mathematics

MDPI

Article

Computer Science Education in ChatGPT Era: Experiences from an Experiment in a Programming Course for Novice Programmers

Tomaž Kosar [1], Dragana Ostojić [1], Yu David Liu [2] and Marjan Mernik [1,*]

[1] Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška cesta 46, 2000 Maribor, Slovenia; tomaz.kosar@um.si (T.K.); dragana.ostojic@um.si (D.O.)
[2] Department of Computer Science, State University of New York at Binghamton (SUNY), 4400 Vestal Parkway East, Binghamton, NY 13902, USA; davidl@binghamton.edu
* Correspondence: marjan.mernik@um.si

Abstract: The use of large language models with chatbots like ChatGPT has become increasingly popular among students, especially in Computer Science education. However, significant debates exist in the education community on the role of ChatGPT in learning. Therefore, it is critical to understand the potential impact of ChatGPT on the learning, engagement, and overall success of students in classrooms. In this empirical study, we report on a controlled experiment with 182 participants in a first-year undergraduate course on object-oriented programming. Our differential study divided students into two groups, one using ChatGPT and the other not using it for practical programming assignments. The study results showed that the students' performance is not influenced by ChatGPT usage (no statistical significance between groups with a p-value of 0.730), nor are the grading results of practical assignments (p-value 0.760) and midterm exams (p-value 0.856). Our findings from the controlled experiment suggest that it is safe for novice programmers to use ChatGPT if specific measures and adjustments are adopted in the education process.

Keywords: large language models; ChatGPT; artificial intelligence; controlled experiment; object-oriented programming; software engineering education

MSC: 97P10

Citation: Kosar, T.; Ostojić, D.; Liu, Y.D.; Mernik, M. Computer Science Education in ChatGPT Era: Experiences from an Experiment in a Programming Course for Novice Programmers. *Mathematics* **2024**, *12*, 629. https://doi.org/10.3390/math12050629

Academic Editor: Chengjie Sun

Received: 19 January 2024
Revised: 16 February 2024
Accepted: 18 February 2024
Published: 21 February 2024

1. Introduction

In recent years, integrating new technologies such as online learning platforms, mobile devices, and virtual learning environments has revolutionized how educators deliver content and engage with students. These innovations have made education more accessible, personalized, and interactive. As a result, students can explore subjects in more depth and at their own pace. While we remain in the process of understanding and applying these technologies, a new one is already on the horizon, in the form of generative artificial intelligence (AI) [1].

The usage of large language models (LLMs) [2] has grown exponentially in recent years. One such model, ChatGPT [3], has garnered significant attention in the public since its launch in November 2022. ChatGPT is a chatbot developed by OpenAI [4] and enables users to have human-like conversations. ChatGPT can answer questions and assist with tasks like composing emails, essays, and even programming code [5,6]. On one hand, the generated text is plausible, making it a powerful tool. On the other hand, ChatGPT can be misused, e.g., students may cheat on their essays.

In the Computer Science education community, there is significant debate over using ChatGPT-generated code in classrooms. On the positive side, the benefits of using ChatGPT in education are well argued for [7]. For example, ChatGPT may provide students with a more

interactive and engaging learning experience, and increase their interest and motivation [8,9]. Computer Science students can ask questions about programming code and receive immediate answers, making learning programming more efficient [10]. Additionally, ChatGPT can generate several different examples to explain complex programming concepts [7].

On the flip side, some believe it is risky [11,12]. The LLM is limited by the knowledge it was trained on [7], giving a possibility of answering complex questions inaccurately [13]. Furthermore, code debugging and interpretation require a deep understanding of the code under consideration. The educator can provide a step-by-step explanation of the code, while current LLMs are still limited in this respect as shown in [14]. Another serious drawback of using ChatGPT is that it could discourage students from developing skills, e.g., reasoning [15]. If students rely excessively on ChatGPT to provide programming code, they may not develop the required skills to solve problems on their own. Excessive use and cheating are some Computer Science educators' major concerns regarding ChatGPT usage, especially for novice programmers (first-year students). In a nutshell, students cannot develop important skills, such as critical thinking, creativity, decision-making [16], and one of the essential capabilities for software developers, problem solving [17]. Some universities even decided to take measures in blocking access to the ChatGPT website on school grounds [18].

However, LLM technologies are probably here to stay. We believe that, rather than avoiding these technologies, we need to embrace LLMs and modernize education [19]. To understand how LLMs and ChatGPT influence the learning process [16], there is a need for experimental studies [10,20–23]. We have to test common beliefs empirically and rigorously, such as the belief that students will use LLMs without hesitation for plagiarism [24], or thinking that using LLMs will affect their critical thinking and problem-solving skills negatively [16]. In this paper, we report our experience in ChatGPT-assisted learning in Programming II, a course in the second semester of the first year of the Computer Science and Information Technologies undergraduate program at the University of Maribor, Slovenia. Our experiment was motivated by the following questions:

- Does the use of ChatGPT affect performance on practical assignments and midterm exam results?
- Does the use of ChatGPT affect the overall student performance in the introductory programming course?
- What impact does ChatGPT usage have on the course final grade?
- For what purpose did students use ChatGPT during the course on Programming II?
- Is ChatGPT useful for learning programming at all, according to students' opinions?

In this context, we performed a controlled experiment [25] using ChatGPT for practical assignments in the first-year undergraduate study of Computer Science. We formed two groups, one using ChatGPT and the other not using it for practical assignments. Several adjustments were made for the execution of this year's introductory course on object-oriented programming.

Our results from the controlled experiment show that overall performance in the course was not influenced by ChatGPT usage or the results on practical assignments or midterm exams. We believe a main contributor leading to this conclusion is the adjustments we have made to the course during (1) constructing assignments, (2) defending assignments, and (3) midterm exams. Those actions encouraged participants not to rely solely on the use of ChatGPT. For example, all our assignments were designed carefully to minimize the chance of ChatGPT answering the questions directly. As another highlight, we introduced an evaluation process, where assignment grading was not based solely on the code submitted to the original assignment questions; instead, grades were given in the lab session based on an extended version of the assignment, through an interactive defense process involving the students and the teaching assistants. Overall, we believe *ChatGPT should be incorporated into future education, and it must be embraced with adjustments in course evaluation to promote learning*.

The paper is divided into sections, presenting a different part of this experiment. Section 2 discusses the background on ChatGPT, Section 3 describes related work, and Section 4 the

experiment design. Section 5 presents the results and data analysis. Section 6 discusses the threats to the validity of our controlled experiment, and, lastly, Section 7 summarizes our key findings from our empirical study.

2. Background

LLMs [2] represent a transformative technology in the field of natural language processing (NLP) [26], bringing a new linguistic capability and opportunities for diverse applications. These models are designed with vast neural architectures, and supported by extensive training data [27]. LLMs empower applications to understand, generate, and manipulate human languages in ways that were previously impossible. The main feature of LLMs is that they generate text similar to human speech. One of the most well-known LLMs is Generative Pre-trained Transformer (GPT-3) [28] based on the transformer architecture [29] that improved NLP significantly.

Chatbots [30] are computer programs designed to simulate conversations in text (or voice [31]) over the Internet. They are programmed to understand natural languages and respond to user questions in a way that imitates human-to-human conversations. Chatbots can be categorized into two main types: rule-based and machine learning (sometimes referred to as AI-powered) chatbots [30]. Rule-based chatbots operate on predefined rules. They follow instructions and can provide responses based on specific keywords or phrases. Rule-based chatbots are limited in their capabilities, and may struggle with complex questions. On the other hand, AI-powered chatbots use advanced technologies, such as LLMs. They are capable of understanding context and learning from interactions and responses. Both types are often used in applications such as customer support, healthcare, information retrieval assistance, virtual assistance, education, marketing, etc.

Chatbots, empowered by LLMs, represent a significant milestone in the evolution of human–computer interaction. These intelligent agents have gone beyond traditional chatbots to engage users in natural, context-aware conversations. LLM-powered chatbots have an understanding of linguistic variations, making interactions feel more human-like and personalized. One such system is ChatGPT [3]. ChatGPT is the most popular chatbot supported by the LLM GPT-3, developed by OpenAI [4] and available publicly. It is proficient in mimicking human-like communication with the users. GPT-3 models are trained on extensive text data (approximately 175 billion trainable parameters and 570 GB of text [32]). During our experiment (from February till June 2023), we used ChatGPT with GPT-3.5, although GPT-4 was already available (March 2023) but not for free usage.

Prompts [33] refer to the input provided to the chatbot to generate responses. Prompts are the human instructions or questions users provide while interacting with the chatbot. There are different types of prompts: text-based, voice-based, task-driven, informational, conversational, and programming prompts. In the latter, programmers can send specific programming prompts, including code snippets, and chatbots can respond with a context using this input. Hence, programmers (and other users) can modify and fine-tune the prompt through a process called prompt engineering, which instructs the LMMs better to provide more accurate and complex solutions. In this process, programmers can use prompt patterns [34], which are similar to software patterns, reusable prompts to solve common problems in LLM interaction. One such prompt pattern is the domain-specific language (DSL) [35,36] creation pattern.

3. Related Work

The recent popularity of ChatGPT has brought much attention to its benefits (e.g., AI pair programming) or drawbacks (e.g., cheating) in different fields, as well as what impact that chatbot has on higher education [12] in general. Studies on its capabilities and limitations emerged almost as soon as the ChatGPT public release [37]. Our study contributes to this field, and we summarize these studies in this section.

One of the most closely related empirical studies involving ChatGPT in learning programming is reported by [21]. Similar to our study, theirs involved undergraduate students

taking the object-oriented programming course, over a smaller number of participants (41) but more experienced programmers, second-year students. Their participants solved practical assignments in different programming languages (Python, Java, and C++), while, in our study, practical assignments were only in C++. The fundamental difference lies in the design: ours is a *between-subjects* (i.e., *differential*) study, while Yilmaz and Yilmaz is a *within-subjects* study: *all* participants in their study used ChatGPT to solve tasks, and then expressed their opinions in a survey with open questions. In contrast, our study consisted of two groups (ChatGPT and no ChatGPT), and compared the results between these two groups of participants. The participants in their study stated that the most significant advantage of ChatGPT is its time-saving factor, as they obtained reasonably accurate answers quickly, and thus saved time searching for answers. They also stated that it helped debug and solve complex problems and can be available 24/7. Although some participants have expressed that ChatGPT has no disadvantages, some pointed out the problem that its use can lead to laziness, weakened thinking skills, and occupational anxiety. ChatGPT may also produce the wrong answer. Overall, the participants had a positive perception, viewing it as beneficial for solving complex problems and learning unfamiliar topics, and as a helpful tool. The feedback study from our controlled experiment confirms the last findings of their work. Ref. [10] reports another related study with undergraduate students. Similarly to our controlled experiment, this paper reports on a between-subjects study with two groups, one utilizing ChatGPT and the other having access only to textbooks and notes without internet usage. While in [10], experimenters gave participants programming challenges *after finishing the course*, our participants worked on their assignments with adjustments *during the course*, during the semester, partially at home, and in the classroom within lab sessions. They did not report on any adjustments in assignments. In regard to findings, Ref. [10] concluded that the ChatGPT group achieved higher scores in less time while attempting tasks with defined problems, inputs and outputs, and constraints. The assessment was based on the number of test cases successfully passed. The results from our controlled experiment did not confirm these findings. Study [10] also reported that participants faced challenges handling more complex tasks and could not solve some problems entirely. They were also more inaccurate and inconsistent with the submitted code.

Another new study [22] treated *ChatGPT as a student*, and tested whether it could complete an introductory functional language programming course. It turned out that ChatGPT achieved an overall grade of 67% (B−), thus ranking 155 out of 314 students. The study was conducted in two ways, unassisted and assisted, as a student would be in a natural process. They assisted ChatGPT using four prompt engineering techniques: paraphrasing the problem, providing hints, teaching by example, and giving test cases. ChatGPT solved 16 out of 31 tasks with a 100% success rate without additional help. However, when errors did occur, they were of one type—compilation or logical errors, with syntax errors being less common. With assistance, the results from the ChatGPT "student" improved from a rank of 220 to 155. We share a similar experience, where for our practical assignments, ChatGPT scored 24 points out of 44 without additional prompts; with proper prompt engineering, the result would be around 34 points. In their study, the authors reported ChatGPT mostly had problems understanding type specifications, inferring the type of expression, and working with larger programming tasks. Because of these results, students who might use just the code provided from ChatGPT must undergo a defense of practical assignments. We followed the practice of defense consistently during our controlled experiment.

To investigate the impact of ChatGPT on Computer Engineering students, Ref. [23] conducted a controlled experiment in the Embedded Systems course. Their main goal was to check how far ChatGPT could help students answer quiz questions without learning the related topics. Afterwards, these results were compared to those of the previous generation of students answering the same questions after learning those topics. In our experiment, tasks were exclusively programming tasks; Shoufan used theoretical questions as well—true/false questions—while our tasks were connected with writing complete code,

code completion (given code), and code analysis (given inputs/outputs, etc.). Their study also differed from ours in topics (embedded systems vs. object-oriented programming), participant experience (senior vs. novice), and experience duration (four quizzes vs. a whole course). The findings from [23] concluded that the ChatGPT group performed better answering code analysis and theoretical questions but faced problems with code completion and questions that involved images. In writing complete code, the results were inconsistent. The author concluded that the usage of ChatGPT is currently insufficient in Computer Engineering programs, and learning related topics is still essential.

An interesting empirical study from Mathematics [19] explored the potential impact of ChatGPT on their students. The study also focused on essential skills for Computer Science students—how the use of ChatGPT can affect critical thinking, problem-solving, and group work skills. The opinions of participants after assignments on these three skills included a five-point Likert scale (from one "no affect") to five ("it will affect a lot")). The average results on critical thinking (2.38), problem-solving (2.39), and group work (2.97) indicate that participants perceive ChatGPT as having a small-to-moderate effect on the acquisition of the skills as mentioned earlier. Group work appears to be the most affected skill. It would be interesting to see the same results for Computer Science students. It might be an exciting set of feedback questions for the replication study [38]. Instead of conducting a feedback study, assessment instruments can validate students' problem-solving skills [39] in, for example, object-oriented programming (OOP). The research findings of this study conclude that the integration of ChatGPT into education introduces new challenges, necessitating the adjustment of teaching strategies and methodologies to develop critical skills among engineers [19]. We followed the advice and adjusted practical assignments in our course, Programming II.

4. Experiment Design and Goals

Our controlled experiment aimed to compare participants' results during the semester. Particularly, we wanted to verify if ChatGPT usage during the semester influenced final grades, midterm exams, and lab work results. The study design and goals are presented in detail in this section.

4.1. Controlled Experiment and Participants

We prepared a controlled experiment that was part of the Programming II course at the University of Maribor, Faculty of Electrical Engineering and Computer Science (FERI), taught by the fourth author; the teaching assistants were the first and second authors, while the third author was a Fulbright researcher visiting the University of Maribor at that time. The topics covered in the course are listed in Table 1. In our undergraduate program, the course Programming II is the first on object-oriented programming. We teach students basic object-oriented topics: we start with class definition, instance variables, methods, associations, inheritance, etc. Since we use C++, we end our class with new C++ features. The participants in our study were first-year undergraduate students in the Computer Science program, one (out of two) major undergraduate Computer Science program in Slovenia, attracting the best students from the country. We started with 198 participants in the study, but eliminated a small number of participants for several reasons. Beyond non-technical reasons, we also excluded those who did not complete any practical assignments or did not take midterm exams. Finally, our study contains the results of 182 students.

The empirical study was a between-subjects study. We met with all the students at the beginning of the semester to discuss our concerns about ChatGPT and our proposed experiment. After the students accepted it unanimously, we proceeded by dividing them into two groups, with 99 participants each. The division was random and performed by technical staff, and participants were advised to keep this information private from the lecturer and teaching assistants. We did not want this information to affect practical assignments, defenses, and midterm exams. In the rest of the paper, we refer to a treatment group, Group I, which was encouraged to use ChatGPT, and a control group, Group II,

which was asked not to use ChatGPT. An additional measure we agreed upon was to remove students from Group II who reported using ChatGPT for practical assignments in the feedback questionnaire, as this would compromise the results of this group.

Table 1. Topics covered in Programming II together with lab work assignments.

Week	Topic	Practical Assignments	
		Mandatory	**Optional**
1	Programming I repetition	Fuel Consumption	Disarium number
2	Basic classes	Exercise	Fuel Log
3	Class variables and methods	Time	Text Utility
4	Aggregation and composition	Exercise Tracker	Mail Box
5	Inheritance	Strength Exercise	Bank
6		*Midterm exam I*	
7	Abstract class	Graph	Graphic Layout
8	Template function	Vector Util	Vector Util
9	Template class	Linear Queue	Linked List
10		*Additional help*	
11	Operator Overloading	Smart Pointer	Smart Pointer
12	C++11 and C++14	Exercise Tracker	Printer
13	Exceptions, File streams	Sensor Hub	Log
14		*Final practical assignments' defense*	
15		*Midterm exam II*	

4.2. Practical Assignments

Weekly assignments were connected with the topics presented in the lecture (see Table 1, column Topics). The participants received a description of an assignment (a few lines of text) after a lecture, and they had to work solely at home on code until the next lab session, which took place at the faculty.

The practical part of the course Programming II consisted of 22 assignments, 11 mandatory and 11 optional (column Practical Assignments in Table 1). Each week, participants received one mandatory and one optional assignment. Both were worth an equal (two) points. Note that the semester lasts for 15 weeks. However, participants did not receive practical assignments during midterm exams. The defense of practical assignments was on the next lab session (after the lecture). Note that we construct new mandatory and optional assignments every year to prevent assignment solutions from being transmitted from the older generation to the new generation of students.

From Table 1, we can observe that the problems were different each week; also, problems were varied for mandatory and optional assignments. We believe providing students with different problems influences their understanding of object-oriented programming positively.

Figure 1 shows a typical example of an assignment (the practical assignments can be found on the project homepage: https://github.com/tomazkosar/DifferentialStudyChatGPT, accessed on 12 January 2024). A short description was given of what code participants needed to provide, together with a UML diagram for further details. Note that both groups were given identical assignments (those in the Mandatory and Optional columns from Table 1).

- Take task 4.1.
- Modify the `ExerciseTracker` class so that instance variable `exercises` is of type `vector<Exercise*>` (composition).
- Add a `StrengthExercise` class that inherits from the `Exercise` class.
- The main function should have an `ExerciseTracker` object and add to `exercises`:
  - 3 examples of `Exercise` and
  - 2 examples of `StrengthExercise`.

Some guidance on solving the problem:

- Beware of using `protected`, `virtual`, and `override`.

When solving the task, take into account all the knowledge acquired so far (use of the initialization list, constant methods, write down the get/set methods where you need them, etc.).

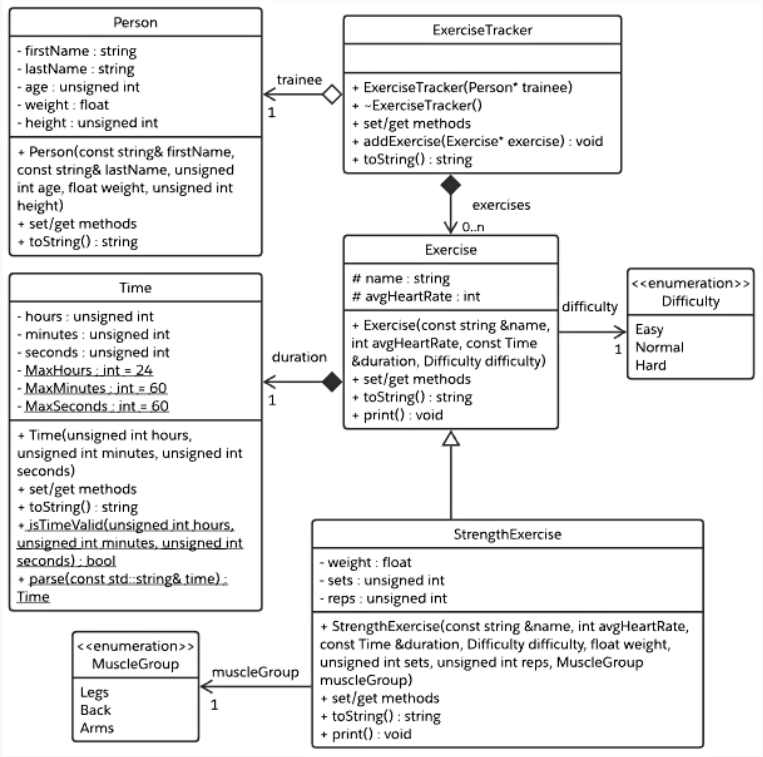

Figure 1. Example of practical assignment (mandatory in week 5).

4.3. ChatGPT-Oriented Adjustments

To prepare for our experiment, we made a number of important adjustments to the course evaluation. In retrospect, these measures likely played an important role in helping us understand the best practices in teaching in the ChatGPT era.

4.3.1. Question Preparation

Before the start of the semester, we analyzed the usage of ChatGPT to answer questions that we had asked Programming II students in prior years. We discovered that ChatGPT excelled at providing solutions to practical assignments with detailed descriptions in the text, a phenomenon also confirmed by recent research [40]. As a result, we decided to provide a number of adjustments, detailed in Table 2. The columns of this table are:

- Type
  For instance, an "extension" assignment means participants had to extend one of the previous practical assignments.
- Code provided
  Some assignments were constructed in a way that participants had to incorporate the given code in their applications.

- Description
  Some assignments were provided with minimal text. Supplemental information was given in the UML diagram.
- Input/output
  This means the student receives the input of their program or the exact output of the program, and they need to follow these instructions.
- Main
  For some assignments, participants receive the main program and the assignment description.

Table 2. Mandatory assignments with explanation.

Problem	Type	Code Provided	Description	Input/Output	Main
Fuel Consumption	new	yes	text	no	yes
Exercise	new	no	text	no	no
Time	new	no	text	no	no
Exercise Tracker	extension	yes	text	no	no
Strength Exercise	extension	no	UML	no	no
Graph	new	yes	UML	yes	yes
Vector Util	new	yes	text	yes	yes
Linear Queue	new	no	text	no	no
Smart Pointer	new	yes	text	no	no
Exercise Tracker	extension	no	text	no	yes
Sensor Hub	new	no	text + UML	no	no

In individual practical assignments, we incorporated one or more adjustments as shown in Table 2. It served as our guideline before constructing practical assignments. Our intention was to prevent ChatGPT from providing direct answers, which would have encouraged Group I students into relying blindly on ChatGPT. We used figures where possible (UML, I/O program, main program, etc.). The deficiency of ChatGPT with non-text-based prompts was known previously [40]. However, we also found that almost all our assignments could be answered by ChatGPT after several rounds of follow-up prompts provided by experienced programmers. On the other hand, novice programmers often have problems constructing the most effective prompts because of a lack of knowledge, the context of the problem, etc.

4.3.2. Extended Assignment

The practical assignment we initially gave at the end of each lecture was incomplete. At the beginning of the following lab session, participants would be asked to work on an extension problem connected to the original assignment. During this session, participants were not allowed to receive assistance from ChatGPT or similar means (such as social media). Nonetheless, resorting to lecture notes, Internet, and application library documentation with examples was permitted and encouraged. Ultimately, students were asked to defend their code developed for the extended assignment, which we detail next. Overall, we found that some participants struggled for the whole lab session (3 h), while the others finished in 15 min. Usually, extensions were small, and the best participants could defend their assignments early in lab sessions. Please refer to Figure 2 for more details on the extended assignments.

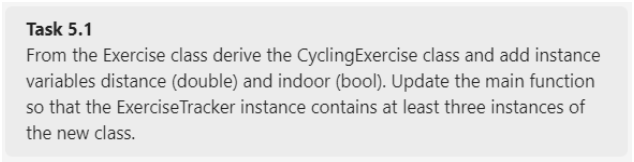

Figure 2. Example of extended assignment.

4.3.3. Assignment Defense

The lab session of Programming II comes with a rigorous and interactive defense procedure. To each student, teaching assistants ask several basic questions regarding topics connected with the last lecture and practical assignment. This year, we made an additional effort in the defense process of the practical assignments. Plagiarism between students was a problem before ChatGPT. With ChatGPT, the defense needs to be even more detailed. Our defense process consists of the following simple questions and tasks for students:

- Conceptual questions
  Typically, we asked participants to explain part of their programming code with an emphasis on object-oriented concepts.
- Code analysis
  Usually, we asked participants to search in code for specific functionality.
- Code changes questions
  Minimal change in the object-oriented part of the programs that change the code's behavior or improve the structure of the code.
- Code completion questions
  Demonstration of using object-oriented code in the main program.

Defense is a time-consuming process. In our opinion, however, it is also essential for developing different programming skills (e.g., code refactoring) and general skills (e.g., critical thinking), particularly in the ChatGPT era.

4.3.4. Paper-Based Midterms

Beyond ChatGPT, there are other options for LLMs, such as CoPilot [41–43]. To evaluate whether students have obtained the knowledge and skills related to Programming II fairly, we decided to use paper-based midterm exams for all participants. Neither group was using computers or IDEs. With that, we had a fair comparison of the results between those two groups.

4.4. Procedure and Data Collection Instrument

The experiment consisted of a background questionnaire at the beginning of the semester, weekly assignments (lab work) with a weekly feedback questionnaire, two midterm exams, and a final feedback questionnaire at the end of the semester.

The background questionnaire aimed to obtain demographic data from participants (age, gender, etc.) and measure their prior experience with the programming language C++, ChatGPT, and their interests in programming, artificial intelligence, etc. The latter questions were constructed using a five-point Likert scale [44] with 1 representing the lowest value and 5 representing the highest value. Altogether, there were ten questions. In Section 5, we only show a subset of questions from the background questionnaires most relevant to our experiment.

Week assignments (discussed extensively in previous subsections) were associated with weekly feedback. The feedback questionnaires were given to participants, measuring the participants' perspectives on assignment complexity and usage of ChatGPT. In particular, the latter aimed to address our concern about participation, ensuring that they still followed our division of two groups, one with ChatGPT support and the other without ChatGPT support.

Instead of theoretical questions, the questions in the midterm exams are closer to practical assignments, covering most of the topics taught in the first half of the semester (first midterm exam) and the second half of the semester (second midterm exam). Each midterm exam consisted of a programming question. Usually, the final result is an object-oriented structure of a given problem and a main program using that structure. Both exams were paper-based.

The feedback questionnaire measured the participants' perspectives on the experiment in the Programming II course. The thirteen questions can be divided into two categories: course and experiment feedback. First, the participants indicated how well they compre-

hended the assignments in Programming II. The second part of the questionnaire focused on ChatGPT (consistency of usage/non usage, purpose of use, etc.). In this paper, we report on a subset of statistics from the feedback questionnaire most relevant to understanding the main study's results. The complete set of questions and answers of our background and feedback questionnaires are available at https://github.com/tomazkosar/DifferentialStudyChatGPT (accessed on 12 January 2024).

4.5. Hypotheses

Our experiment was aimed at confirming/unconfirming three hypotheses: one on midterm exams, one on lab work, and one on overall results. This leads to six possibilities:

- $H1_{\text{null}}$ There is no significant difference in the score of the participants' lab work when using ChatGPT vs. those without ChatGPT.
- $H1_{\text{alt}}$ There is a significant difference in the score of the participants' lab work when using ChatGPT vs. those without ChatGPT.
- $H2_{\text{null}}$ There is no significant difference in the results of the participants' midterm exams when using ChatGPT vs. those without using ChatGPT for lab work.
- $H2_{\text{alt}}$ There is a significant difference in the results of the participants' midterm exams when using ChatGPT vs. those without using ChatGPT for lab work.
- $H3_{\text{null}}$ There is no significant difference in the final grade of the participants when using ChatGPT vs. those without ChatGPT for lab work.
- $H3_{\text{alt}}$ There is a significant difference in the final grade of the participants when using ChatGPT vs. those without ChatGPT for lab work.

These hypotheses were tested statistically, and the results are presented in the next section.

5. Results

This section compares the participants' performance in Programming II in a ChatGPT treatment group (Group I) vs. a control group without ChatGPT (Group II). To understand the outcome of our controlled experiment, this section also presents a study on the background and feedback questionnaires. Hence, the results of the feedback study affected a number of participants in the groups. As explained in the feedback subsection, we eliminated eight students from Group II due to the usage of ChatGPT. The inclusion would have affected the results and represented a threat to the validity of our study.

All the observations were tested statistically with $\alpha = 0.05$ as a threshold for judging significance [45]. The Shapiro–Wilk test of normal distribution was performed for all the data. If the data were not normally distributed, we performed a non-parametric Mann–Whitney test for two independent samples. We performed the parametric Independent Sample *t*-test to check if the data were normally distributed.

5.1. Participant Background

The background questionnaire measured the participants' demographics, prior experiences, and interests. The students' average age was 19.5 years. Regarding gender, 85.9% defined themselves as men, 12.4% female, and 1.7% preferred not to say.

In this paper, we only show a comparison of the participants' opinions about knowledge of ChatGPT. We used a five-point Likert scale in the question, with one representing "very bad knowledge" and five representing "very good knowledge". Table 3 confirms no statistically significant differences between Group I and Group II. However, we were surprised by the participants' confidence in their knowledge of ChatGPT (the median for both groups was 3). The background study was conducted in February 2023, confirming our assumption that students would use ChatGPT in our course. Therefore, this evidence showed us that adjustments were needed in the execution of the Programming II course.

Table 3. ChatGPT knowledge comparison between groups (Mann–Whitney test).

Part	Mean	N	Std. Dev.	B	Mean Rank	Z	*p*-Value
Group I	2.97	89	1.08	3.00	84.79	−1.145	0.252
Group II	3.17	88	1.05	3.00	93.26		

5.2. Comparison

Table 4 shows the results of both groups' performance in lab work. The average lab work success of Group I, which used ChatGPT, was 65.27%, whilst the average score of Group II (no ChatGPT) was only slightly better, 66.72%. Results around 66% are due to participants' decisions to finish just mandatory assignments; only a small number of students decided to work on optional assignments. Note that the mandatory and optional weekly assignments are complementary—usually, optional assignments cover advanced topics. Table 4, surprisingly, shows that results from the lab work on Group I were worse, and, with that, not statistically significantly better compared to the lab work results from Group II. Hence, we can conclude that using LLM is not a decisive factor if the right actions are taken before the execution of the course. These results are discussed further in the section on threats to validity, where concerns are provided regarding our controlled experiment.

Table 4. Comparison of practical course success between groups (Mann–Whitney Test).

Part	Mean	N	Std. Dev.	Median	Mean Rank	Z	*p*-Value
Group I	65.27	93	26.11	63.00	92.67	−0.306	0.760
Group II	66.72	89	19.71	63.00	90.28		

Table 5 compares the performance (by percentage) of the first, second, and overall (average) groups in the midterm exams. Group I (ChatGPT) and Group II (no ChatGPT) solved the same exams. From Table 5 it can be observed that Group I (ChatGPT) performed slightly better than Group II (no ChatGPT) in terms of average success (mean) on the first midterm. However, the difference was small, and not statistically significant. In both groups, the results of the second midterm exam were approximately 10% worse compared to the first midterm exam. We believe these results are connected with the advanced topics in the second part of the semester in the course of Programming II; this is a common pattern observed every year. In the second midterm exam, the results were opposite to the first midterm exam—Group II (no ChatGPT) outperformed treatment Group I (ChatGPT) by around 2%. Still, the results were not statistically significantly better. The latter observation is also accurate for the overall midterm results (average between the first and second midterms)—we could not confirm statistically significant differences between the midterm results between both groups. However, Group II (no ChatGPT) performed slightly better (65.96% vs. 66.58%). Before the experiment, we assumed that Group I (ChatGPT) would have significantly worse results than Group II, which was wrong. As described earlier, both groups were involved in paper-based midterm exams.

Table 5. Comparison of midterm success between the groups (Mann–Whitney test).

Midterm	Part	Mean	N	Std. Dev.	Median	Mean Rank	Z	*p*-Value
First	Group I	68.98	93	24.94	79.00	93.11	−0.421	0.674
	Group II	67.89	89	23.66	74.00	89.82		
Second	Group I	55.72	81	21.45	60.00	75.23	−0.666	0.505
	Group II	58.12	73	21.17	60.00	80.02		
Overall	Group I	65.96	81	18.29	71.00	76.88	−0.181	0.856
	Group II	66.58	73	17.70	70.50	78.18		

The results were similar for comparison of the overall results. Table 6 shows that Group I's average score of overall success was 65.93%. In contrast, Group II achieved a slightly higher average score of 66.61%. Note that the overall grade breakdown was constituted from 50% of midterm exams and 50% of practical assignments. The students received bonus points for extra tasks (usually, no more than 5%). Table 6 reveals no statistically significant difference between the overall success of Group I and Group II, as determined by the Mann–Whitney test.

Table 6. Comparison of course final achievements between the groups (Mann–Whitney test).

Part	Mean	N	Std. Dev.	Median	Mean Rank	Z	*p*-Value
Group I	65.93	93	25.14	68.00	92.82	−0.345	0.730
Group II	66.61	89	21.34	66.00	90.12		

These results (see Tables 4–6, again) allow us to accept all three null hypotheses, and confirm that, in our study, there was no influence of ChatGPT on midterm exams, practical assignments, and final results.

5.3. Feedback Results

As described in Section 4.4, in the last week of the semester, we asked participants to complete a questionnaire about the course and specific actions devoted to ChatGPT. The feedback provided by the students at the end of the semester provided a further understanding of the previous subsection's results.

The participants could answer questions from home (the questionnaire was on our course web page). However, if they came to the last week's lab session, they were encouraged to fill in a questionnaire at the beginning. Note that the number of received answers deviates from the number of participants involved in the midterm exams (i.e., Group I submitted 69 answers while 81 participated in the second midterm exam). We submitted additional messages to participants, but some did not respond to our calls. The missing feedback corresponds to dropout students who did not finish this course and were not present in the classroom at the end of the semester. This is one of the threats to validity and is discussed further later.

5.3.1. Course Complexity

Table 7 shows the results from the feedback questionnaire, where the participants' perspectives were captured on the complexity of the whole course. We used a five-point Likert scale, with one representing "low complexity" and five representing "high complexity". Unsurprisingly, the results show that course complexity was higher for Group II (3.01 vs. 3.17), which did not use ChatGPT. However, the Mann–Whitney test did not exhibit statistically significant differences (Table 7). The statistical test results suggest that the course was equally complex for both groups. We speculate that the slightly different results in course complexity may result from the support of ChatGPT in helping participants understand the course topics better.

Table 7. Participants' opinion on course complexity between the groups (Mann–Whitney test).

Part	Mean	N	Std. Dev.	Median	Mean Rank	Z	*p*-Value
Group I	3.01	69	0.80	3.00	63.45	−1.205	0.228
Group II	3.17	64	0.72	3.00	70.83		

5.3.2. The Usage of ChatGPT during the Semester—Group II

Although we agreed with students to have two groups—one using ChatGPT and the other not, we were not sure if they would obey this decision. Therefore, we asked both

groups weekly if they were using ChatGPT and for what purpose. The results showed that both groups followed our suggestions.

However, some students from Group II did not follow the instructions, as indicated in Figure 3, and used ChatGPT for almost every practical assignment. We decided to eliminate these eight students from the background, study, and feedback results since they corrupted the group, and inclusion would compromise the statistical results as explained earlier in this section. This is why the number of participants in Group II is slightly smaller than in Group I.

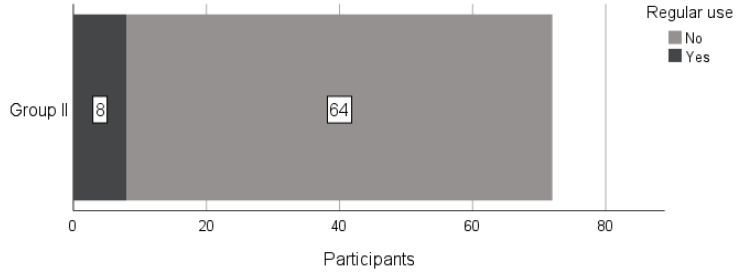

Figure 3. Number of participants in Group II that used ChatGPT regularly for practical assignments.

5.3.3. The Usage of ChatGPT during the Semester—Group I

We warned Group I that excessive use of ChatGPT can lead to worse results on the paper-based midterm exams. Figure 4 confirms that most participants took our advice.

One of the motivating research questions from the Introduction section is whether students would use ChatGPT without hesitation if we allowed it. Figure 4 shows that, although 69 participants were allowed to use ChatGPT in Group I, only 21 reported using it for all assignments. These results indicate that the measures taken before and between semesters (e.g., hand-written exams and additional tasks in the classroom) probably affected the participants' decision not to use ChatGPT too frequently.

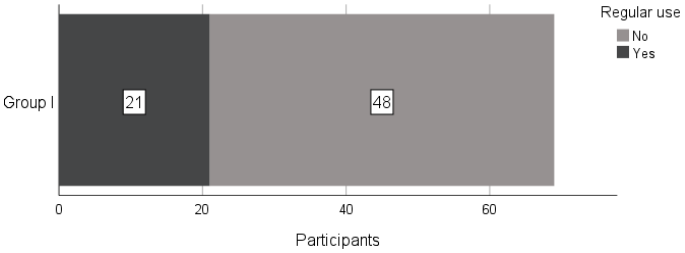

Figure 4. Number of participants in Group I that used ChatGPT regularly for practical assignments.

5.3.4. Influence of ChatGPT on Exam Grade

Our concern before the semester was how ChatGPT usage would influence knowledge and students' grades. Despite our concerns, Figure 5 shows only 44 participants from Group I (ChatGPT), which is only half of all responses, answered with the benefits of its use with lab work assignments. From these results, we cannot state that we received unanimous results.

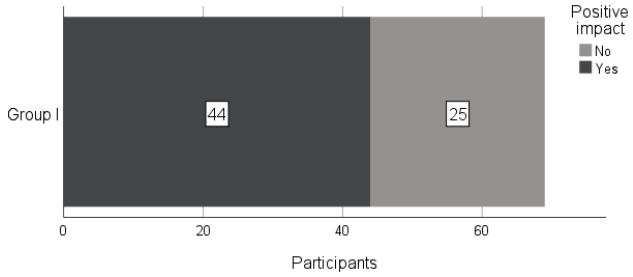

Figure 5. The positive impact of ChatGPT on course grade (Group I).

5.3.5. Means of Use

Software engineers may use ChatGPT for a wide variety of purposes, such as code generation, optimization, comparison, and explanation. In our classroom setting we wanted to minimize code generation as much as possible through adjustments for practical assignments.

Figure 6 reveals that the adjustments served their intended goals. The participants of Group I were using it more for code optimization and comparison with their code than code generation. Again, we can assume that our actions and specific decisions before the semester did affect the use of code generation. Note that Figure 6 shows the results of a multiple-response question—participants chose one or more answers from various alternatives.

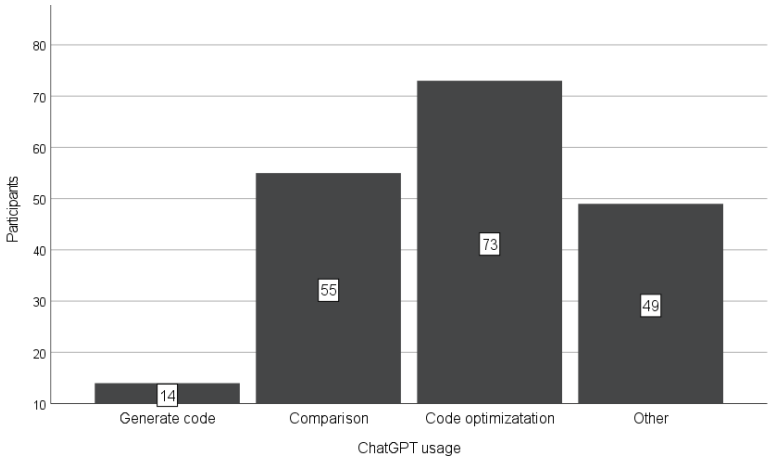

Figure 6. The purpose of ChatGPT use (Group I).

5.3.6. Code Understandability

We were also interested in the satisfaction of the programming code received from ChatGPT. Therefore, Group I (ChatGPT) answered questions regarding the understandability of the code received from ChatGPT.

Again, we used a five-point Likert scale in this question, with one representing "not understandable" and five meaning "very understandable". From Figure 7, we can see that the code received from ChatGPT was very understandable to Group I. Most participants marked understandable or very understandable (four or five in Figure 7).

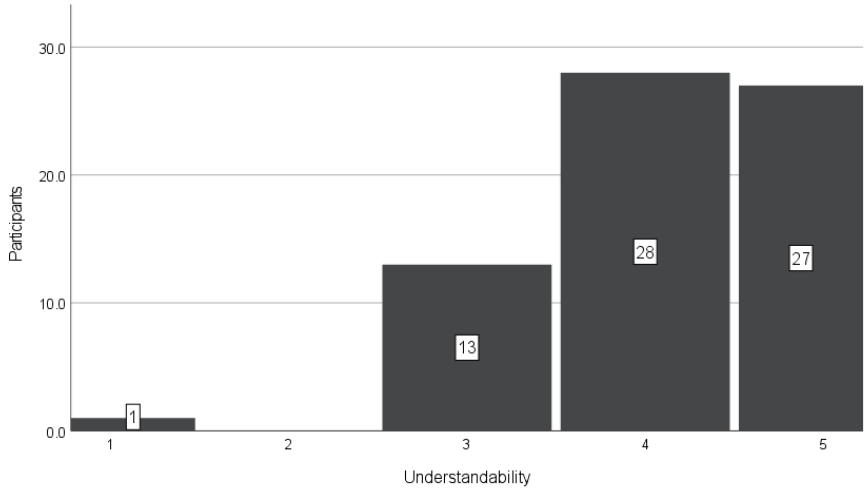

Figure 7. The understandability of code received from ChatGPT (Group I).

5.3.7. Acceptance of ChatGPT among Students outside Programming

We wanted to know whether ChatGPT is accepted among students beyond programming assistance. Since its release in November 2022, 70% of our students reported regular use in June 2023. The participants often reported explanations, instructions, understanding, examples, etc. Some exciting uses can also be seen from the word cloud in Figure 8 (life, generating, summaries), some study-specific (theory, algorithms, concepts, code, syntax, etc.), and some general ones as well (search, every day, everywhere, etc.).

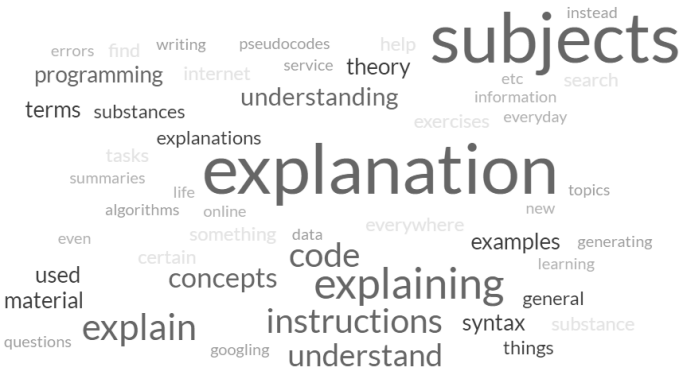

Figure 8. Uses of ChatGPT beyond programming.

5.3.8. Future Use of ChatGPT in Programming

As a final question, the participants answered whether they would use ChatGPT for programming in the future. Most answers from Group I were positive as seen in Figure 9. To comprehend the high confidence in future technology adoption depicted in Figure 9, it is imperative to contextualize these findings within a broader context and correlate them with the insights gleaned from Figures 6 and 7. As illustrated in Figure 6, Computer Science students leverage ChatGPT for multifaceted self-assistance, extending beyond tasks such as code optimization and comparison to encompass various other applications as evidenced by the notable proportion of respondents selecting "other" in Figure 6. Additionally, Figure 7 underscores the exceptional clarity of the generated program code. We believe the combination of these factors has influenced the substantial percentage of students expressing their intent to continue utilizing ChatGPT in the future significantly.

The comparison between Figures 4 and 9 reveals that we introduced specific changes to lab work assignments successfully. Although they liked to use the ChatGPT as a programming assistance tool (Figure 9), they were not able to use it in our specific execution as much as they would want to (Figure 4).

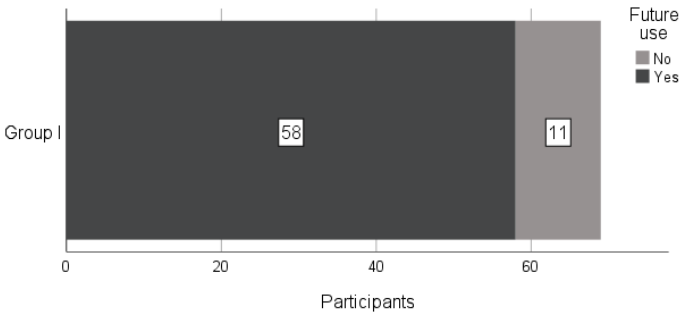

Figure 9. Future use of ChatGPT for programming.

6. Threats to Validity

This section discusses the construct, internal, and external validity threats [46] of our controlled experiment.

6.1. Construct Validity

Construct validity is how well we can measure the concept under consideration [47,48]. In our experiment, we wanted to measure the effect of ChatGPT on the Programming II course results.

We designed several assignments in which the participants were asked to understand the description of the problem and provide implementation in C++ code. With ChatGPT available, we adjusted the assignment definitions. These adjustments made assignments diverse in type (new or extensions), provided code, input/output, the given main program, etc. Figures were given where possible. The participants had to provide a complete implementation. Hence, additional functionalities were given to the participants in the lab session. It is possible that a specific assignment chosen or additional functionality given in the lab session or assessment could have affected the results. However, we have no evidence to suggest that this threat was present.

The complexity of assignments could cause another threat to validity. Altogether, there were 22 assignments, 11 of which were mandatory. We started with simple object-oriented problems: the first assignments had only one class, the following mandatory assignment used aggregation, the compulsory next assignment included inheritance, etc. These assignments started with straightforward problems and advanced during the semester. At the end of the semester, practical assignments contained ten or even more classes. It is unclear if our conclusions would remain the same if all the assignments were equally complex and how the complexity of programming tasks affected the assistance from ChatGPT (Group I). Indeed, the number of classes included in a single assignment is only one complexity metric; additional considerations include control flows and the programming constructs in the code. In general, however, our participants needed much more time to solve practical assignments at the end of the semester than at the beginning.

Our midterm exams are designed to test theoretical knowledge through practical assignments. Therefore, midterm exams are close to the practical assignments given to participants just before the midterm exam. From the point of view of construct validity, we would not know the outcome if we had theoretical questions directly in the midterm exams.

Another construct validity concern is the choice of programming languages used for the course: how programming languages affect ChatGPT-generated results and the measured impact of ChatGPT. It would be interesting to have a replication study [49,50] with another programming language for first-year students (e.g., Python).

While ChatGPT is a leading contender in LLMs, alternative models exist (e.g., Claude). Our weekly feedback questionnaires did not explore the usage of other LLMs, and our findings are restricted to ChatGPT. In addition, in spite of our recommendation of GPT-3.5, some participants may have acquired GPT-4, which was released during the execution of our experiment. We did not account for or inquire about this variable in the experiment or feedback questionnaire. Consequently, we cannot ensure that certain students in Group I did not leverage GPT-4, potentially influencing better results, particularly in practical assignments involving UML class diagrams.

In our experiment, we did not isolate the impact of ChatGPT on Group II participants. Alternative assistance sources were available to Group II participants throughout the semester. Considering the potential substitution effect of other online resources, any observed differences between the groups may not precisely capture the distinct influence of ChatGPT access. It is worth mentioning that these alternative resources were equally accessible to Group I participants, too. Consequently, the sole distinguishing factor between the two groups is the utilization of ChatGPT for Group I.

The infrequent ChatGPT usage by Group I participants (Figure 4) may in part result from the format of paper-based midterm exams. In addition, infrequent use may contribute to a lack of familiarity with employing ChatGPT effectively, potentially reducing the difference between the control and treatment groups. While these factors may be viewed as threats to the construct validity, they are aligned with our goal of collecting empirical evidence if we should encourage/discourage future students in Programming II from

utilizing ChatGPT and similar LLMs. In our view, the *autonomy* granted to participants in Group I to decide whether to leverage LLMs when encountering challenges or learning new concepts—instead of forcing all participants in that group to use ChatGPT frequently—is a *feature* consistent with realistic classroom learning. Our results affirm the potential to permit and facilitate the use of LLMs in the subsequent executions of the Programming II course if the assessment stays the same or similar.

6.2. Internal Validity

Internal validity is the degree of confidence that other confounding or accidental factors do not influence the relationship under test.

There is a potential risk that the participants in Group II also used ChatGPT. To overcome this threat, participants were required to report their ChatGPT usage weekly. We asked them again in the final feedback questionnaire. We saw no considerable deviation from the final report compared to the weekly reports. Therefore, we only provide final feedback on ChatGPT usage in the paper (see Figure 3, again), and, as described in Section 5, eliminated eight participants (who reported using ChatGPT in Group II) from the statistical results.

Some participants also expressed their disapproval of the midterm exams being performed on paper rather than on computers. The participants were missing the basic tools provided by IDEs, like code auto-complete, syntax highlighting, and code generation (constructors, set/get methods) that are usually provided by IDEs like CLion, a recommended IDE in the course Programming II development environment for lab work. We chose the offline approach for two different reasons. First, this controlled experiment was performed with first-year students. Some of them started with programming a few months ago. Therefore, we did not want them to use Copilot or similar AI tools that help students with code generation. The other reason is connected to their experience with IDEs. We wanted to measure the participants' understanding and ability to provide object-oriented code, and avoid the influence of their experience with IDEs.

The feedback study was an optional assignment at the end of the semester. Some participants did not complete the questionnaire in our e-learning platform (49 out of 182). The missing responses represent a selection threat because the effect on the feedback of the missing results is unknown. Nonetheless, since the missing feedback represents 27% of all the participants, this is not a severe threat to the validity of our controlled experiment.

Another internal threat to validity arises due to the absence of training on the effective utilization of ChatGPT provided to students. Had the tool been employed in conjunction with training, the learning outcomes could have potentially differed for Group I, and, consequently, the outcomes could have varied from the one obtained in our study. It is worth mentioning that novice programmers frequently encounter challenges in formulating prompts, due to a deficit in understanding the notion of large language models. This limitation underscores the importance of incorporating comprehensive training to enhance students' proficiency in utilizing ChatGPT effectively and maximizing its potential. The same is true for conducting empirical studies on ChatGPT.

6.3. External Validity

External validity examines whether the findings of an experiment can be generalized to other contexts.

Since only first-year students participated in our experiment, there is an external validity threat to whether the results can be generalized to students from other study years. Again, in this empirical research, we were interested in novice programmers.

Another external validity threat is the generalization of the study results to other courses—we do not know what the results would be if this experiment were part of other courses in the same school year.

This study shows no statistically significant difference in the usage of ChatGPT on midterm results and practical assignments. It would be interesting to see whether results

may differ for sub-categories of students taking Programming II, such as those with significant prior programming experiences, or those who have previously taken certain courses.

The findings of this study stem from an experiment conducted at a single university, prompting considerations regarding the generalizability of the results. Our outcomes may be subject to influence from factors such as demographic characteristics, cultural nuances, and the scale of the institution (specifically, the number of computer science students). To address this limitation, we reveal data from our experiment and encourage replications. Engaging in multi-institutional and multinational studies could provide a more comprehensive understanding of ChatGPT's impact on the learning experiences of novice programmers in computer science education, yielding more precise and robust results.

7. Conclusions

ChatGPT has proven to be a valuable tool for many different purposes, like providing instant feedback and explanations. However, many skeptics emphasize that ChatGPT should not substitute learning and understanding in classrooms. We must exchange opinions and experiences when a transformative and disruptive technology occurs in the education process. Our study was motivated by this high-level goal.

This paper presents a controlled experiment that analyzes whether ChatGPT usage for practical assignments in a Computer Science course influences the outcome of learning. We formed two groups of first-year students, one that was encouraged to use ChatGPT and the other that was discouraged. The experiment evaluated a set of common hypotheses regarding the results from lab work, midterm exams, and overall performance.

The main findings suggest the following:

- *Comparing the participants' success in practical assignments between groups using ChatGPT and others not using it, we found that the results were not statistically different (see Table 4, again).*
  We prepared assignments and lab sessions in a way that minimized the likelihood that ChatGPT may help participants blindly without learning. Our results confirm that our efforts were successful.
- *Comparing the participants' success in midterm exams between groups using ChatGPT and others not using it, we found that the results were also not statistically different (see Table 5, again).*
  Although Group I was using ChatGPT, our adjustments probably resulted in enough effort in learning by that treatment group. Therefore, their results were equal to the control group that was discouraged from using ChatGPT.
- *Comparing the participants' overall success in a course on Programming II between groups using ChatGPT and others not using it, we found that the results were also not statistically different (see Table 6, again).*
  This means that our specific execution of the course (with all the introduced adjustments) allows using ChatGPT as an additional learning aid.

Our results also indicate that participants believe ChatGPT impacted the final grade positively (Figure 5), but the results do not confirm this on the lab work (Table 4), midterm exams (Table 5), nor the final achievements (Table 6). They also reported positive learning experiences (e.g., program understanding; see Figure 7). In addition, we found that ChatGPT was used for different purposes (code optimization, comparison, etc., as indicated in Figure 6). The participants confirmed strongly that they will most likely use ChatGPT (Figure 9).

Future Work

Our study results and ChatGPT-oriented adjustments must be taken with caution in the future. Improvements in large language models will likely affect adjustments (specifically for practical assignments). ChatGPT's ever-evolving nature will probably drive our adjustments tailored to AI technology's current state. ChatGPT-oriented adjustments might erode relevance swiftly, necessitating periodic updates and reassessments to remain robust and practical. We wish to emphasize the importance of assignment defenses and the accompanying discussion with a student. For courses where interactive assignment

defenses are not used as a key form of evaluation, the adoption of ChatGPT may need to be considered carefully. For example, if teaching assistants were only to test the correctness of the code submitted by students without any interactive communication, the results of the evaluation may be different.

This study needs additional replications [38,49]. Different problems (applications) need to be applied to lab work with a different programming language, to name a few possibilities for strengthening the validity of our conclusions. In addition, we need to compare the results from midterm exams using IDE support. It would be interesting to see how the use of development tools affects the results of midterm exams. We are also interested in using our experiment design and specific adjustments in the introductory programming (CS1) course, an introductory course in the Computer Science program, as ChatGPT is successful with basic programming concepts and providing solutions. An empirical study to understand the obtained essential skills (critical thinking, problem-solving, and group work skills) [19] is also necessary, to understand its potential impact for future Computer Science engineers. As discussed in the section on threats to validity, broadening the perspective in replicated studies to involve more institutions and conducting a multi-institutional and multinational study has the potential to yield a deeper comprehension of the integration of large language models in education, leading to more precise and robust outcomes compared to the results presented in this study.

Our future research endeavors in empirical studies with students and ChatGPT should address the limitations of traditional performance comparative metrics (also used in this empirical study). By enriching our research with qualitative assessments, we could uncover profound insights into cognitive engagement and pedagogical interactions driven by AI technology. These metrics could offer a more comprehensive understanding of its impact on teaching and learning processes.

Besides the research directions highlighted above, there exist a multitude of directions associated with the integration of AI technology into pedagogical processes that warrant further investigation. It is essential to delve deeper into the identified risks associated with integrating ChatGPT into educational settings. These risks include the potential unreliability of generated data, students' reliance on technology, and the potential impact on students' cognitive abilities and interpersonal communication skills. Exploring these risks comprehensively is crucial for informing educators about the challenges and limitations of incorporating AI technology in pedagogy. Another intriguing direction for research would involve examining the positive impacts of ChatGPT. Future experiments aimed at investigating the potential educational benefits of large language models could yield crucial insights into their overall impact on learning. On the other hand, we must study the benefits not only for students but also for educators. These include educators' abilities to automate various tedious tasks, such as assessment preparation, monitoring academic performance, generating reports, etc. This technology can act as a digital assistant for educators, assisting with generating additional demonstration examples and visual aids for instructional materials. Understanding how educators utilize these positive features can provide insights into optimizing ChatGPT's role in educational environments. Furthermore, future research should address the limitations of ChatGPT in answering questions. Examining students' reactions when ChatGPT fails to answer questions correctly is essential for understanding their perceptions and experiences with AI technology in learning contexts. This insight can guide the development of interventions to support students' interaction with ChatGPT and mitigate potential frustrations or challenges they may encounter. These and many more topics hold great relevance for the education community and merit thorough exploration.

Author Contributions: Conceptualization, T.K.; methodology, T.K.; software, T.K. and D.O.; validation, M.M. and Y.D.L.; investigation, T.K., D.O., M.M. and Y.D.L.; writing—original draft preparation, T.K., D.O., M.M. and Y.D.L.; writing—review and editing, T.K., D.O., M.M. and Y.D.L. All authors have read and agreed to the published version of the manuscript.

Funding: The first, second and fourth authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0041). The third author acknowledges the financial support from the Fulbright Scholar Program.

Institutional Review Board Statement: Ethical review and approval were waived for this study because the tests had the form of a midterm exam.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available in https://github.com/tomazkosar/DifferentialStudyChatGPT, accessed on 12 January 2024.

Acknowledgments: The authors wish to thank the whole team of the Programming Methodologies Laboratory at the University of Maribor, Faculty of Electrical Engineering and Computer Science, for their help and fruitful discussions during the execution of the controlled experiment.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Stokel-Walker, C.; Van Noorden, R. What ChatGPT and generative AI mean for science. *Nature* **2023**, *614*, 214–216. [CrossRef]
2. MacNeil, S.; Tran, A.; Mogil, D.; Bernstein, S.; Ross, E.; Huang, Z. Generating diverse code explanations using the GPT-3 large language model. In Proceedings of the 2022 ACM Conference on International Computing Education Research, Virtual, 7–11 August 2022; Volume 2, pp. 37–39.
3. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://www.mikecaptain.com/resources/pdf/GPT-1.pdf (accessed on 24 September 2023).
4. OpenAI. ChatGPT. 2023. Available online: https://chat.openai.com/ (accessed on 24 September 2023).
5. Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.P.d.O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating large language models trained on code. *arXiv* **2021**, arXiv:2107.03374.
6. Tian, H.; Lu, W.; Li, T.O.; Tang, X.; Cheung, S.C.; Klein, J.; Bissyandé, T.F. Is ChatGPT the Ultimate Programming Assistant–How far is it? *arXiv* **2023**, arXiv:2304.11938.
7. Rahman, M.M.; Watanobe, Y. ChatGPT for education and research: Opportunities, threats, and strategies. *Appl. Sci.* **2023**, *13*, 5783. [CrossRef]
8. Shoufan, A. Exploring Students' Perceptions of ChatGPT: Thematic Analysis and Follow-Up Survey. *IEEE Access* **2023**, *11*, 38805–38818. [CrossRef]
9. Muñoz, S.A.S.; Gayoso, G.G.; Huambo, A.C.; Tapia, R.D.C.; Incaluque, J.L.; Aguila, O.E.P.; Cajamarca, J.C.R.; Acevedo, J.E.R.; Rivera, H.V.H.; Arias-Gonzáles, J.L. Examining the Impacts of ChatGPT on Student Motivation and Engagement. *Soc. Space* **2023**, *23*, 1–27.
10. Qureshi, B. Exploring the use of ChatGPT as a tool for learning and assessment in undergraduate computer science curriculum: Opportunities and challenges. *arXiv* **2023**, arXiv:2304.11214.
11. Milano, S.; McGrane, J.A.; Leonelli, S. Large language models challenge the future of higher education. *Nat. Mach. Intell.* **2023**, *5*, 333–334. [CrossRef]
12. Dempere, J.; Modugu, K.; Hesham, A.; Ramasamy, L.K. The Impact of ChatGPT on Higher Education. *Front. Educ.* **2023**, *8*, 1206936. [CrossRef]
13. DeFranco, J.F.; Kshetri, N.; Voas, J. Are We Writing for Bots or Humans? *Computer* **2023**, *56*, 13–14. [CrossRef]
14. Cao, J.; Li, M.; Wen, M.; Cheung, S.C. A study on prompt design, advantages and limitations of ChatGPT for deep learning program repair. *arXiv* **2023**, arXiv:2304.08191.
15. Qin, C.; Zhang, A.; Zhang, Z.; Chen, J.; Yasunaga, M.; Yang, D. Is ChatGPT a general-purpose natural language processing task solver? *arXiv* **2023**, arXiv:2302.06476.
16. Dwivedi, Y.K.; Kshetri, N.; Hughes, L.; Slade, E.L.; Jeyaraj, A.; Kar, A.K.; Baabdullah, A.M.; Koohang, A.; Raghavan, V.; Ahuja, M.; et al. "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *Int. J. Inf. Manag.* **2023**, *71*, 102642. [CrossRef]
17. Winslow, L.E. Programming pedagogy—A psychological view. *ACM SIGCSE Bull.* **1996**, *28*, 17–22. [CrossRef]
18. Lukpat, A. ChatGPT Banned in New York City Public Schools over Concerns about Cheating, Learning Development. 2023. Available online: https://www.wsj.com/articles/chatgpt-banned-in-new-york-city-public-schools-over-concerns-about-cheating-learning-development-11673024059 (accessed on 24 September 2023).
19. Sánchez-Ruiz, L.M.; Moll-López, S.; Nuñez-Pérez, A.; Moraño-Fernández, J.A.; Vega-Fleitas, E. ChatGPT Challenges Blended Learning Methodologies in Engineering Education: A Case Study in Mathematics. *Appl. Sci.* **2023**, *13*, 6039. [CrossRef]
20. Susnjak, T. ChatGPT: The end of online exam integrity? *arXiv* **2022**, arXiv:2212.09292.
21. Yilmaz, R.; Yilmaz, F.G.K. Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Comput. Hum. Behav. Artif. Hum.* **2023**, *1*, 100005. [CrossRef]

22. Geng, C.; Yihan, Z.; Pientka, B.; Si, X. Can ChatGPT Pass An Introductory Level Functional Language Programming Course? *arXiv* **2023**, arXiv:2305.02230.
23. Shoufan, A. Can Students without Prior Knowledge Use ChatGPT to Answer Test Questions? An Empirical Study. *ACM Trans. Comput. Educ.* **2023**, *23*, 45. [CrossRef]
24. King, M.R.; ChatGPT. A conversation on artificial intelligence, chatbots, and plagiarism in higher education. *Cell. Mol. Bioeng.* **2023**, *16*, 1–2. [CrossRef]
25. Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M.C.; Regnell, B.; Wesslén, A. *Experimentation in Software Engineering*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
26. Chowdhary, K.R. Natural language processing. In *Fundamentals of Artificial Intelligence*; Springer: New Delhi, India, 2020; pp. 603–649.
27. King, M.R. The future of AI in medicine: A perspective from a Chatbot. *Ann. Biomed. Eng.* **2023**, *51*, 291–295. [CrossRef]
28. Floridi, L.; Chiriatti, M. GPT-3: Its nature, scope, limits, and consequences. *Minds Mach.* **2020**, *30*, 681–694. [CrossRef]
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
30. Adamopoulou, E.; Moussiades, L. Chatbots: History, technology, and applications. *Mach. Learn. Appl.* **2020**, *2*, 100006. [CrossRef]
31. Jeon, J.; Lee, S.; Choe, H. Beyond ChatGPT: A conceptual framework and systematic review of speech-recognition chatbots for language learning. *Comput. Educ.* **2023**, *206*, 104898. [CrossRef]
32. Hughes, A. ChatGPT: Everything You Need to Know about OpenAI's GPT-4 Tool. 2023. Available online: https://www.sciencefocus.com/future-technology/gpt-3 (accessed on 26 September 2023).
33. White, J.; Fu, Q.; Hays, S.; Sandborn, M.; Olea, C.; Gilbert, H.; Elnashar, A.; Spencer-Smith, J.; Schmidt, D.C. A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv* **2023**, arXiv:2302.11382.
34. White, J.; Hays, S.; Fu, Q.; Spencer-Smith, J.; Schmidt, D.C. ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design. *arXiv* **2023**, arXiv:2303.07839.
35. Giner-Miguelez, J.; Gómez, A.; Cabot, J. A domain-specific language for describing machine learning datasets. *J. Comput. Lang.* **2023**, *76*, 101209. [CrossRef]
36. de la Vega, A.; García-Saiz, D.; Zorrilla, M.; Sánchez, P. Lavoisier: A DSL for increasing the level of abstraction of data selection and formatting in data mining. *J. Comput. Lang.* **2020**, *60*, 100987. [CrossRef]
37. Kasneci, E.; Sessler, K.; Küchemann, S.; Bannert, M.; Dementieva, D.; Fischer, F.; Gasser, U.; Groh, G.; Günnemann, S.; Hüllermeier, E.; et al. ChatGPT for good? On opportunities and challenges of large language models for education. *Learn. Individ. Differ.* **2023**, *103*, 102274. [CrossRef]
38. Kosar, T.; Gaberc, S.; Carver, J.C.; Mernik, M. Program comprehension of domain-specific and general-purpose languages: Replication of a family of experiments using integrated development environments. *Empir. Softw. Eng.* **2018**, *23*, 2734–2763. [CrossRef]
39. Sonnleitner, P.; Brunner, M.; Greiff, S.; Funke, J.; Keller, U.; Martin, R.; Hazotte, C.; Mayer, H.; Latour, T. The Genetics Lab. Acceptance and psychometric characteristics of a computer-based microworld to assess complex problem solving. *Psychol. Test Assess. Model.* **2012**, *54*, 54–72.
40. Ouh, E.L.; Gan, B.K.S.; Shim, K.J.; Wlodkowski, S. ChatGPT, Can You Generate Solutions for my Coding Exercises? An Evaluation on its Effectiveness in an undergraduate Java Programming Course. *arXiv* **2023**, arXiv:2305.13680.
41. Moradi Dakhel, A.; Majdinasab, V.; Nikanjam, A.; Khomh, F.; Desmarais, M.C.; Jiang, Z.M.J. GitHub Copilot AI pair programmer: Asset or Liability? *J. Syst. Softw.* **2023**, *203*, 111734. [CrossRef]
42. Imai, S. Is GitHub Copilot a Substitute for Human Pair-Programming? An Empirical Study. In Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings (ICSE '22), Pittsburgh, PA, USA, 21–29 May 2022; pp. 319–321.
43. Asare, O.; Nagappan, M.; Asokan, N. Is GitHub's Copilot as bad as humans at introducing vulnerabilities in code? *Empir. Softw. Eng.* **2023**, *28*, 129. [CrossRef]
44. Likert, R. A technique for the measurement of attitudes. *Arch. Psychol.* **1932**, *22*, 55.
45. Sheskin, D.J. *Handbook of Parametric and Nonparametric Statistical Procedures*, 5th ed.; Chapman and Hall/CRC: New York, NY, USA, 2011.
46. Feldt, R.; Magazinius, A. Validity Threats in Empirical Software Engineering Research—An Initial Survey. In Proceedings of the 22nd International Conference on Software Engineering & Knowledge Engineering (SEKE'2010), Redwood City, CA, USA, 1–3 July 2010; Knowledge Systems Institute Graduate School: Skokie, IL, USA, 2010; pp. 374–379.
47. Ralph, P.; Tempero, E. Construct Validity in Software Engineering Research and Software Metrics. In Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018 (EASE'18), Christchurch, New Zealand, 28–29 June 2018; pp. 13–23.
48. Sjøberg, D.I.K.; Bergersen, G.R. Construct Validity in Software Engineering. *IEEE Trans. Softw. Eng.* **2023**, *49*, 1374–1396. [CrossRef]

49. Shull, F.J.; Carver, J.C.; Vegas, S.; Juristo, N. The role of replications in empirical software engineering. *Empir. Softw. Eng.* **2008**, *13*, 211–218. [CrossRef]
50. Carver, J.C. Towards reporting guidelines for experimental replications: A proposal. In Proceedings of the 1st International Workshop on Replication in Empirical Software Engineering, Cape Town, South Africa, 2–8 May 2010; pp. 1–4.