Generative AI-Enabled Conversational Interaction to Support Self-Directed Learning Experiences in Transversal Computational Thinking

Abdessalam Ouaazki
abdessalam.ouaazki@unine.ch
University of Neuchâtel
Neuchâtel, Switzerland

Kristoffer Bergram
kristoffer.bergram@unine.ch
University of Neuchâtel
Neuchâtel, Switzerland

Juan Carlos Farah
juancarlos.farah@epfl.ch
École Polytechnique Fédérale de
Lausanne, Switzerland

Denis Gillet
denis.gillet@epfl.ch
École Polytechnique Fédérale de
Lausanne, Switzerland

Adrian Holzer
adrian.holzer@unine.ch
University of Neuchâtel
Neuchâtel, Switzerland

ABSTRACT

As computational thinking (CT) becomes increasingly acknowledged as an important skill in education, self-directed learning (SDL) emerges as a key strategy for developing this capability. The advent of generative AI (GenAI) conversational agents has disrupted the landscape of SDL. However, many questions still arise about several user experience aspects of these agents. This paper focuses on two of these questions: personalization and long-term support. As such, the first part of this study explores the effectiveness of personalizing GenAI through prompt-tuning using a CT-based prompt for solving programming challenges. The second part focuses on identifying the strengths and weaknesses of a GenAI model in a semester-long programming project. Our findings indicate that while prompt-tuning could hinder ease of use and perceived learning assistance, it might lead to higher learning outcomes. Results from a thematic analysis also indicate that GenAI is useful for programming and debugging, but it presents challenges such as over-reliance and diminishing utility over time.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; • **Social and professional topics** → **Student assessment**; • **Information systems** → **Computing platforms**.

KEYWORDS

ChatGPT, Programming, Student Perceptions, Chatbots, Generative AI, Education

ACM Reference Format:
Abdessalam Ouaazki, Kristoffer Bergram, Juan Carlos Farah, Denis Gillet, and Adrian Holzer. 2024. Generative AI-Enabled Conversational Interaction to Support Self-Directed Learning Experiences in Transversal Computational Thinking. In *ACM Conversational User Interfaces 2024 (CUI '24), July 08–10, 2024, Luxembourg, Luxembourg*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3640794.3665542

CUI '24, July 08–10, 2024, Luxembourg, Luxembourg

ACM ISBN 979-8-4007-0511-3/24/07
https://doi.org/10.1145/3640794.3665542

1 INTRODUCTION

Computational thinking (CT) has been increasingly considered an important transversal skill set in educational settings [41]. It allows students to model and solve complex problems across various disciplines not limited to computer science, based on principles such as decomposition and algorithm design, contributing to their academic success [58, 59]. Self-direct learning (SDL) activities have been used as a successful strategy to teach and foster CT skills. These learning activities allow individuals to take the initiative to identify their learning needs, set goals, find resources, and evaluate their progress without formal instruction [6]. They emphasize autonomy and the ability to learn independently, adapting to one's own pace and interests. Examples of SDL activities include creating coding projects to solve real-life problems, watching instructional videos, engaging in online discussions, participating in collaborative work, and using online learning platforms for self-evaluation through tests [10].

The advent of generative AI (GenAI) large language models (LLMs), such as GPT-3 [19] and GPT-4 [44], alongside interfaces that leverage these LLMs such as ChatGPT [60], has disrupted the landscape of self-directed learning activities [21]. Since their launch, they have been extensively used to support these activities, providing assistance in learning and tutoring, and demonstrating their potential as valuable teaching assistants [24, 42]. The transformative impact of GenAI-enabled conversational interfaces has been felt across various educational domains such as programming, economics, and mathematics, prompting a surge in academic interest and research [38, 60]. As these conversational agents have pushed the limits of what is possible in natural language processing and coding capabilities [24], they have been extensively used in software development to assist programmers by providing code suggestions, debugging help, and writing code snippets [3].

However, many questions still arise about several user experience aspects of these tools [65]. In this paper, we focus on two

of these issues: personalization and long-term support. Personalization refers to the process of augmenting a GenAI model with strategic prompting techniques to enhance the accuracy and relevance of its output [39]. Recent literature highlights the challenge of personalizing GenAI-based conversational interfaces in educational settings to enhance learning [28]. A solution that has been suggested to overcome this challenge is prompt-tuning, a technique that aims at refining the input prompts to direct the behavior of these models on particular tasks, without the need to modify their underlying parameters [35]. Multiple techniques have been investigated to fine-tune prompts, such as providing representative examples in prompts [15] or providing error-assisted guidance where mistakes found in the generated output are fed back to the system [61]. An alternative strategy that aligns with the principles of computational thinking involves personalizing a GenAI model to serve as a teaching assistant that guides students towards independent problem-solving. This approach is rooted in educational literature suggesting that students benefit most when encouraged to explore and deduce answers on their own [56], a core aspect of computational thinking. To explore the utility of this approach in GenAI-enabled conversational agents, we propose prompt-tuning the model to act as a teaching facilitator based on CT principles. Incorporating computational thinking, the GenAI model can be designed to assist students in deconstructing problems into manageable parts and guiding them towards devising algorithmic solutions to them [59]. In the first phase of this paper, we devise an experiment within a software design class to test the impact of this approach on student's perceptions of chatbots and learning outcomes. We focus on GPT-3 and ChatGPT as GenAI-conversational agents in this study because of their popularity and groundbreaking capabilities in technological research [49].

Regarding the long-term support of GenAI-enabled conversational interfaces, recent literature indicated the usefulness of ChatGPT as a learning assistant in solving complex tasks in semester-long projects in fields like physics [36]. However, in the context of CT, literature on LLMs' usability for assisting with open-ended self-directed activities, especially those extending over several months, remains sparse. Additionally, several studies also indicate that even if ChatGPT holds promise as a valuable tool for learning, it faces challenges such as producing errors or misleading information [5, 38]. These challenges raise questions about the ability of GenAI to support students in long-term projects. In the second phase of this study, we explore this aspect by examining the strengths and limitations of ChatGPT as an assistant in a semester-long self-directed project. This phase of the study aimed to assess the student user experience with ChatGPT for larger tasks that can become increasingly complex over time. Overall, this paper aims to investigate the aforementioned issues by addressing the following overarching research question:

RQ: Can GenAI-enabled conversational interactions support self-directed CT learning activities?

The remainder of this paper is structured as follows: section 2 reviews relevant literature and presents more granular research questions. Following this, section 3 introduces our proposed solution, and sections 4 and 4.3 detail our experimental setup. Subsequently, section 5 reports on the findings of our study, and section 6 discusses the results in light of related literature. Finally, section 7 summarizes our work and concludes our paper.

2 RELATED WORK

In this section, we provide an overview of related literature on conversational agents within educational settings, focusing on their application, and the emerging practice of prompt-tuning. Additionally, we examine the diverse perceptions of these technologies among users.

2.1 Conversational Agents in Education

Conversational agents (CAs) have gained prominence in educational contexts due to their potential to enhance learning experiences [50]. These agents simulate human dialog, engaging learners in natural language conversations. They have the potential to facilitate content integration, offering rapid access to learning materials, boosting motivation and engagement, enabling simultaneous access for numerous users, and providing instant support [43]. In various educational scenarios, chatbots have proven effective, contributing to time efficiency [47], enhancing students' learning capabilities, and fostering greater engagement among students [13]. Moreover, chatbots have been developed and utilized as virtual aides, facilitating administrative and academic duties [53], or addressing common inquiries to customize and improve the user's experience [46, 47, 51].

The launch of ChatGPT in November 2022 has addressed many of the challenges encountered in earlier chatbot models, including the ability to remember past conversation points, comprehend user corrections, and decline unsuitable queries [27]. ChatGPT represents an instance of GenAI LLMs. These models are defined as "a machine-learning system that autonomously learns from data and can produce sophisticated and seemingly intelligent writing after training on a massive data set of text" [57, p.224]. Their advent has considerably expanded the applicability of chatbots to a broader array of educational situations than before [30].

As such, multiple studies have explored the potential of using the GenAI in educational settings. For instance, Firat [18] revealed that integrating AI tools like ChatGPT in education can significantly enhance student engagement and performance by allowing educators to focus more on higher-order skills and mentoring. After analyzing the perceptions of scholars from multiple countries, they suggested that AI can support personalized learning and act as an extension of the human brain, offering transformative changes in the learning process. In another study, Zhai [63] found that ChatGPT can efficiently aid in academic writing with minimal input from the author. The paper suggested a need to adjust educational goals, focusing more on developing students' creativity and critical thinking rather than general skills. These studies, collectively, underscore the potential of GenAI in facilitating more interactive and personalized learning experiences.

2.2 GenAI in Computational Thinking & the Role of Prompt-Tuning

Integrating GenAI conversational agents into computational thinking and programming education has shown promising results. Ouaazki et al. [45] integrated ChatGPT into a master-level computational

thinking course. The study revealed positive associations between the use of ChatGPT and learning outcomes, suggesting that ChatGPT can be a valuable tool in educational settings to enhance learning. It also indicated that ChatGPT has been successfully used by students for coding and debugging purposes. Similarly, Jalil et al. [26] investigated the capabilities of ChatGPT in the context of a traditional software testing course. The paper conducted a comprehensive empirical study, tasking ChatGPT with answering questions from five chapters of a popular software testing textbook. The study showed that ChatGPT successfully responded to a majority of the examined questions. In another study investigating the effectiveness of ChatGPT in automating code review tasks, the authors underscored the superiority of the LLM in performing code reviews compared to traditional methods [22]. These studies underscore ChatGPT's ability to support learning in both educational and professional software development contexts.

Recent literature has investigated prompt-tuning as a technique to augment the powerful capabilities of GenAI conversational agents [15]. Cain [9] discussed the potential of this technique in fostering personalized, engaging, and equitable learning experiences. In the context of programming, Jury et al. [29] introduced an innovative tool that applies prompt-tuning to an LLM to generate interactive worked examples for introductory Python programming. The study indicated that this approach enhanced personalized learning experiences by offering tailored educational content, confirming the effectiveness of prompt-tuning in creating more engaging programming education environments. Furthermore, Zhang et al. [64] evaluated ChatGPT's capabilities in automated program repair using prompts on a dataset containing programming problems and corresponding buggy programs. The study demonstrated ChatGPT's success in fixing bugs, outperforming state-of-the-art LLMs, highlighting the impact of prompt engineering on repair effectiveness, and suggesting that carefully crafted prompts can significantly boost performance.

However, despite indications about the usefulness of prompt-tuning in enhancing GenAI conversational agents for educational purposes, as shown in the studies presented above, the specific design and implementation strategies for prompt-tuning within the context of SDL remain unclear. This leads us to explore the following open research question:

RQ1: How does CT-based prompt-tuning affect the usability of a GenAI-enabled conversational agent, student attitudes, and learning outcomes?

2.3 Perceptions of GenAI-Enabled Conversational Agents

Recent literature has also investigated attitudes towards GenAI conversational agents in educational contexts, showing mixed perceptions about these tools [34]. On the one hand, several studies highlighted a positive perception of these tools, along with key strength elements that contribute to this perception. For instance, Shoufan [52] explored student perceptions of ChatGPT in a computer programming educational context. The study involved a two-stage experiment with computer engineering students who used ChatGPT for various learning activities and provided feedback about their experience. The findings reveal that students generally viewed ChatGPT positively, appreciating its capabilities, human-like interaction, and helpfulness in studies. Similarly, Albayati [1] examined the factors influencing user acceptance of ChatGPT among undergraduate students. The study indicated that undergraduate students take into account several important factors when using ChatGPT, including perceived ease of use, perceived usefulness, and privacy concerns. This highlights the importance of studying these factors to facilitate the integration of LLMs into learning environments. The use of ChatGPT in higher education was also investigated by Elkhodr et al. [16], focusing on its impact on learning outcomes and experiences through three case studies involving undergraduate and postgraduate information and communication technology students. The findings revealed a positive perception of ChatGPT as a useful and enjoyable learning resource, with most students willing to use AI tools in the future.

On the other hand, recent studies also highlighted weaknesses and negative perceptions associated to the use of GenAI. For instance, Zhu et al. [65] indicated that students expressed concerns about ChatGPT's accuracy, with some responses requiring fact-checking and verification. Particularly, students noticed that there were issues related to the system providing misleading or biased decision-making, generic responses lacking depth and context, and limitations based on input prompts. Students also noted ChatGPT's tendency towards repetition, and potential negative impacts on self-discipline and critical thinking. The study also raised concerns about dependency on AI for idea generation and problem-solving. Similarly, [40] expressed worries related to over-dependence and ethical concerns in a comprehensive study of how early adopters use and perceive ChatGPT in educational settings. Given these diverse perspectives on the strengths and weaknesses of GenAI conversational agents, and considering the scarcity of research tackling the use of these agents as assistants in CT semester-long projects, we formulate our second research question:

RQ2: What are the strengths and limitations associated with using a GenAI-enabled conversational agent in the context of a CT semester-long project?

3 SOLUTION DESCRIPTION

To answer our first research question—pertaining to prompt-tuning—we used a system that supports self-directed learning activities (Graasp), augmented with a GenAI-enabled agent (Graasp Bot). Additionally, we used a CT-based prompt to prompt-tune the conversational agent. In this section, we explain this solution in detail.

3.1 Graasp Notebook

Graasp [20] is an open-source digital education platform featuring two distinct interfaces. The first interface—called the *builder*—allows educators to construct their online lessons or assignments using various resources called apps. These apps allow the creation of detailed exercises enhanced with text, images, links, chatbots, and other interactive elements. The second interface—known as the *player*—caters to students. It provides access to the online lessons, allowing students to engage with the content. This includes navigating through sections that present the lectures and exercises developed by the instructor. Figure 1 shows the user interface of the *player* where students can view exercises and write code to

solve them. Students then get feedback on their code, including a validation code if they write the correct code to solve the assignment at hand.

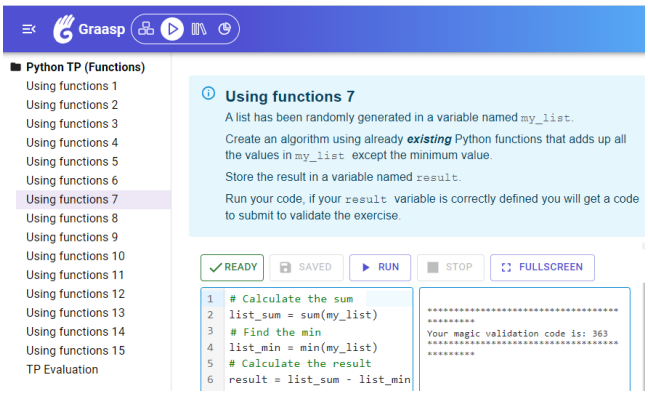


Figure 1: Graasp interface showcasing a sample exercise and code to solve it.

3.2 Graasp Chatbot

Graasp also offers the possibility to integrate the Graasp Bot, a chat agent built on top of OpenAI's API, with every exercise. More specifically—at the moment of this study—Graasp's OpenAI integration used the GPT-3 `text-davinci-003` model, with a `temperature` parameter of 0.9 (higher values make completions of the same prompt more random), a `presence_penalty` of 0.6 (higher values penalize new tokens), a `top_p` parameter of 1 (always returning the best completion), and the maximum number of tokens to be included in the chatbot response fixed at 150 [17]. Students can then interact with this agent to help them solve their exercises. The integration of the chatbot directly within the learning activity provides a seamless interface, allowing students to access this resource within the same web page hosting their exercises.

3.3 CT Prompt Tuning

The Graasp platform allows the customization of the chatbot to include an initial system prompt that is sent to OpenAI's API before students start interacting with the Graasp Bot. This way, the context of the conversation is pre-configured before every new interaction. We used this functionality to introduce a CT prompt to OpenAI's GPT-3 model. This prompt was developed in an iterative way, where our team tested different prompts directly on ChatGPT as suggested by Lingard [37]. We then incrementally improved the prompt until we attained one that behaves properly according to the design principles outlined by Bsharat et al. [8]. These principles emphasize brevity, clarity, and task-specific guidance. Accordingly, the prompt (presented in Figure 2) invites students to describe their current Python exercise in detail or share their existing code, ensuring a focused and contextually relevant starting point for assistance. The prompt then guides the assistant to employ computational thinking strategies—decomposition, pattern recognition, abstraction, and algorithm design—to aid students in understanding and tackling their problems step by step. When providing code solutions, the prompt instructs the assistant to include explanations, enhancing learning through clear annotations. This design maintains a strict focus on Python exercises, with the assistant prepared to gently redirect off-topic inquiries back to the subject at hand, ensuring efficiency and relevance in student support.

```
Context:
You are a highly pedagogical Python teaching assistant designed
   to help students with basic Python exercises in a lab setting.
Mission:
1- Begin by asking about the specific exercise the student is
   working on. Encourage students to provide detailed instructions
   for the exercise or share the code they have written so far.
2- Assist students using computational thinking principles:
   decomposition, pattern recognition, abstraction, and algorithm
   design.
3- Focus on helping students decompose the problem and solve it
   step-by-step, offering the requested Python code as needed.
4- If you provide students with code, explain what the code does,
   preferably in code comments.
5- Remain strictly within this context. If asked unrelated
   questions, kindly remind the student of your primary mission
   and refocus on Python exercises.
```

Figure 2: Fine-tuned prompt provided to the Graasp Bot.

4 EXPERIMENTAL SETUP

In this semester-long study, we investigated CT-based prompt-tuning in closed-ended self-directed lab assignments and explored the long-term user experience with a GenAI conversational agent in an open-ended programming project. The study took place within a software design class during the fall semester of 2023 at a Swiss university. In this class, students were supposed to (a) learn how to formulate a problem in a computational way, (b) devise an algorithmic solution, and (c) design a software solution. Throughout the course, students explored the fundamental principles of software design along with elementary Python programming language elements such as conditions and loops, lists, functions, and dictionaries, as well as more advanced object-oriented concepts like classes, inheritance and error handling. The instructional approach included theoretical lectures using Jupyter Notebooks [32], which allowed students to interactively alter and run code in class, along with self-directed learning tasks.

4.1 Participants

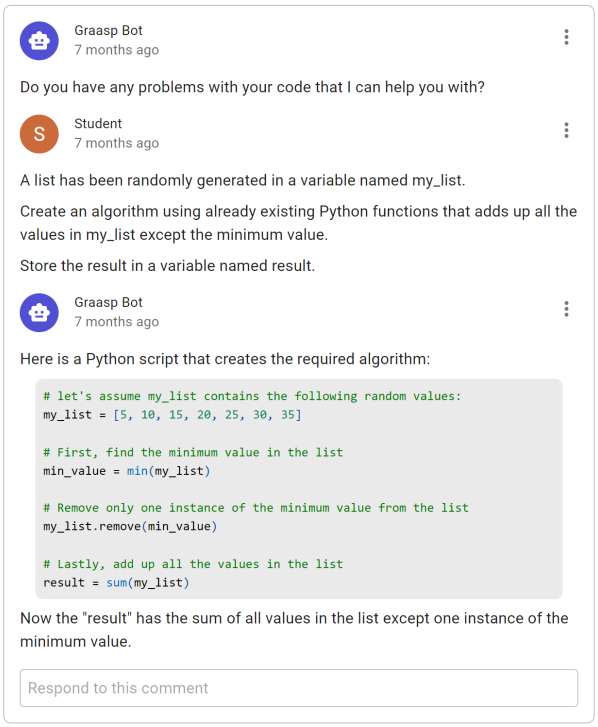
Participants initially included 23 students (8 female, 15 male) enrolled in their 3rd year of bachelor studies in business and economics. However, two students (1 female, 1 male) who did not complete all their assignments were excluded from the analysis. The remaining 21 students reported having used ChatGPT for 6.11 months on average (SD = 3.71) before the beginning of the course. They also reported interacting with ChatGPT for 1.74 days per week on average (SD = 1.63) prior to the course. When inquired about their previous programming experience, 17 participants expressed having prior experience with programming, 2 reported having none, and 2 did not respond to the question.

4.2 Prompt-Tuning Experiment

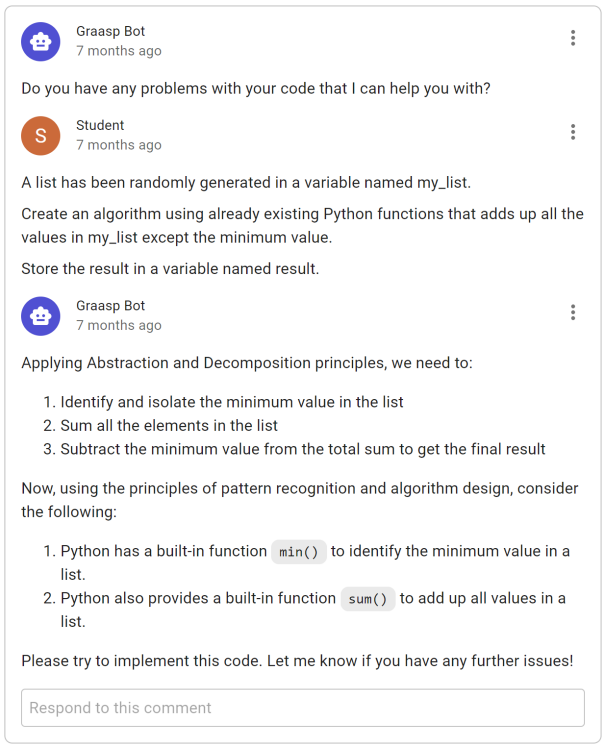
In the first part of this study, we investigated the use of CT-based prompt-tuning in self-directed lab assignments. Students took part in four self-directed lab sessions, during which they were required

to solve Python programming challenges on Graasp. These tasks focused on fundamental concepts such as conditions, loops, lists, functions, and dictionaries. The primary objective of these exercises was to develop practical skills related to creating algorithms and learning Python syntax. Labs 1 through 4 were composed of 25, 15, 15 and 10 exercises, respectively. Students were encouraged to use the Graasp Bot when approaching their exercises. Because the Graasp Bot could provide Python code directly to answer this question, we wanted to decrease the emphasis on performance. We communicated to students that completing 50% or more of the exercises would ensure they receive a full grade, aiming to foster learning rather than just completion.

In order to assess whether the Graasp Bot would be useful for students under varying instructional settings, we employed an ABBA experimental design. Accordingly, the first and last sessions (lab 1 and lab 4) acted as control periods (condition A, hereafter referred to as the *no-prompt* condition) where the Graasp Bot was present but not specifically configured. In contrast, the middle sessions (lab 2 and lab 3) represented the experimental condition (condition B, hereafter referred to as the *CT prompt* condition), where the bot was configured using the fine-tuned prompt presented in section 3.3. Figure 3 illustrates responses of the Graasp Bot when a student copies and pastes a question from the assignment into the chatbot interface, for the two study conditions.



(a) Example interaction in the *no prompt* condition.



(b) Example interaction in the *CT prompt* condition.

Figure 3: Example interactions with the Graasp Bot.

4.3 Long-Term Self-Directed Project

The second part of this study explored the long-term use of ChatGPT as an assistant in a self-directed semester-long project. The project consisted of creating a unique game using as a starting point a sample code featuring an avoidance game where the primary objective is for the player to avoid obstacles or enemies. This initial game (featured in Figure 4) requires players to capture viruses using a face mask to save pangolins. The codebase for this game consisted of 67 lines of code. To produce their own games, students worked in self-organized groups of 3 to 5 members. There were 7 groups in total. Groups were required to devise new rules and objectives, and incorporate clear win-and-lose conditions for their games. The games needed to visually display levels and points to track progress. Students were also instructed to organize their code into distinct classes (such as defining classes for different game characters) and ensure readability and structure. Students implemented their projects on the Replit online IDE, which allowed them to work collaboratively on their codebase [25]. While students were allowed to use ChatGPT as a resource, they were warned against plagiarism and were required to keep a record of their ChatGPT interactions to submit with their final project, ensuring academic integrity and transparency in their use of AI assistance. Figure 5 shows screenshots of the games produced by students for their group projects. To create these games, groups 1 to 7 produced respectively 619, 315, 801, 899, 439, 501 and 290 non-empty lines of code ($M = 552, SD = 216$). It is worth noting that this part of the study was conducted without the Graasp Bot or the use of prompt-tuning. Students interacted with ChatGPT directly through OpenAI's official interface using their personal accounts.

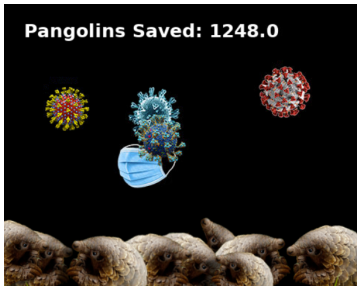


Figure 4: Screenshot of the initial game provided to students.

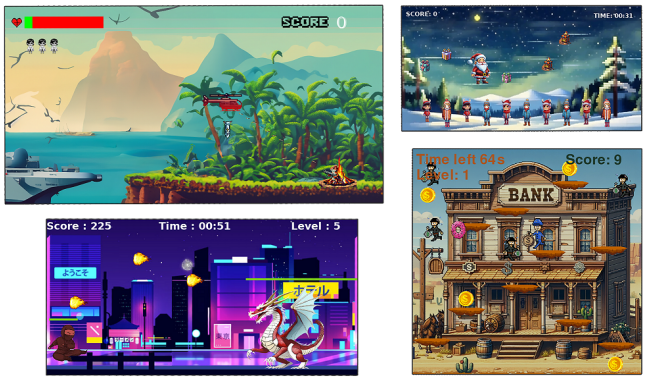


Figure 5: Screenshots of the games produced by students.

4.4 Metrics and Data Collection

In order to gain an understanding of student perceptions and interactions with Graasp Bot, we collected the following data.

4.4.1 Usability. : To assess the usability of Graasp Bot, we focused on its usefulness, ease of use , and the level of learning assistance it provided to students. Following each lab assignment, students were asked to evaluate how useful Graasp Bot was, how easy it was to use, and how effective it was in assisting them with their assignment. Responses were collected on a 7-point Likert scale, ranging from 1 (not at all) to 7 (extremely). Additionally, students were invited to share comments about their use of the Graasp Bot through an open-ended question.

4.4.2 Attitudes Towards Chatbots. : To measure students' attitudes towards Graasp Bot, we used the General Attitude Towards Robots Scale (GAToRS) [33], before and after the experiment. We specifically used the 5 questions from the *personal level positive attitude* section of the scale and the 5 questions from the *societal level positive attitude* section. Each question was answered on a 7-point Likert scale. The total attitudes score was then calculated as the sum of the responses to these 10 questions. These metrics were collected before and after the course.

4.4.3 Learning Outcomes. : We measured learning outcomes through the scores students obtained on their lab assignments. Students got one point when they provided a correct answer to a given exercise, and zero points when they failed to do so. The final lab score was calculated as the sum of the scores for each exercise.

4.4.4 Interaction Logs. : We collected the interaction logs of students with Graasp Bot and ChatGPT when they were using them as assistants. For the lab assignments, we collected the interactions with the Graasp Bot directly from the platform. For the semester projects, we asked students to provide us with their interaction with ChatGPT in the form of an HTML file.

4.4.5 Reflection Reports. : At the end of their projects, students were asked to write reflection reports. We asked them to reflect on their use of the GenAI agents, along with the perceived strengths and weaknesses of these tools according to their experience.

5 RESULTS

We first analyzed the chat logs of student conversations with the Graasp bot in the lab assignments. The chatbot gave 523 answers to students in total across all assignments. We tagged these responses in order to investigate the impact of prompt-tuning on the responses of the conversational agent. We assigned to each response one of three codes: a) *direct answer*, when Graasp Bot provided a direct answer solving the student's problem (similar to Figure 3a), b) *CT-based answer*, when the chatbot responded based on CT principles, such as providing an algorithmic approach or suggesting a breakdown of the problem (similar to Figure 3b), and c) *other*, when the response did not fall on any of these categories. As can be seen in Table 1, the presence of the CT prompt affected the distribution of response types provided by the chatbot. The average number of direct answers per exercise is higher for the *no prompt* condition at 6.54 compared to 3.33 in the *CT prompt* condition. In contrast, the average number of CT-based answers per exercise is higher with the CT prompt, recording 2.13 compared to 0.7 without it. These metrics suggest that the CT-based prompt-tuning shifted the chatbot's response strategy, favoring more computationally focused answers over direct solutions to problems. Additionally, these results also show that the average total number of responses per exercise provided by the chatbot was higher in the *no prompt* condition (8.4) compared to the *CT prompt* condition (6.26).

Table 1: Comparison of Graasp Bot response types.

Metric	No Prompt	CT Prompt
Number of exercises	35	30
Avg. direct answers per exercise	6.54 (78%)	3.33 (53%)
Avg. CT-based answers per exercise	0.7 (8%)	2.13 (34%)
Avg. other answers per exercise	1.16 (14%)	0.8 (13%)
Total answers per exercise	8.4 (100%)	6.26 (100%)

In the rest of this section, we present a quantitative analysis related to our first research question, then we present the results of a qualitative analysis pertaining to the second research question.

5.1 How does CT-based prompt-tuning affect the usability of a GenAI-enabled conversational agent, student attitudes, and learning outcomes?

To provide insights related to this research question, we present a comparative analysis of usability metrics, attitudes towards chatbots, and learning outcomes, across conditions.

5.1.1 Usability. Table 2 shows the results of our analysis of usability between the *no prompt* group and the *CT prompt* one using paired-samples t-tests. For the usefulness of interactions with Graasp Bot, mean scores were marginally lower for the *CT prompt* group ($M = 3.71$) compared to the *no prompt* group ($M = 3.85$), although this difference was not statistically significant ($t = 0.52, p = 0.607$). This suggests that the introduction of CT prompts does not significantly alter students' perceptions of the chatbot's usefulness in this context. Contrastingly, the ease of use experienced a notable decline in the *CT prompt* group ($M = 5.33$) compared to the *no prompt* group ($M = 6.11$), with the difference being statistically significant ($t = 2.33, p = 0.032$). In terms of the perception of the learning assistance Graasp Bot provided to students, the *CT prompt* group reported lower mean scores ($M = 3.42$) compared to the *no prompt* group ($M = 4.05$), with this difference reaching statistical significance ($t = 2.41, p = 0.027$). These perceptions were echoed in some student comments although students expressed overall satisfaction with using the Graasp Bot. For instance, students in the *no prompt* condition noted: "S*o far so good. Grassp bot is very useful for a student who does not have much experience in coding before (for example, Me). It gives me a nice kick off. Thank you for using it,*" and "*it was helpful and it helped me a lot with this challenging assignment.*" Comments from students in the *CT prompt* condition were more mixed with participants noting: "*Graasp bot was a little bit less useful than it was in the previous weeks,*" and "*I prefer to use ChatGPT, he can answer better than the Graasp bot.*"

5.1.2 Attitudes Towards Chatbots. In order to analyze the association between students' interaction with ChatGPT and their attitudes toward chatbots, we used a Partial Least Squares (PLS) path analysis. PLS is a statistical method commonly used in information systems research to discover and model relationships and interactions among research variables [23]. The results of the analysis are presented in Figure 6. These results show that within the *CT prompt* group, there is a significant negative relationship between the number of chats students had with ChatGPT and their post-test attitudes towards the chatbot ($\beta = -0.541, p = 0.036$). The pattern observed suggests that more frequent usage in this group is associated with less favorable perceptions of the chatbot. Conversely, in the *no prompt* condition, we observed a non-significant positive correlation between the number of chats and post-test attitudes ($\beta = 0.188, p = 0.200$). The model also shows that the pre-test attitudes do not significantly predict the number of chats in either the CT prompt or the no-prompt groups, with negligible path coefficients and high p-values, indicating a lack of statistical significance. Furthermore, these initial attitudes do not seem to have a substantial predictive power directly on the post-test attitudes ($\beta = 0.356, p = 0.207$).

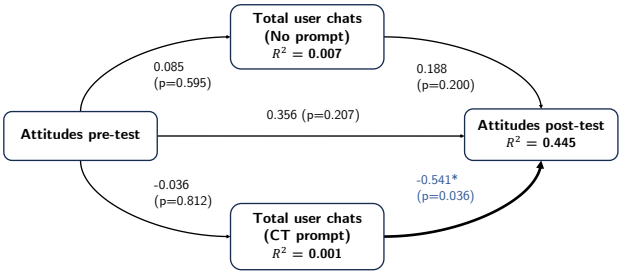


Figure 6: PLS path model of chat interactions and attitudes towards chatbots.

5.1.3 Learning Outcomes. In examining the impact of the CT prompt on student learning outcomes, our analysis revealed a trend towards improved lab grades in the *CT prompt* condition, as illustrated in Figure 7. To ensure consistency and comparability, all lab grades were normalized to a 100-point scale based on the number of exercises in each lab. The analysis reveals that average grades were 86.98 ($SD = 12.06$) during the lab sessions with the CT prompt, compared to 80.81 ($SD = 12.66$) during the *no prompt* phases. A t-test comparing the grade distributions between the two conditions yielded a statistic of -2.366 and a p-value of 0.028, with 20 degrees of freedom. These results suggest a potential positive effect of the CT prompt on lab grades. However, we cannot completely rule out the influence of varying exercise difficulty on these results.

We further examined the potential causes behind the observed improvement in lab grades. Specifically, we considered whether the improvement could be attributed to either an increased rate of lab completion or a rise in the accuracy of completed labs. Firstly, we calculated the completion rate for each condition. Completion rate was defined as the ratio of the number of exercises attempted to the total exercises available. An exercise was marked as attempted if the student ran code in the corresponding code capsule. Our results showed an average completion rate of 98.14 percent ($SD = 4.60$)

Table 2: Descriptive statistics and t-test results for usability metrics (n = 21)

	No Prompt		CT Prompt		t-statistic	p-value
Metric	*M*	*SD*	*M*	*SD*		
Usefulness	3.85	1.61	3.71	1.48	0.52	0.607
Ease of use*	**6.11**	0.72	**5.33**	1.45	2.33	**0.032**
Learning assistance*	**4.05**	1.52	**3.42**	1.64	2.41	**0.027**

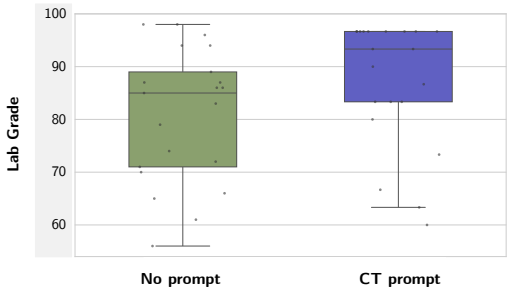


Figure 7: Student lab grades across the *no prompt* and the *CT prompt* conditions.

in the *no prompt* condition, and 96.66 percent ($SD = 8.02$) in the *CT prompt* condition. Statistical analysis using a t-test indicated no significant difference in completion rates between the two conditions ($t = 0.73, p = 0.47$). Next, we assessed the accuracy rate, defined as the percentage of exercises correctly completed out of those attempted. The findings revealed an average accuracy rate of 82.30 percent ($SD = 12.35$) in the *no prompt* condition and 89.81 percent ($SD = 8.61$) in the *CT prompt* condition. A t-test confirmed a statistically significant difference in accuracy rates between the conditions ($t = -2.28, p = 0.028$). Based on these results, the improvement in lab grades appears to be more closely associated with an increase in the accuracy of completing exercises rather than changes in the completion rates.

5.2 What are the strengths and limitations associated with using a GenAI-enabled conversational agent in the context of a CT semester-long project?

To answer this question, we conducted a thematic analysis of the reflection reports students wrote after working on their semester projects. We opted for a coding approach inspired by Braun and Clark's thematic analysis procedure [7]. Initially, two researchers engaged with the students' reflection reports to familiarize themselves with the data and generate initial codes. These codes represented the main ideas included in the reports. Following the initial coding phase, one researcher undertook a detailed coding process, systematically applying the developed codes to each sentence of the reports. Subsequently, a second researcher reviewed this initial coding, verifying code attributions, and ensuring the codes were consistently applied, resulting in agreement between the two researchers. Following this verification, our team engaged in a collaborative process to identify and discuss emerging themes. These themes were then discussed and reviewed to reach a consensus. Lastly, final themes were established and named. This process resulted in identifying several themes that emerged consistently across the groups. These themes describe the student user experience with the tool, along with its strengths and limitations. Out of the five themes that emerged, three relate to ChatGPT's strengths in helping understand, create, and debug code. The other two themes reveal limitations of the LLM, notably the errors it produces, the risk of dependency on the tool, and a diminishing utility over time.

In reporting the results, group numbers have been renamed from Group 1 to Group 6 based on the number of chat prompts the group made to ChatGPT. Based on the submitted chat interactions, Group 1 to Group 6 reported respectively 12, 16, 35, 44, 57, and 170 chat prompts made to ChatGPT. As for group 7, they did not provide their interaction logs.

5.2.1 ***Understanding existing code.*** The first theme that emerged while analyzing student reflections and comments is that they considered ChaGPT as a valuable tool for understanding existing code snippets. Since students were provided with a sample code base to start with, they mentioned that the LLM helped them understand the existing code, which was valuable in their initial development phase. In this regard, students from Group 2 wrote:

> *ChatGPT helped us demystifying Pygame, offering clear explanations and practical code examples. The tool was particularly helpful in the bootstrap phase, providing quick solutions and creative ideas, reducing the initial overwhelm.*

Since the projects were implemented in groups and needed collaborative efforts, students also stated that ChatGPT assisted them in understanding complex code written by their peers. Group 3 explained:

> *ChatGPT also helped us to understand the work of our colleagues. Indeed, within our group, there were more levels of understanding, and some were more quickly at ease with coding and able to produce passages of rather complex code, which might be difficult for another person to understand, despite the comments we tried to add throughout our work.*

This statement underscores ChatGPT's role in facilitating collaborative learning and mutual understanding within diverse skill levels in a team.

5.2.2 ***Generating a new codebase.*** Students also explained how ChatGPT helped them write code for new functionalities. For instance, Group 1 mentioned how the GenAI agent helped them accelerate the creation of the game based on their instructions:

> *ChatGPT has been a great tool throughout our project. It was a great help in generating code and solving problems. Code generation based on specific instructions accelerated the creation of our game.*

Similarly, Group 5 found ChatGPT useful for establishing a foundational understanding of specific coding tasks. They noted:

> *At the beginning, we mainly used it to have a basis for certain parts of the code. We asked very general questions about how to do certain things, for example "how to size an image". The aim wasn't to generate all the code, but to find the right basic functions.*

Furthermore, Group 3 appreciated the ability of ChatGPT to transform their ideas into a tangible code base, especially helpful for beginners. They remarked:

> *Firstly, ChatGPT could give us a code base when we did not know how to write an idea in Python [...] especially as a beginner. By asking ChatGPT for a command in our own words, it provided us with a code example that we*

> *could then work on. It was easier this way than starting from scratch.*

These reflections demonstrate a consistent theme across the student groups: ChatGPT served as a valuable tool in the initial coding phases, offering foundational guidance and accelerating the development process.

5.2.3 ***Identifying errors and debugging.*** The third theme relates to how valuable ChatGPT was for debugging their code and speeding up their development process. Students across all groups agreed about the power of the GenAI agent in assisting them with their code. For instance, students from Group 3 wrote:

> *we used ChatGPT to debug our code. Sometimes, despite the instructions given on the console, we couldn't work out why our code wasn't working. ChatGPT turned out to be a great tool to help us find the error(s) that prevented what we'd just written from working properly.*

Similarly, students in Group 4 emphasized the tool's proficiency in handling complex code. They remarked:

> *ChatGPT is a valuable asset for quickly spotting errors in the code, offering a quick and effective solution, particularly when the code becomes complex.*

Student comments consistently underscored ChatGPT's ability to simplify the debugging process and enhance overall code efficiency and functionality.

5.2.4 ***Errors and risks of over-reliance.*** The third theme that appeared in students' comments covers risks and limitations associated with ChatGPT. In fact, students did not only perceive the tool positively as a valuable aid in building code and debugging, but also noted several of its limitations and shortcomings. For instance, Group 6 highlighted that ChatGPT can make errors and create additional bugs in their already functional code. They noted:

> *With how ChatGPT works, it will keep trying to answer even though it's not capable of giving a solution. This can result in the AI making a mistake (creating a bug) and when incapable to resolve it, modifying other parts of the code for the sake of answering, resulting in further issues.*

Group 4 also highlighted the potential for over-reliance, emphasizing the risk of dependency. They cautioned:

> *It's also important to recognize the risk of dependency. Even when dealing with simple tasks, it can be tempting to look directly on ChatGPT for solutions, neglecting the opportunity to deepen our comprehension and develop our individual skills. Maintaining a balance between using ChatGPT as a resource and solving problems independently is essential.*

Finally, Group 6 noted the challenges of effectively communicating with the LLM, emphasizing the importance of clarity and specificity in instructions. They reflected:

> *We had to learn how to communicate with it and it makes a lot of mistakes. What we learned is that it is vital before asking it to make a change, to be sure about what we want and how this change should affect the rest of the game. Giving specific conditions about how we want it to tackle the problem is crucial.*

5.2.5 ***Diminishing utility as the project advances.*** One of the important themes that emerged from analyzing student reflections is how ChatGPT became less useful to students as their projects reached more advanced levels. They expressed that when their code attained a certain level of complexity, ChatGPT was less helpful, and their reliance on it decreased. For instance, Group 6 noted:

> *We found its use more complicated as our game became more complex and we had to rely on ourselves more in the end.*

Similar experiences were echoed by Group 3, which further highlights the limitations of ChatGPT in handling complex coding challenges. They shared their frustration, stating:

> *In some cases, even ChatGPT couldn't find the problem preventing our code from running; this happened in cases where the code became a little more complex.*

Lastly, Group 2 mentioned that in addition to the shift in their reliance on ChatGPT as the project advanced, the response time became increasingly significant, triggering their impatience:

> *As the project advanced, our reliance on ChatGPT began to shift. Some of our team members have more experience in developing software, and found that ChatGPT has certain limitations for more advanced queries. The time taken by ChatGPT to generate responses, although informative, started to become a factor. As the project's complexity increased, our impatience grew with the response time.*

6 DISCUSSION

Our findings reveal that prompt-tuning GPT-3 using an optimized CT-based prompt resulted in a decrease in the tool's perceived ease of use and learning assistance. Additionally, the use of the prompt-tuned LLM degraded the student perceptions towards chatbots. One possible explanation for this is that students may have preferred to receive direct answers rather than being guided through a process to decompose and solve their programming questions. These results confirm findings from prior literature where the introduction of a context-specific prompt did not necessarily improve the participants' perceptions of GenAI use. For instance, the optimization of an LLM for particular software contexts based on prompt guidelines and domain context by Khurana et al. [31] did not seem to have a significant positive impact on the user perceptions of accuracy, relevance or trust. Similar results have been reported by Clapper et al. [12] where the customization of an LLM with positive tone and encouragement failed to make it perceived as highly supportive of a growth mindset by evaluators [15]. These findings provide insightful implications. Indeed, while the intention behind CT-based prompt-tuning is to enhance learning by guiding users through problem-solving processes, it may inadvertently lower user satisfaction by deviating from users' preferences for direct assistance, potentially leading to lower usage of the GenAI agent. The lower number of chats in the prompt-tuned version of the conversational agent might be a result of this decreased satisfaction, or alternatively, an indicator that students were using

the chatbot's guidance to work through problems on their own, lessening the need for further assistance. Future research could further investigate the factors leading to a decreased interaction with a CT-based prompt-tuned conversational agent. Nonetheless, these results imply that system designers should make sure to strike a balance between GenAI customization and user satisfaction in the design of AI conversational agents.

Conversely, our findings also showed that the introduction of a CT prompt led to an improvement in learning outcomes, It is worth noting that this result might have been influenced by variability in exercise difficulty across lab sessions despite employing an ABBA counterbalancing design to mitigate potential biases related to the labs' incremental nature and increasing difficulty. Nonetheless, the potential improvement in learning outcomes could be interpreted through the lens of *productive failure*, a concept suggesting that struggling through challenges without immediate solutions fosters a deeper understanding and critical thinking skills [54]. This result also aligns with the concept of *desirable difficulties* proposed by Bjork [4], suggesting that introducing certain hurdles during the learning process, such as those presented by the CT-based prompt, can enhance long-term retention and understanding, despite potentially hindering immediate performance or satisfaction. In this study, CT-based prompt-tuning was associated with higher learning outcomes, showing the potential of this approach in enhancing learning. This might have come as a result of compelling students to engage more deeply with the material, thereby achieving higher academic performance. However, future research is needed to confirm these hypotheses and further explore how prompt-tuning can leverage the learning outcome gains without compromising the user experience.

Results from our thematic analysis reveal that ChatGPT served as an effective initial coding assistant, enhancing efficiency and collaboration in the first phases of the project. These findings align with prior work, highlighting the extended capabilities of the LLM as an assistant in programming projects [45, 55]. Findings from this analysis also point out several limitations related to the use of the LLM, in alignment with previous literature, including errors [62] and the risk of over-reliance [48]. These results imply that employing GenAI must be used with caution, in order to leverage its advantages and circumvent its limitations. For instance, users could double-check GenAI's output for accuracy and avoid becoming overly reliant on ChatGPT for every aspect of their work.

Finally, findings from our thematic analysis also indicate that the utility of ChatGPT diminished over time as projects reached more advanced phases. This resonates with findings from Dell'Acqua et al. [14] indicating that for tasks that fall within the AI capability *frontier*, employees leveraging AI were significantly more productive, completing tasks more quickly and with higher quality. Conversely, for tasks that fall outside this frontier—becoming harder for AI to complete with a simple copy-pasting of instructions—employee performance decreased [14]. In the context of our study, this could mean that as projects increased in complexity, they might have moved from within to outside the frontier, thereby transitioning tasks from areas where AI could enhance productivity to areas where its support becomes less evident. However, it is not clear whether the reduced reliance on ChatGPT over time is due to its limitations becoming more apparent as project complexity increased, or whether it comes as a result of an improvement in student skills, leading to a decreased need for assistance from the tool. A controlled experiment could be useful in detecting if a GenAI conversational agent can only be used as a scaffolding [2] tool in the initial phases of a project, or if its support can extend consistently throughout the whole lifetime of a project. However, ethical considerations must be taken into account when conducting such an experiment in a classroom setting, to ensure that no students are disadvantaged or left behind.

6.1 Limitations

This study has some limitations that should be acknowledged. First, the study was conducted with a relatively small sample size of 21 students enrolled in specific programs, using a specific set of tools (Graasp, Replit). This limits the generalizability of the findings to other student populations, disciplines, and educational levels. Second, our study methodology did not allow us to conclusively discern whether the observed improvements in learning outcomes were driven by the effectiveness of the CT prompt, variations in lab difficulty, or both factors combined. However, the 'difficulty' of a CT exercise is a challenging property to operationalize, especially when exercises are incremental. This limitation should be further addressed in future research, possibly through a between-subjects experiment to isolate these variables more effectively. Additionally, the use of self-reported measures of usability and attitudes towards chatbots introduces subjectivity and potential response biases into the data [11]. Finally, while the study's semester-long time frame provides insights into user perceptions about ChatGPT use, it does not fully capture the long-term impact on students' attitudes and performance. Future studies could address these limitations by employing larger samples, exploring a variety of instructional settings and disciplines, and adopting more longitudinal designs.

7 CONCLUSION

In this paper, we conducted an experiment where we tested the impact of prompt-tuning GPT-3 using a CT-based prompt on usability metrics, perceived attitudes towards chatbots, and learning outcomes. Our results indicate that prompt-tuning the LLM resulted in decreased perceptions of ease of use and learning assistance. Additionally, it might have led to an overall degradation in students' attitudes towards chatbots. Conversely, using the prompt-tuned version of the LLM led to higher learning outcomes. Furthermore, our results highlight several strengths of using ChatGPT as an assistant in a semester-long project, such as helping with understanding, building, and debugging code. Finally, our study highlights several limitations associated with the use of LLMs in this context, including the generation of errors, the risk of over-reliance, and a diminishing utility over time. These findings open up avenues for future research that could explore how GenAI-enabled conversational agents can be leveraged to achieve a balance between customization, user satisfaction, and enhanced learning outcomes.

ACKNOWLEDGMENTS

This project was partially funded by Swissuniversities through the P8 project titled Transversal CT — supporting responsible computational problem-solving across domains.

REFERENCES

[1] Hayder Albayati. 2024. Investigating undergraduate students' perceptions and awareness of using ChatGPT as a regular assistance tool: A user acceptance perspective study. *Computers and Education: Artificial Intelligence* (2024), 100203.

[2] Brian R Belland, Andrew E Walker, Nam Ju Kim, and Mason Lefler. 2017. Synthesizing results from empirical research on computer-based scaffolding in STEM education: A meta-analysis. *Review of Educational Research* 87, 2 (2017), 309–344.

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

[4] Robert A Bjork. 1994. Memory and metamemory considerations in the. *Metacognition: Knowing about knowing* 185, 7.2 (1994).

[5] Ali Borji. 2023. A categorical archive of chatgpt failures. *arXiv preprint arXiv:2302.03494* (2023).

[6] Stefanie L Boyer, Diane R Edmondson, Andrew B Artis, and David Fleming. 2014. Self-directed learning: A tool for lifelong learning. *Journal of marketing education* 36, 1 (2014), 20–32.

[7] Virginia Braun and Victoria Clarke. 2012. *Thematic analysis.* American Psychological Association.

[8] Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2023. Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4. *arXiv preprint arXiv:2312.16171* (2023).

[9] William Cain. 2024. Prompting Change: Exploring Prompt Engineering in Large Language Model AI and Its Potential to Transform Education. *TechTrends* 68, 1 (2024), 47–57.

[10] Ünal Çakıroğlu and Mücahit Öztürk. 2017. Flipped classroom with problem based activities: Exploring self-regulated learning in a programming language course. JSTOR.

[11] David Chan. 2010. So why ask me? Are self-report data really that bad? In *Statistical and methodological myths and urban legends.* Routledge, 329–356.

[12] M. Clapper et al. 2023. Evaluating LLM's generation of growth-mindset supportive language in middle years math. In *Proc. Worksh. on Equity, Diversity, and Inclusion in Educational Technology Research and Development, 24th Int. Conf. on Artificial Intelligence in Education.* Artificial intelligence in education.

[13] Fabio Clarizia, Francesco Colace, Marco Lombardi, Francesco Pascale, and Domenico Santaniello. 2018. Chatbot: An education support system for student. In *Cyberspace Safety and Security: 10th International Symposium, CSS 2018, Amalfi, Italy, October 29–31, 2018, Proceedings 10.* Springer, 291–302.

[14] Fabrizio Dell'Acqua, Edward McFowland, Ethan R Mollick, Hila Lifshitz-Assaf, Katherine Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R Lakhani. 2023. Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. *Harvard Business School Technology & Operations Mgt. Unit Working Paper* 24-013 (2023).

[15] Dorottya Demszky, Diyi Yang, David S Yeager, Christopher J Bryan, Margarett Clapper, Susannah Chandhok, Johannes C Eichstaedt, Cameron Hecht, Jeremy Jamieson, Meghann Johnson, et al. 2023. Using large language models in psychology. *Nature Reviews Psychology* 2, 11 (2023), 688–701.

[16] Mahmoud Elkhodr, Ergun Gide, Robert Wu, and Omar Darwish. 2023. ICT students' perceptions towards ChatGPT: An experimental reflective lab analysis. *STEM Education* 3, 2 (2023), 70–88.

[17] Juan Carlos Farah, Sandy Ingram, Basile Spaenlehauer, Fanny Kim-Lan Lasne, and Denis Gillet. 2023. Prompting Large Language Models to Power Educational Chatbots. In *Advances in Web-Based Learning – ICWL 2023*, Haoran Xie, Chiu-Lin Lai, Wei Chen, Guandong Xu, and Elvira Popescu (Eds.). Vol. 14409. Springer Nature Singapore, Singapore, 169–188. https://doi.org/10.1007/978-981-99-8385-8_14

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

[19] Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines* 30 (2020), 681–694.

[20] Denis Gillet, Isabelle Vonèche-Cardia, Juan Carlos Farah, Kim Lan Phan Hoang, and María Jesús Rodríguez-Triana. 2022. Integrated Model for Comprehensive Digital Education Platforms. In *2022 IEEE Global Engineering Education Conference (EDUCON) (IEEE Global Engineering Education Conference).* IEEE, New York, NY, USA, 1586–1592. https://doi.org/10.1109/EDUCON52537.2022.9766795

[21] Sharon Gumina, Travis Dalton, and John Gerdes. 2023. Teaching IT Software Fundamentals: Strategies and Techniques for Inclusion of Large Language Models: Strategies and Techniques for Inclusion of Large Language Models. In *Proceedings of the 24th Annual Conference on Information Technology Education.* 60–65.

[22] Qi Guo, Junming Cao, Xiaofei Xie, Shangqing Liu, Xiaohong Li, Bihuan Chen, and Xin Peng. 2024. Exploring the potential of chatgpt in automated code refinement: An empirical study. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering.* 1–13.

[23] Joe Hair, Carole L. Hollingsworth, Adriane B. Randolph, and Alain Yee Loong Chong. 2017. An updated and expanded assessment of PLS-SEM in information systems research. *Industrial Management and Data Systems* (2017), 442–458. https://doi.org/10.1108/IMDS-04-2016-0130

[24] Md Asraful Haque. 2022. A Brief Analysis of "ChatGPT"–A Revolutionary Tool Designed by OpenAI. *EAI Endorsed Transactions on AI and Robotics* 1, 1 (2022), e15–e15.

[25] Justin O Holman. 2018. Teaching statistical computing with python in a second semester undergraduate business statistics course. *Business Education Innovation Journal* 10, 2 (2018), 104–110.

[26] 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.

[27] Jaeho Jeon and Seongyong Lee. 2023. Large language models in education: A focus on the complementary relationship between human teachers and ChatGPT. *Education and Information Technologies* (2023), 1–20.

[28] Ishika Joshi, Ritvik Budhiraja, Harshal Dev, Jahnvi Kadia, M Osama Ataullah, Sayan Mitra, Harshal D Akolekar, and Dhruv Kumar. 2023. ChatGPT in the Classroom: An Analysis of Its Strengths and Weaknesses for Solving Undergraduate Computer Science Questions. *arXiv preprint arXiv:2304.14993* (2023).

[29] Breanna Jury, Angela Lorusso, Juho Leinonen, Paul Denny, and Andrew Luxton-Reilly. 2024. Evaluating LLM-generated Worked Examples in an Introductory Programming Course. In *Proceedings of the 26th Australasian Computing Education Conference.* 77–86.

[30] 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.

[31] Anjali Khurana, Hari Subramonyam, and Parmit K Chilana. 2024. Why and When LLM-Based Assistants Can Go Wrong: Investigating the Effectiveness of Prompt-Based Interactions for Software Help-Seeking. *arXiv preprint arXiv:2402.08030* (2024).

[32] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, et al. 2016. Jupyter Notebooks-a publishing format for reproducible computational workflows. *Elpub* 2016 (2016), 87–90.

[33] Mika Koverola, Anton Kunnari, Jukka Sundvall, and Michael Laakasuo. 2022. General attitudes towards robots scale (GAToRS): A new instrument for social surveys. *International Journal of Social Robotics* 14, 7 (2022), 1559–1581.

[34] Shalom Lappin. 2023. Assessing the Strengths and Weaknesses of Large Language Models. *Journal of Logic, Language and Information* (2023), 1–12.

[35] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).

[36] Yicong Liang, Di Zou, Haoran Xie, and Fu Lee Wang. 2023. Exploring the potential of using ChatGPT in physics education. *Smart Learning Environments* 10, 1 (2023), 52.

[37] Lorelei Lingard. 2023. Writing with ChatGPT: An illustration of its capacity, limitations & implications for academic writers. *Perspectives on Medical Education* 12, 1 (2023), 261.

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

[39] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, and Jiebo Luo. 2023. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780* (2023).

[40] Reza Hadi Mogavi, Chao Deng, Justin Juho Kim, Pengyuan Zhou, Young D Kwon, Ahmed Hosny Saleh Metwally, Ahmed Tlili, Simone Bassanelli, Antonio Bucchiarone, Sujit Gujar, et al. 2024. ChatGPT in education: A blessing or a curse? A qualitative study exploring early adopters' utilization and perceptions. *Computers in Human Behavior: Artificial Humans* 2, 1 (2024), 100027.

[41] Dr Mahsa Mohaghegh and Michael McCauley. 2016. Computational thinking: The skill set of the 21st century. (2016).

[42] Marta Montenegro-Rueda, José Fernández-Cerero, José María Fernández-Batanero, and Eloy López-Meneses. 2023. Impact of the implementation of ChatGPT in education: A systematic review. *Computers* 12, 8 (2023), 153.

[43] Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. 2021. Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence* 2 (2021), 100033.

[44] OpenAI. 2023. *GPT-4 Technical Report.* https://cdn.openai.com/papers/gpt-4.pdf

[45] Abdessalam Ouaazki, Kristoffer Bergram, and Adrian Holzer. 2023. Leveraging ChatGPT to Enhance Computational Thinking Learning Experiences. In *Proceedings of the 2023 TALE Conference on Teaching Assessment and Learning for Engineering.*

[46] Kevin Peyton and Saritha Unnikrishnan. 2023. A comparison of chatbot platforms with the state-of-the-art sentence BERT for answering online student FAQs. *Results in Engineering* 17 (2023), 100856.

[47] Bhavika R Ranoliya, Nidhi Raghuwanshi, and Sanjay Singh. 2017. Chatbot for university related FAQs. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).* IEEE, 1525–1530.

[48] Partha Pratim Ray. 2023. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of*

Things and Cyber-Physical Systems (2023).

[49] Stephen Rice, Sean R Crouse, Scott R Winter, and Connor Rice. 2024. The advantages and limitations of using ChatGPT to enhance technological research. *Technology in Society* 76 (2024), 102426.

[50] Carlos Rodrigues, Arsénio Reis, Rodrigo Pereira, Paulo Martins, José Sousa, and Tiago Pinto. 2022. A Review of Conversational Agents in Education. In *International Conference on Technology and Innovation in Learning, Teaching and Education*. Springer, 461–467.

[51] Farhana Sethi. 2020. FAQ (frequently asked questions) chatbot for conversation. *Authorea Prepr* 8 (2020).

[52] Abdulhadi Shoufan. 2023. Exploring Students' Perceptions of CHATGPT: Thematic Analysis and Follow-Up Survey. *IEEE Access* (2023).

[53] Sharob Sinha, Shyanka Basak, Yajushi Dey, and Anupam Mondal. 2020. An educational Chatbot for answering queries. In *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*. Springer, 55–60.

[54] Tanmay Sinha and Manu Kapur. 2021. When problem solving followed by instruction works: Evidence for productive failure. *Review of Educational Research* 91, 5 (2021), 761–798.

[55] 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.

[56] Roland Tormey, Siara Isaac, Cécile Hardebolle, and Ingrid Le Duc. 2021. *Facilitating experiential learning in higher education: Teaching and supervising in labs, fieldwork, studios, and projects*. Routledge.

[57] Eva AM Van Dis, Johan Bollen, Willem Zuidema, Robert van Rooij, and Claudi L Bockting. 2023. ChatGPT: five priorities for research. *Nature* 614, 7947 (2023), 224–226.

[58] Jeanette Wing. 2011. Research notebook: Computational thinking—What and why. *The link magazine* 6 (2011), 20–23.

[59] Jeannette M Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.

[60] 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.

[61] Liqiao Xia, Chengxi Li, Canbin Zhang, Shimin Liu, and Pai Zheng. 2024. Leveraging error-assisted fine-tuning large language models for manufacturing excellence. *Robotics and Computer-Integrated Manufacturing* 88 (2024), 102728.

[62] Ramazan Yilmaz and Fatma Gizem Karaoglan Yilmaz. 2023. Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Computers in Human Behavior: Artificial Humans* 1, 2 (2023), 100005.

[63] Xiaoming Zhai. 2022. ChatGPT user experience: Implications for education. *Available at SSRN 4312418* (2022).

[64] Quanjun Zhang, Tongke Zhang, Juan Zhai, Chunrong Fang, Bowen Yu, Weisong Sun, and Zhenyu Chen. 2023. A critical review of large language model on software engineering: An example from chatgpt and automated program repair. *arXiv preprint arXiv:2310.08879* (2023).

[65] Gaoxia Zhu, Xiuyi Fan, Chenyu Hou, Tianlong Zhong, Peter Seow, Annabel Chen Shen-Hsing, Preman Rajalingam, Low Kin Yew, and Tan Lay Poh. 2023. Embrace Opportunities and Face Challenges: Using ChatGPT in Undergraduate Students' Collaborative Interdisciplinary Learning. *arXiv preprint arXiv:2305.18616* (2023).