore, We investigated the impact of using ChatGPT on participants' learning by analyzing the correlations between the use of ChatGPT and their scores of the three tasks and the time allocated to the tasks.

Table 2 shows the average scores and difference when ChatGPT is being used or not. The difference in the scores between the two groups is relatively small and does not survive the significant test.

We also investigated how the use of ChatGPT would affect the speed at which students complete different tasks in this study. Table 3 shows the time in minutes participants spent on UML, programming, and post-evaluation tasks and the difference between the ChatGPT and NonChatGPT groups. It indicates that the ChatGPT group spent less time on the UML task ($Delta = -14.23$) but was similar on the programming task. However, similar to the task scores, none of the time differences survived the significant test.

We further investigated the potential effect of the mid-term exam scores and the use of ChatGPT on participants' learning outcomes also considering the random effect of different course sections. From a multi-level mixed-effect analysis, we found no statistically significant relationship between any of the factors.

Summary: The use of ChatGPT does not demonstrate a statistically significant impact on participants' learning outcomes.

4.2 RQ2: Utilization of Resources

To comprehensively understand how participants utilize and interact with various educational resources, especially in juxtaposition to ChatGPT, we meticulously analyzed the annotated log data derived

Table 2: Participants' average scores of UML, Programming, and Post-evaluation tasks. *Delta* represents the difference between *ChatGPT* and *NonChatGPT* group (*Delta* = *ChatGPT* − *NonChatGPT*). None of the differences survives the significant test.

Task	ChatGPT	NonChatGPT	Delta
UML Diagram	27.45	28.91	-1.46
Java Programming	30.64	29.69	0.95
Post-evaluation	4.73	4.64	0.09

Table 3: Participants' average time spent (minutes) on UML, Programming, and Post-evaluation tasks. *Delta* represents the difference between *ChatGPT* and *NonChatGPT* group (*Delta* = *ChatGPT* − *NonChatGPT*). None of the differences survive the significant test.

Task	ChatGPT	NonChatGPT	Delta
UML Diagram	30.76	44.99	-14.23
Java Programming	54.68	51.82	2.86
Post-evaluation	9.09	8.9	0.19

from participants' screen recordings. Specifically, we compared participants' utilization of resources between the ones with the top learning outcomes and the ones with inferior outcomes.

We discerned a variety of resources participants favored: ChatGPT, Google, lecture slides of the course, zyBooks - an interactive digital learning platform used in the course, programming assignments (PAs) that participants previously completed, and Piazza – the course's online discussion forum. The usage data is illustrated in Table 4. The NonChatGPT group demonstrated more frequent and extended interactions with traditional resources, such as lecture slides, zyBooks, previous PAs, and Piazza, compared to those in the ChatGPT group.

We analyzed the prompts students used with ChatGPT. Out of the students in the Experimental Group, 11 participants actively utilized ChatGPT. From our analysis, we identified two distinct types of prompts:

(1) **Direct Code Generation (DCG)**: In this approach, participants directly requested ChatGPT to generate code based on their prompts. They would then copy and paste this generated code directly into their tasks.

(2) **Conceptual Inquiry (CI)**: Here, participants posed concept-based questions to ChatGPT. They learned from the responses provided by the AI and then proceeded to complete their tasks based on this newfound understanding.

In Figure 2, we illustrate an example of a student's interaction with ChatGPT using the Direct Code Generation approach for a task. In this instance, the student initially asked ChatGPT to assist in generating UML diagrams for every class (i.e., "the information above"). Subsequently, the student requested ChatGPT to produce Java code based on the UML diagram that was drawn.

While nine participants from the NonChatGPT group turned to Google for assistance, only one participant from the ChatGPT group

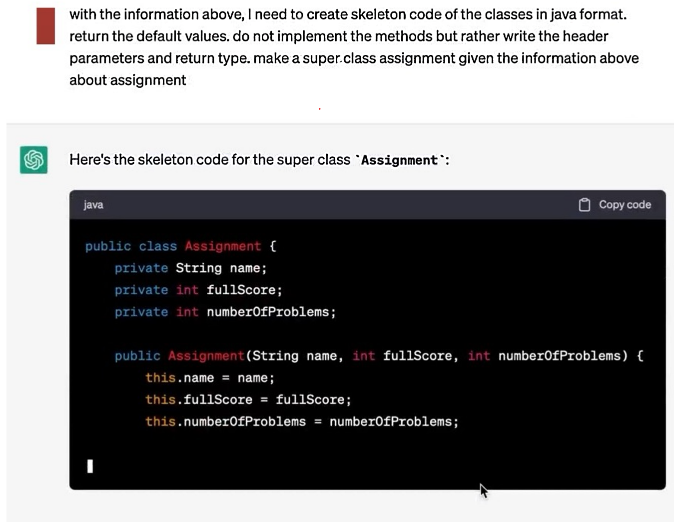

Figure 2: Example of a participant interacting with ChatGPT. The participant first asked ChatGPT to assist in generating UML diagrams for every class (i.e., "the information above"), then the participant requested ChatGPT to produce Java code based on the UML diagram that was drawn

did. Lecture slides were consulted by eight participants in the NonChatGPT group, whereas none from the ChatGPT group referred to them. In terms of zyBooks references, 19 participants from the NonChatGPT group utilized this resource, contrasted with only three from the ChatGPT group. Lastly, ten NonChatGPT participants referred to previous PAs and Piazza, while only one participant from the ChatGPT group consulted previous PAs. ChatGPT group members seemed to rely predominantly on ChatGPT, spending considerably less interest and time exploring other educational CS resources compared to the NonChatGPT group. Figure 2 shows an example of one participant's interaction with ChatGPT.

To better understand if the utilization of ChatGPT affects students' outcomes, we first did a general comparative analysis between all participants whose learning outcomes fall within the top quartile (14 participants) and the bottom quartile (also, 14 participants). Among the ChatGPT group, we further compared participants whose learning outcomes fall within the top quartile (T1 and T2, $n = 2$) with those whose learning outcomes are situated in the bottom quartile (B1–3, $n = 3$)

The total scores of UML and programming tasks of the 12 non-ChatGPT participants in the top quartile ranged from 75.0 to 80.0 and their post-evaluation scores varied between 3.0 and 7.0. Two of them achieved the full score of 80.0.

While T1 and T2 have a total UML and programming score of 78.00, positioning them second within the top quartile. However, their strategies in approaching the task were markedly different. T1 primarily sought conceptual and syntax-related clarifications from ChatGPT and utilized newfound knowledge to refine his coding process and write the code, while T2 exhibited a more pronounced dependence on ChatGPT for code generation. T2 requested ChatGPT to write complete code solutions with detailed directions and

Table 4: Participants' Utilization of Different Resources.

Resources	Count/Total (Percentage)	Usage
ChatGPT	11/20 (55%)	Generate code snippets, seek explanations, draw UML diagrams
Google	10/56 (17.86%)	Search for online responses and get directed to other websites
Lecture Slides	8/56 (14.29%)	Refer to educational content (slides) provided by the course
ZyBooks	22/56 (39.29%)	Refer to interactive animations, practice questions, and concept explanations
Programming Assignments (PAs)	6/56 (10.71%)	Check Java's programming style and format, problem-solving strategies
Piazza	5/56 (8.92%)	Refer to the record of peers' inquiries and answers, and ask instructors

directly copied ChatGPT's output without further evaluation. The varied approaches used by T1 and T2 might indicate the potential impact of individual learning dynamics when engaging with ChatGPT. Following are two example prompts T1 and T2 used in ChatGPT:

> "*T1: Does a Java UML diagram include constructions?*"

> "*T2: "Write the class code based on the following information in Java: " followed by the task descriptions provided in the experiment.*"

T1, T2, and B1–3, hold similar midterm scores, ranging between 89 and 93. Their self-reported Java experience levels were also relatively consistent, found to be 1 or 2.

Eleven participants whose learning outcomes fall within the bottom quartile did not use ChatGPT. The total UML and programming scores for these 11 spanned from 26.0 to 45.0, accompanied by post-evaluation scores from 3.0 to 6.0. B1–3 attained total scores of UML and Java ranging from 39.0 to 43.0. The post-evaluation scores for them ranged from 5.0 to 7.0. They only asked concept-based questions in ChatGPT:

> "*B1: FullScore is a variable found in a separate class. How can I pull fullScore from that separate class into this Java class?*"

> "*B3: For the superclass in java, should instance variables that are also accessed by subclasses be public or private?* "

Summary: Reliance on chatGPT did not guarantee enhanced learning performance outcomes. However, the availability of ChatGPT may largely reduce students' interest of exploring other educational resources.

4.3 RQ3: Perceptions for ChatGPT

To understand students' perceptions of ChatGPT for CS1, we analyzed their responses in the post-survey (see Section 3.5).

Attitudes towards ChatGPT: We investigated participants' attitudes toward ChatGPT with their responses to the Likert scale questions about negative and positive statements for ChatGPT, as shown in Table 5, and answers to the open-ended question about suggestions on how to use ChatGPT for CS1 students.

From the Likert scale results, the majority of students indicate a neutral or positive attitude towards ChatGPT on its assistance in programming and learning in CS1 (i.e., Statements 1–7). 56% of the students hold neutral opinions on ChatGPT's capability of providing high-quality answers for programming problems (Statment 2) and an average of 43% of students hold neutral opinions for all statements. Only 9% of students disagree with Statement 10 which indicates concerns related to ChatGPT on ethical issues. Finally, 45% of students have neither a positive nor negative opinion on ChatGPT to help with education equity. Furthermore, we found no statistically significant difference on the ratings between ChatGPT and NonChatGPT groups, highlighting similar perceptions for ChatGPT among participants.

From participants' free input text, interestingly, only two participants presented a positive attitude toward ChatGPT. For example:

> "*Chatgpt should be used to aid education*"

Most participants (i.e., 50 out of 55) presented neutral attitudes (i.e., support the use of ChatGPT but with obvious concerns). For instances:

> "*I think it should be allowed to be used, but should be equally monitored.*"

> "*It is helpful to use to come up with organization tips for a program. However, you will not improve as a programmer if you can't tackle problems without the aid of ChatGPT.*"

Three out of 55 participants held negative attitudes. For examples:

> "*ChatGPT is mostly useless in college/ CS.*"

> "*I would say not to. Simple because ChatGPT will not give you the right code and sometimes doesn't even do the method.*"

Participants' advice on how to use ChatGPT: Table 6 shows the themes explaining participants' suggestions about how to properly use ChatGPT in CS1. *Debugging* is the most mentioned (16.36%) task that benefits the most from ChatGPT. Getting inspiration when stuck or at the starting point and using ChatGPT to learn are ranked second (9.09%).

"Writing entire code" is the most mentioned (14.55%) limitation of ChatGPT. For instance:

> "*Absolutely use it but beware it can be wrong. Use it for debugging, not writing programs. Its logic is often off because it can be hard to describe to it the problem. Debugging is it best ability..*"

> "*From what I've heard, it's terrible at writing code so far. That will likely change in the future, but right now it's completely unreliable.*"

Summary: Students demonstrate a neutral or slightly positive opinion on ChatGPT's capability of helping CS1 programming, while the majority of students are more concerned about potential ethical issues linked to ChatGPT. Students suggest ChatGPT's capability is not reliable across different programming tasks.

5 DISCUSSIONS AND LIMITATIONS

5.1 Implications

Our study underscores the transformative potential of AI tools like ChatGPT in the realm of CS1 education, while also illuminating potential pitfalls. The burgeoning influence of ChatGPT and similar tools is undeniable, with their promise of instant, contextually

Table 5: Participants' Attitudes toward ChatGPT. The column *Distribution* refers to the distribution of responses of (from left to right) strongly disagree (light gray), disagree or somewhat disagree (gray), neither agree nor disagree (medium gray), agree or somewhat agree (dark gray), or strongly agree (darker gray). The column *Rating* indicates the average rating by all participants for each statement.

	Statement	Distribution	Rating
1	"ChatGPT is very good at programming."		4.0
2	"ChatGPT gives me high-quality answers on programming questions."		3.7
3	"I trust ChatGPT on programming questions "		3.3
4	"ChatGPT is helpful for CS1XXX students to develop programming skills"		4.2
5	"We should allow students in CS1XXX to use ChatGPT in the assignments "		3.9
6	"ChatGPT will only help CS1XXX students to develop programming skills if we use it under guidance"		4.6
7	"ChatGPT will help me become a better programmer"		4.3
8	"I'm worried ChatGPT will make it harder for me to get a job in computer science-related fields. "		4.5
9	"ChatGPT will become part of my study/work/programming routine soon"		3.9
10	"I'm concerned about potential ethics, fairness, security problems coming along with the popularity of ChatGPT "		4.9
11	"I think ChatGPT will help solve the problem of education equity"		3.9

Table 6: Themes of Participants' Advice on How to Properly Use ChatGPT for CS1 Students

Theme	Description
To learn	ChatGPT can help students understand flow of code, learn advanced knowledge.
To practice	ChatGPT can help students practice programming and related tasks.
To debug and optimize code	ChatGPT can help students fix bugs in programs and optimize programs.
To get inspiration	ChatGPT can help inspire students when they are stuck or when they start working on a problem.
Not to write the entire code	Don't use ChatGPT to write the entire program for you because it is usually incorrect.
Not to do the assignment	Don't let ChatGPT do the assignment for you because it can be wrong or hinder your own hard work.

relevant solutions. Yet, students and instructors alike must exercise discernment in how they integrate these tools into the learning process:

For Students: Since the usage of ChatGPT seems to largely reduce students' tendency to use other educational resources, students may need to be alerted and avoid relying only on ChatGPT considering ChatGPT may provide inaccurate information. While AI tools offer quick solutions, students are recommended to use them as a supplement rather than a substitute [7]. ChatGPT can be a go-to for quick doubts, relying on it excessively can hinder the development of problem-solving skills. ChatGPT might not guarantee that a student will get better learning outcomes. However, using ChatGPT with traditional learning resources might provide structured and systematic learning for better foundational understanding. If ChatGPT is allowed in their course, it might be beneficial for students to learn different ways to prompt ChatGPT properly, but students should avoid excessive dependence on the tool. On the other hand, since ChatGPT can provide correct programs, especially with proper prompts, it will likely prevent students from independent thinking and problem-solving practice.

For Instructors: *Role Transition*: Instructors, recognizing the emergence of AI-driven tools, may need to consider transitioning to an extent from content providers to facilitators. This shift acknowledges the growing autonomy and self-direction that tools like ChatGPT offer students, allowing educators to guide rather than dictate the learning process [7, 13]. *Guidelines on ChatGPT Usage*: Instructors may want to provide guidelines on the appropriate use of ChatGPT such as holding discussions to familiarize students with ChatGPT's capabilities and limits, offering sessions on prompt engineering to optimize students' queries, ensuring more accurate and contextually relevant responses from ChatGPT [27, 36]. What is more, instructors should educate students about the risks of overreliance on a single resource as well as encourage a diversified approach to learning. The rapidly evolving landscape of generative AI in education necessitates collaboration across institutions. By sharing experiences, best practices, and challenges, instructors and institutions can co-develop innovative teaching strategies and curricula that truly leverage the potential of AI in education.

There are both advocates and opponents of the use of ChatGPT in CS1 education. Advocates of ChatGPT in CS1 education emphasize its ability to deliver immediate and contextually appropriate solutions. Many students gravitate towards such AI-driven tools, believing they can significantly aid in their academic pursuits. The promise of quick answers and clarifications can be especially appealing in a domain as challenging as CS1, where concepts can sometimes seem abstract and daunting. Conversely, opponents voice concerns about potential pitfalls. They fear that heavy reliance on tools like ChatGPT might lead students to a superficial grasp of foundational concepts, bypassing the deeper understanding crucial to mastering the subject [1]. They worry that the ease of obtaining answers might dissuade students from the critical thinking and problem-solving processes that are essential for deep learning. There's a fear that while students might get solutions, they might miss out on the journey of arriving at these solutions—a journey that in itself is a significant learning experience [16]. Interestingly, our findings offer a nuanced perspective. In the scope of our study, which revolved around a single task, we observed no significant difference in learning outcomes between those who used ChatGPT and those who didn't. This suggests that, at least in the short term, the use of ChatGPT neither particularly advantages nor disadvantages students in terms of learning performance. However, the observed behavior of CS1 students - once using ChatGPT, students' tendency to explore other traditional educational resources is largely reduced (though available) and they tend to rely solely on ChatGPT - can be concerning especially considering the inconsistent quality of ChatGPT's output.

In essence, the future of CS1 education, shaped by the confluence of traditional pedagogies and advanced AI tools, beckons a harmonized approach. It's a future where technology complements rather than competes with human-centric learning, and where students are empowered, informed, and critically engaged.

5.2 Limitations and Threats to Validity

Understanding the limitations of a study is paramount to interpreting its results within an accurate context. While our research provides invaluable insights into the role of ChatGPT in CS1, several potential limitations merit discussion.

One of the primary concerns is the active engagement level of participants with ChatGPT. While we designated groups specifically for ChatGPT usage, not all participants in these groups utilized the tool. Such differential engagement can introduce variability in the data, potentially affecting the robustness of our conclusions. However, in our analysis, we've factored in these differential engagement levels, aiming to present a nuanced understanding of how students perceive and adopt AI tools like ChatGPT.

The distribution disparity between the control and Experimental Groups warrants clarification. Our initial approach involved considering all students as subjects. Their subsequent grouping, based on midterm scores, aimed at ensuring a randomized, unbiased assignment. However, the voluntary nature of the study, combined with its presentation as an survey opportunity, led to the observed uneven distribution of our control and Experimental Groups. This skewness, while acknowledged, reflects genuine student choices, adding a layer of realism to our experimental design.

One potential limitation of our study is its relatively small sample size. While we acknowledge this concern, it's worth noting that research in similar domains with innovative tools have been conducted with comparable sample sizes. For example, Bitzenbauer conducted a pilot study in physics education exploring the use of ChatGPT to foster critical thinking skills at the secondary school level with a sample of 53 participants[5]; Zastudil delved into the perspectives of both students and instructors about the use of generative AI in computing education, involving 18 participants[39]; Leiker investigated the potential of AI-generated synthetic video as educational content in an online setting with 83 adult learners, comparing traditional video methods with AI-generated content[23]. Such studies, including ours, provide preliminary insights and set the stage for more extensive research in the future.

Our study's temporal and task-related constraints are notable. Limited to a few hours and encompassing only three tasks, the results predominantly shed light on short-term interactions with ChatGPT. While this focus was intentional, it does raise questions about the tool's long-term implications. We took steps to ensure the tasks were based on foundational concepts and employed meticulous statistical methods to minimize potential biases.

An inherent challenge in any observational study is the potential for observation bias, where participants might modify their behavior due to the awareness of being observed. To counteract this, we provided students with the flexibility to conduct the study at a time that was most convenient for them and in their preferred locations. This approach aimed to replicate a naturalistic setting as closely as possible, allowing students to interact authentically and without undue pressure.

6 CONCLUSIONS

Generative AI tools, like ChatGPT, have progressively carved out a niche in various academic and professional domains. Their potential has been touted in multiple applications, yet the specific influence of ChatGPT in the sphere of CS education, particularly CS1, hasn't been exhaustively explored. In this study, we meticulously conducted a controlled experiment involving 56 CS1 students, aiming to shed light on ChatGPT's potential influence on their learning outcomes. Intriguingly, our findings did not reveal a pronounced or statistically significant impact of ChatGPT on the student's learning performance. Moreover, there weren't discernible, consistent patterns in ChatGPT usage that could be directly linked to the successful completion or enhancement of introductory programming tasks. An interesting observation, however, was the apparent shift in resource utilization — students who actively engaged with ChatGPT demonstrated a diminished reliance on other traditional educational resources. Students' perceptions of ChatGPT were mixed, oscillating between neutral to mildly positive when it came to aiding their CS1 programming endeavors. Nonetheless, a recurring sentiment was the perceived unpredictability or lack of consistent reliability of ChatGPT across a spectrum of tasks. Furthermore, students exhibited heightened awareness and reservations about the ethical ramifications tied to the pervasive use of ChatGPT. We hope our study can pave the way to more future research to understand ChatGPT's role in potentially reshaping the contours of computer science education. Specifically, as we stand on the brink of a new educational paradigm with AI-driven tools at the forefront, it's imperative to ensure that our classrooms remain diverse, inclusive, and are equipped to provide efficient and transformative learning experiences.

7 ACKNOWLEDGMENTS

We would like to thank all the student participants in our study and the partial support from NSF grant CCF-2211429.

REFERENCES

[1] David Baidoo-Anu and Leticia Owusu Ansah. 2023. Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. *Available at SSRN 4337484* (2023).
[2] Paramarshi Banerjee, Anurag Srivastava, Donald Adjeroh, Y Ramana Reddy, and Nima Karimian. 2023. Understanding ChatGPT: Impact Analysis and Path Forward for Teaching Computer Science and Engineering. (2023).
[3] BRETT A BECKER, MICHELLE CRAIG, PAUL DENNY, HIEKE KEUNING, NATALIE KIESLER, JUHO LEINONEN, ANDREW LUXTON-REILLY, JAMES

PRATHER, and KEITH QUILLE. 2023. Generative AI in Introductory Programming. (2023).

[4] Som Biswas. 2023. Role of ChatGPT in Computer Programming.: ChatGPT in Computer Programming. *Mesopotamian Journal of Computer Science* 2023 (2023), 8–16.

[5] Philipp Bitzenbauer. 2023. ChatGPT in physics education: A pilot study on easy-to-implement activities. *Contemporary Educational Technology* 15, 3 (2023), ep430.

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

[7] Aras Bozkurt. 2023. Generative artificial intelligence (AI) powered conversational educational agents: The inevitable paradigm shift. *Asian Journal of Distance Education* 18, 1 (2023).

[8] Davide Castelvecchi. 2022. Are ChatGPT and AlphaCode going to replace programmers? *Nature* (2022).

[9] E Chen, R Huang, HS Chen, YH Tseng, and LY Li. 2023. GPTutor: A ChatGPT-powered programming tool for code explanation. arXiv. *arXiv preprint arXiv:2305.01863* (2023).

[10] Debby RE Cotton, Peter A Cotton, and J Reuben Shipway. 2023. Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in Education and Teaching International* (2023), 1–12.

[11] Ramon Dijkstra, Zülküf Genç, Subhradeep Kayal, Jaap Kamps, et al. 2022. Reading Comprehension Quiz Generation using Generative Pre-trained Transformers.

[12] Yunhe Feng, Sreecharan Vanam, Manasa Cherukupally, Weijian Zheng, Meikang Qiu, and Haihua Chen. 2023. Investigating Code Generation Performance of ChatGPT with Crowdsourcing Social Data. In *Proceedings of the 47th IEEE Computer Software and Applications Conference*. 1–10.

[13] Mehmet Firat. 2023. What ChatGPT means for universities: Perceptions of scholars and students. *Journal of Applied Learning and Teaching* 6, 1 (2023).

[14] Mohammad Fraiwan and Natheer Khasawneh. 2023. A Review of ChatGPT Applications in Education, Marketing, Software Engineering, and Healthcare: Benefits, Drawbacks, and Research Directions. *arXiv preprint arXiv:2305.00237* (2023).

[15] Sajed Jalil, Suzzana Rafi, Thomas D LaToza, Kevin Moran, and Wing Lam. 2023. Chatgpt and software testing education: Promises & perils. In *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 4130–4137.

[16] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences* 103 (2023), 102274.

[17] Majeed Kazemitabaar, Justin Chow, Carl Ka To Ma, Barbara J Ericson, David Weintrop, and Tovi Grossman. 2023. Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–23.

[18] Natalie Kiesler, Dominic Lohr, and Hieke Keuning. 2023. Exploring the Potential of Large Language Models to Generate Formative Programming Feedback. *arXiv preprint arXiv:2309.00029* (2023).

[19] Natalie Kiesler and Daniel Schiffner. 2023. Large Language Models in Introductory Programming Education: ChatGPT's Performance and Implications for Assessments. *arXiv preprint arXiv:2308.08572* (2023).

[20] Lucas Kohnke, Benjamin Luke Moorhouse, and Di Zou. 2023. ChatGPT for language teaching and learning. *RELC Journal* (2023), 00336882231162868.

[21] Bass Dina Lanxon, Nate and Jackie Davalos. 2023. A Cheat Sheet to AI Buzzwords and Their Meanings. *The Washington Post* (Mar 2023). https://www.washingtonpost.com/business/2023/03/10/glossary-of-ai-terms-meaning-of-gpt-openai-machine-learning-chatbots/3c8a59d8-bf44-11ed-9350-7c5fccd598ad_story.html

[22] Sam Lau and Philip Guo. 2023. From" Ban It Till We Understand It" to" Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot. In *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1*. 106–121.

[23] Daniel Leiker, Ashley Ricker Gyllen, Ismail Eldesouky, and Mutlu Cukurova. 2023. Generative AI for learning: Investigating the potential of synthetic learning videos. *arXiv preprint arXiv:2304.03784* (2023).

[24] Chung Kwan Lo. 2023. What is the impact of ChatGPT on education? A rapid review of the literature. *Education Sciences* 13, 4 (2023), 410.

[25] Stephen MacNeil, Andrew Tran, Dan Mogil, Seth Bernstein, Erin Ross, and Ziheng Huang. 2022. Generating diverse code explanations using the gpt-3 large language model. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 2*. 37–39.

[26] Matthew B Miles, A Michael Huberman, and Johnny Saldaña. 2014. Qualitative data analysis: A methods sourcebook. 3rd.

[27] Michael Neumann, Maria Rauschenberger, and Eva-Maria Schön. 2023. "We Need To Talk About ChatGPT": The Future of AI and Higher Education. (2023).

[28] Emmanuel Opara, Adalikwu Mfon-Ette Theresa, and Tolorunleke Caroline Aduke. 2023. ChatGPT for teaching, learning and research: Prospects and challenges. *Opara Emmanuel Chinonso, Adalikwu Mfon-Ette Theresa, Tolorunleke Caroline Aduke (2023). ChatGPT for Teaching, Learning and Research: Prospects and Challenges. Glob Acad J Humanit Soc Sci* 5 (2023).

[29] Zachary A Pardos and Shreya Bhandari. 2023. Learning gain differences between ChatGPT and human tutor generated algebra hints. *arXiv preprint arXiv:2302.06871* (2023).

[30] Basit Qureshi. 2023. 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).

[31] Saumyaranjan Sahoo, Satish Kumar, Mohammad Zoynul Abedin, Weng Marc Lim, and Suresh Kumar Jakhar. 2023. Deep learning applications in manufacturing operations: a review of trends and ways forward. *Journal of Enterprise Information Management* 36, 1 (2023), 221–251.

[32] Malik Sallam. 2023. ChatGPT utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. In *Healthcare*, Vol. 11. MDPI, 887.

[33] Nigar M Shafiq Surameery and Mohammed Y Shakor. 2023. Use chat gpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290* 3, 01 (2023), 17–22.

[34] Ahmed Tlili, Boulus Shehata, Michael Agyemang Adarkwah, Aras Bozkurt, Daniel T Hickey, Ronghuai Huang, and Brighter Agyemang. 2023. What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education. *Smart Learning Environments* 10, 1 (2023), 15.

[35] Xingzhi Wang, Nabil Anwer, Yun Dai, and Ang Liu. 2023. ChatGPT for design, manufacturing, and education. *Procedia CIRP* 119 (2023), 7–14.

[36] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382* (2023).

[37] Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023. A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica* 10, 5 (2023), 1122–1136.

[38] Da Yan. 2023. Impact of ChatGPT on learners in a L2 writing practicum: An exploratory investigation. *Education and Information Technologies* (2023), 1–25.

[39] Cynthia Zastudil, Magdalena Rogalska, Christine Kapp, Jennifer Vaughn, and Stephen MacNeil. 2023. Generative AI in Computing Education: Perspectives of Students and Instructors. *arXiv preprint arXiv:2308.04309* (2023).