

How Novice Programmers Use and Experience ChatGPT when Solving Programming Exercises in an Introductory Course

Andreas Scholl
Computer Science
Nuremberg Tech
Nuremberg, Germany
andreas.scholl@th-nuernberg.de

Natalie Kiesler
Computer Science
Nuremberg Tech
Nuremberg, Germany
natalie.kiesler@th-nuernberg.de

Abstract—This research paper contributes to the computing education research community’s understanding of Generative AI (GenAI) in the context of introductory programming, and specifically, how students utilize related tools, such as ChatGPT. An increased understanding of students’ use is mandatory for educators and higher education institutions, as GenAI is here to stay, and its performance is likely to improve rapidly in the near future. Learning about students’ use patterns is not only crucial to support their learning, but to develop adequate forms of instruction and assessment. With the rapid advancement of AI, its broad availability, and ubiquitous presence in educational environments, elaborating how AI can enhance learning experiences, especially in courses such as introductory programming is important. To date, most studies have focused on the educator’s perspective on GenAI, its performance, characteristics, and limitations. However, the student perspective, and how they actually use GenAI tools in course contexts, has not been subject to a great number of studies. Therefore, this study is guided by the following research questions: (1) What do students report on their use pattern of ChatGPT in the context of introductory programming exercises? and (2) How do students perceive ChatGPT in the context of introductory programming exercises? To address these questions, computing students at a large German university were asked to solve programming tasks with the assistance of ChatGPT as part of their introductory programming course. Students (n=298) provided information regarding the use of ChatGPT, and their evaluation of the tool via an online survey. This research provides a comprehensive evaluation of ChatGPT-3.5’s application by novice programmers in a higher education context. The findings reveal that while students widely adopt GenAI, their use varies significantly, ranging from acceptance of generated solutions to dynamic, and critical engagement. Therefore, this work has implications for educators designing guardrails or forms of instructions on the use of GenAI tools in the classroom.

Index Terms—ChatGPT, generative AI, large language models, students, application, introductory programming

I. INTRODUCTION

While educators and researchers are still trying to grasp the full impact of Generative Artificial Intelligence (GenAI) and related tools on our educational system(s), students are quickly moving forward, and open to experimenting with Copilot, and ChatGPT [1]–[3]. Even though it is challenging to predict which programming competencies [4] students will require in

the future and how to adapt curricula respectively [5], [6], computing educators need to gain an understanding of how students currently apply GenAI tools.

There is no doubt that GenAI tools, such as ChatGPT, are powerful and performing well in the context of computing, and particularly in introductory programming courses. An increasing body of research has been dedicated to showing its potential in solving CS1 and CS2 assignments [7]–[9]. Another use case for students is the enhancement of programming error messages [10]–[13], and the generation of formative feedback [14]–[18]. Generating personalized feedback can potentially benefit both students, and educators, especially in large scale classes, such as introductory programming [19], [20]. At the same time, GenAI tools still produce misleading information for novice learners [15].

At this point, however, hypothesizing about students’ use cases is a debate mostly led by educators [21]–[23], as they are reflecting and discussing the alignment of their learning objectives, instruction, learning activities and assessments in the GenAI era. So far, few studies have investigated GenAI tools from the student perspective (e.g., trust or attitude towards LLMs [24], [25]), or how students use such tools in the context of a computing course [26], [27]. Due to the broad availability of GenAI tools, and their well-known limitations, it is crucial to investigate how and for which purposes computing students use such tools, and how they perceive the benefits and limitations of GenAI.

It is thus **the goal** of this research to gather the student perspective on using GenAI tools, such as ChatGPT. This perspective encompasses use pattern, but also students’ reflections upon their experiences with ChatGPT in the context of a curricular course. To gather the student perspective, we developed an exercise sheet as part of an introductory programming course at a large, public university (Goethe University Frankfurt, Germany) in the winter term 2023/24. Students were asked to solve programming tasks with the assistance of ChatGPT-3.5 (freely available in Dec. 2023), and to participate in an online survey to report their experiences.

The quantitative and qualitative analysis of the survey

responses (n=298) provides novel insights into students' use of ChatGPT in a curricular course setting. The **contribution of this paper** is a summary of students' use patterns, and reflections on applying ChatGPT while solving programming exercises. The presented research thereby supports educator's understanding of students' applying GenAI tools. This is crucial to help inform and develop adequate forms of instruction and pedagogical approaches supporting the critical and reflective use of GenAI tools within computing education.

II. RELATED WORK

GenAI and related tools have taken the world by storm [3]. Due to the broad availability of GenAI tools, starting with the launch of OpenAI's ChatGPT in late November 2022, computing educators are discussing implications for higher education institutions, curricula, learning objectives, assignments, and assessments ever since [3], [5], [28]. In this section, we summarize benefits and limitations of GenAI tools in introductory programming. Moreover, we present recent findings on computing students' perspective on GenAI tools.

A. Potential Benefits and Limitations of GenAI in Introductory Programming Contexts

The performance of GenAI tools can be impressive, especially in introductory programming exercises and respective exams [29], [30]. Several studies showed the capabilities of tools based on Large Language Models (LLMs), such as Copilot and ChatGPT [7]–[9]. For example, Geng et al. [7] treated ChatGPT as a student participating in an introductory-level functional language programming course. The tool achieved a grade B and was ranked as 155 out of 314 students. Kiesler and Schiffner [8] had ChatGPT 3.5 and GPT-4 generate solutions to 72 standard introductory programming tasks available on CodingBat [31]. ChatGPT-3.5 immediately solved the task correctly in 69 cases (95,8%), GPT-4 in 68 tasks (94,4%) [8]. Savelka et al. [9] showed that GPT models are more successful when prompted in natural language, or when asked to fill in blanks. Multiple-choice questions requiring the analysis or reasoning about code were more challenging for the models [9].

Despite these promising results, many studies have reported limitations of GenAI tools, such as (re-)producing biases or inaccurate information [32]. Another obvious problem is its knowledge cutoff so the tools lack recent data, e.g., on the latest Python version, which may cause irritations. Hallucinations within the generated data, and the fact that it can bypass plagiarism detection tools are additional issues for students and educators. As GenAI tools challenge the integrity of academic work, educational institutions and faculty need to take action on several levels [3], [33]. Prather et al. [3] emphasize the need for ongoing research to keep up with the fast pace of technological developments.

In the context of introductory programming education, GenAI tools have been subject to research w.r.t. their application by educators and students [11], with a focus on students' potential use [10], [12], [15], [17], [34]–[36]. One of these use cases is the generation of code explanations

in a web software development e-book [12]. MacNeil et al. [12] analyzed students' perceptions towards automatically generated line-by-line code explanations by Codex and GPT-3. The majority of students evaluated as the explanations as helpful [12]. Leinonen et al. [10] found that programming error messages (i.e., by the compiler and interpreter) can be enhanced by GenAI, as they can help create novice-friendly, actionable explanations.

GenAI tools were also shown to analyze students' code, and fix errors [35], [36]. Phung et al. [35] studied the application of LLMs with the goal of fixing syntax errors in Python. They developed a technique to receive high precision feedback [35]. In other studies, it was shown that GenAI tools are capable of generating individual, elaborate feedback [37] to students' solutions to programming exercises [15]–[17].

For example, Kiesler et al. [15] qualitatively analyzed the feedback generated by ChatGPT-3.5 in response to authentic student solutions to introductory programming exercises. In their study, they identified stylistic elements, textual explanations of the cause of errors and their fix, illustrating examples, meta-cognitive and motivational elements. However, ChatGPT generated misleading information, claimed uncertainty in its responses, and requested more information, thereby indicating its limitations, especially for novice learners [15].

Similarly, Azaiz et al. [16] analyzed the output generated by ChatGPT in response to introductory programming exercises and student solutions. They summarize a number of challenges, e.g., w.r.t. the formatting of the output, recognizing correct solutions, and hallucinations of errors in the students' code. In a follow-up study with GPT-4 Turbo, Azaiz et al. [17] conclude notable improvements of the generated feedback, as the outputs were more structured, consistent, and always personalized. Due to this potential, we currently see the development of new educational tools and environments incorporating GenAI tools to create novice-friendly code explanations [38], or to not give the solution away [39].

B. Computing Students' Perspective on GenAI tools

All of the aforementioned studies focused on students' potential use of GenAI tools, and thus hypothetical application scenarios. Some of them used authentic student data (e.g., students' solutions) as an input to GenAI tools [15], [17]. However, few studies directly involved students, their perspective, or their actual use of GenAI.

In a study by Prather et al. [40], students' use of GitHub Copilot as part of an introductory programming assignment was evaluated through observations and interviews. They asked students about their perceptions of the tool's usability, challenges and benefits, and implications for the tool's design. Vaithilingam et al. [41] also explored the usability of Copilot via user study with 24 participants. The participants appreciated the ability of Copilot to provide a useful starting point, and help find information (instead of having to search for it online). At the same time, programmers reported challenges related to understanding, editing, and debugging code snippets

generated by Copilot. Overall, they conclude that Copilot’s design should be improved [41].

The student perspective on program synthesizers including Copilot was gathered by Jayagopal et al. [42], who focused on its learnability. They observed and interviewed 22 novice programmers who worked with the respective systems, and evaluated the systems’ design. Students, for example, reported that prompting is perceived as less confusing and more exciting than writing format specifications [42].

Denny et al. [43] also investigated first year students’ interactions (n=54) with the tool *Promptly*. Promptly was designed to teach students how to prompt GenAI tools based on predefined problems. It executes and tests the code generated by an LLM. In the respective pilot study, Denny et al. [43] quantified students’ interactions with ChatGPT, analyzed their interactions, and their reflections on the use of GenAI. However, only 15 chat protocols were analyzed. Moreover, the character of the pilot was exploratory with the goal of evaluating the relevance of prompting skills for students [43]. The focus was thus not on the identification of interaction pattern or use cases.

An observation of 20 experienced programmers and an analysis of their interactions with Copilot led to the distinction of two operation modes: acceleration and exploration mode [44]. Copilot is thus either used to achieve a goal faster, or to explore different solution options. Barke et al. [44] further conclude that over-reliance on Copilot can hinder task completion, and that multiple suggestions can lead to a high cognitive load.

In a recent controlled experiment, ChatGPT’s application in a CS1 classroom was investigated via the evaluation of 56 students’ task performance, interactions with, and reflection upon using ChatGPT [45]. The study did not find evidence for ChatGPT’s impact on students’ performance, but students who used ChatGPT tended to solely rely on it (instead of exploring other resources). Students further seemed neutral towards ChatGPT, but they noted potential ethical issues, and varying performance depending on the tasks [45].

Students’ (and instructors’) perceptions towards GenAI in academia were also surveyed by Amani et al. [1]. The student survey focused on the current use of GenAI tools and their general perceptions. 813 students at Texas A&M university responded to the survey. As a result, students seem to value the high availability of the tools, while recognizing their potential to be misused and that academia must adapt. Forman et al. conducted a similar online survey exploring high school students’ perceptions towards ChatGPT [46]. The analysis of 71 responses revealed students’ positive attitude towards GenAI technology, as they are assuming it to play a role long term. Moreover, students reported to rely on GenAI tools to help save time during assignments and projects. Raman et al. [2] investigated university students’ adoption of ChatGPT by addressing the attributes relative advantage, compatibility, ease of use, observability, and trialability. The analysis of 288 student responses revealed that all five factors influence students’ intentions to use ChatGPT. The last example of a

student survey is the study conducted by Prather et al. [3] in 2023. They gathered 171 computing student responses from 17 countries, about half of them being in their first year of studies. Most students reported using GenAI for text or code generation, paraphrasing or summarizing text, and debugging. Some students, however, refuse to use GenAI tools due to their risks and ethical concerns. Students’ perceptions regarding the relevance of GenAI for their future careers were mixed [3].

C. Research Gap

Despite the rapid increase of studies focusing on the application of GenAI in computing education and, specifically, by novice learners of programming [12], [13], [15], [17], many of these studies are controlled, or define certain scenarios. They thus represent the educator’s perspective or focus on usability and design aspects, and are not necessarily authentic for the context of introductory programming exercises and courses. At the same time, the studies surveying students broadly address university and high school students’ perception towards GenAI tools [1]–[3], [46] but lack specific educational contexts. The student perspective on using GenAI for programming exercises while enrolled in a curricular programming course at university level is thus not yet investigated and well understood.

III. METHODOLOGY

In this section, we outline the research questions and goals guiding the present study. In addition, we introduce the survey design, the context of the data collection within an introductory programming course, and present the data analysis method.

A. Research Questions and Goals

The present study has the goal to gather the perspective of novice programmers using ChatGPT in the context of introductory programming exercises. Specifically, we are addressing the following research questions (RQs):

1. *What do students report on their use patterns of ChatGPT in the context of introductory programming exercises?*
2. *How do students perceive ChatGPT in the context of introductory programming exercises?*

To answer these questions, the study leverages empirical data collected in a survey with computing students (n=298) enrolled in an introductory programming course at a large German university. Before completing the survey, students were asked to complete a set of programming exercises with the assistance of ChatGPT-3.5.

B. Survey Development

The survey design was guided by the RQs and informed as far as possible by prior work. Most importantly, we wanted to gather students’ use patterns when working on the exercises as part of the course (RQ1). Therefore, we asked students how often, how long and for which purposes they use ChatGPT in the problem solving process. For the latter we used some of the application scenarios identified in related work (e.g., [3], [10], [12], [16], [35], [41]).

Similarly, the questions focusing on students' perceptions of ChatGPT (RQ2) were informed by prior work, for example, regarding ChatGPT's usability [40]–[43]. Therefore, our survey addresses the concepts *ease of use*, *perceived skills gain*, *accuracy and relevance of the responses*, and *user satisfaction*. The survey also incorporates open-ended questions to allow participants to provide more detailed insights regarding their experiences and perceptions, both positive and negative.

Students' prior programming experience was the only demographic question we considered relevant. Even though the course focuses on novice learners as a target group, some students do have prior programming experience due to, for example, vocational training. Finally, the survey was developed with clear and unbiased language to ensure comprehensibility and accuracy in the responses. The survey questions are available in Appendix A.

C. Course Context of the Data Collection

The present study was conducted within an introductory programming course for first-year computing students at Goethe University, Frankfurt. 790 students were enrolled in that course in the winter term 2023/24. Most students were enrolled in the computer science (CS) program, even though some students pursue CS as a minor. The course is dedicated to novice learners of programming, as there are no prerequisites to participate. It is accompanied by an online course in the university's learning management system Moodle.

The class consists of a 2-hour lecture per week for all students, and a 2-hour tutorial session in groups of 20-30 students. In the tutorial, students are supposed to solve weekly or bi-weekly exercise sheets with programming problems. Even though this is voluntary, students can collect points for correct submissions (two for the exercises required for this study). These points can later be used to improve exam grades.

For this study, a new exercise sheet was developed for one of the tutorial sessions. Students were asked to individually work on these exercises for two weeks from December 6 onwards. Group work was not allowed. Students were instructed to "complete the tasks using ChatGPT via the free version" (3.5) on the web interface, and to submit "all prompts and responses" as paired entries in a spreadsheet via the Moodle course. Students were not instructed on how to use ChatGPT-3.5. They were only provided with a link to OpenAI's guide on prompt engineering [47]. This approach was used to avoid influencing students' use of the tool. They had only been introduced to the study's objective and the process during the lecture preceding the tutorial.

D. Task Design

The new exercise sheet addressed several concepts, e.g., *recursion*, *functions*, *lists*, *conditionals*, *string manipulation*, and *documentation*. It comprised two tasks with sub-tasks.

Task 1 consisted of four sub-tasks. It presented students with code snippets with recursive elements for several operations:

- 1a summation of the digits of a number,
- 1b reversing a list,

- 1c performing multiplication, and
- 1d computing the Ackermann function.

The task was to read and interpret the given code snippets with the goal of determining the output of the code, the number of function calls, and identifying the type of recursion.

Task 2 was designed to be more complex. It required the implementation of a function that determines the number of "happy strings" within all sub-strings of a given string. A "happy string" was defined as a string that can either be rearranged into (or already is) a repetition of some string. In particular, students were asked to:

- 2a test a given string for the "happy" property, and
- 2b test all possible sub-strings of the given string.

Using the recursive approach was awarded with extra points.

An important criteria for the selected tasks was the performance of ChatGPT-3.5 in solving them. For example, ChatGPT demonstrated a high success rate in identifying the correct functions for the first task. However, it frequently failed in calculating the number of calls, as it often overlooked the base case. Task 2 proved to be even more challenging, and ChatGPT-3.5 never generated a correct solution in 10 regeneration attempts. It was only able to describe the necessary steps to solve the problem.

E. Data Analysis

To evaluate students' self-reported use patterns and perception of ChatGPT-3.5 in the context of solving selected programming exercises, 298 valid survey responses were analyzed. A quantitative approach was used to evaluate closed questions (Q1-Q11), while responses to open questions were qualitatively analyzed (Q12-Q14).

The survey addresses RQ1 (students' use patterns) within Q1-Q6, which are all closed questions. The quantitative analysis comprises students' programming experience prior to the introductory course in years, whether they used ChatGPT before this exercise, usage frequencies and duration, how ChatGPT was usually accessed, and for which tasks or problems it was consulted (all in the context of solving the exercise sheet).

RQ2 on students' perceptions and user experience is addressed by survey questions Q7-Q14. Questions 7-11 refer to the ease of use of ChatGPT and other adoption criteria. The evaluation is based on the responses on the 5-point-Likert scale to indicate students' distribution and the degree to which they agree, or were satisfied. The remaining open questions (Q12-14) were qualitatively analyzed by applying a content analysis [48] of the responses. As each response contained multiple meaningful elements (mostly 3, as requested by Q12 and 13), each positive or negative aspect was coded once. Each student's full response to the respective open survey questions was used as context unit.

Deductive categories reflecting students' perspective as shown in related work (see Section II) constituted the starting point for building a category system (e.g., ease of use, code explanations, use as study buddy, or for debugging). Due to the large number of responses to the open questions, and the variety of aspects, new inductive categories were built by

using the psychology of text processing [49], [50]. Thus, the responses were summarized, paraphrased and abstracted so that new categories were constructed based on the analyzed material. Developing the inductive categories was an iterative process. After coding an initial proportion of the material (approx. 10%), the initial categories were adapted based on students' responses. This procedure was repeated multiple times until all responses were analyzed.

IV. RESULTS

This section presents the students' responses to the survey questions (available online [51]) with the goal of answering both RQs. Students' reported use patterns of ChatGPT (RQ1) will be summarized, before introducing their perceptions of ChatGPT in the context of introductory programming exercises (RQ2). As RQ2 was addressed by closed and open questions in the survey, we outline quantitative and qualitative results.

A. Students' reports on their use patterns of ChatGPT (RQ1)

Students reported several insights into their use patterns of ChatGPT when solving the programming exercises for the introductory course. The analysis of survey questions Q1 to Q6 (n=298 each) provides a comprehensive snapshot as an answer to RQ1.

First of all, it should be noted that the surveyed students were indeed mostly programming novices (Q1), with 34% having no programming experience, and 43% having limited experience of less than one year. Another 17% of the students reported to have one to two years of programming experience before enrolling in the course, and 6% of the students had more than 3 years of programming experience.

When asked about students' use of ChatGPT prior to the given exercise (Q2), 84% reported to have used ChatGPT for assignments. Only 16% negated that question. Thus, a majority of the students in the course had little to no programming experience, yet nearly most of them had previously used ChatGPT for their coursework. When it comes to the frequency (Q3) of using ChatGPT for their programming exercises, the data suggests that about half of the students (52%) engage with the GenAI tool on a weekly basis, 18% even daily. 6% reported on using it monthly, 16% do so rarely, and only 8% of student have never used ChatGPT.

Regarding duration (Q4), 43% seem to use ChatGPT for quick requests lasting less than 15 minutes, while 38% use it for 15-30 minutes. The remaining students use it for 30-60 minutes (16%), or for more than 60 minutes (3%). Students clearly prefer using the ChatGPT's interface, with 95% of the respondents indicating this approach. Only 2%, respectively 3% use the messaging App integration or the API integration.

Students further selected all tasks for which they applied ChatGPT when working on the given programming exercises (Q6). As shown in Figure 1, students used the tool for multiple purposes, ranging from problem understanding (223), conceptual understanding (178), code generation (176), debugging

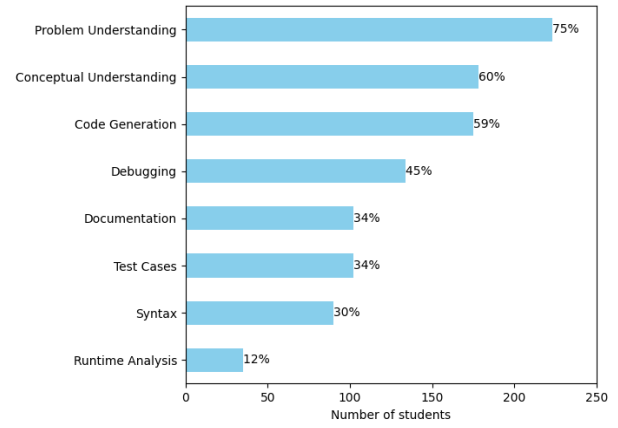


Fig. 1. Responses to Q6: Tasks for which students used ChatGPT.

(134), producing documentation (102) and test cases (102), correcting syntax (90), to performing a runtime analysis (35).

In addition to the closed answer options, students provided the following applications of ChatGPT: used for coding (13), to get general explanations (9), as a writing assistant (8), but also for private uses (7), as study buddy (4), search tool (4), or as a starting point when solving programming tasks (3).

B. Students' perceptions of ChatGPT (RQ2)

The second part of the survey quantitatively and qualitatively explored the subjective experiences and evaluation of the GenAI tool as reported by learners (Q7-Q14).

1) *Quantitative results:* The diverging stacked bar chart in Figure 2 illustrates the distribution of student responses to several 5-point-Likert-scale questions (Q7-Q11, n=298 each) about their perspectives on using ChatGPT. Each row of the chart corresponds to a different question, with responses ranging from negative (left-hand side, shades of red) to positive (right-hand side, shades of blue), and a neutral midpoint (grey).

For Q7 on the ease of using ChatGPT, the median and mean of the responses were 4 and 4.06 respectively. The majority of responses were positive, with the largest proportion of students rating ChatGPT the highest on the Likert scale, indicating a perception of ChatGPT as user-friendly.

When asked about the extent to which ChatGPT has helped improve programming skills or solve coding problems (Q8), the responses were generally positive, with a median of 3 and a mean of 3.40. A significant number of students felt that ChatGPT has been helpful in these areas.

The accuracy and relevance of the responses provided by ChatGPT (Q9) yielded a more varied distribution of opinions, with a median of 3 and a mean of 2.87. These statistics suggest some students found the accuracy and relevance to be less than satisfactory, leaning towards the lower end of the scale.

Regarding overall satisfaction with ChatGPT for programming assistance (Q10), the median was 4 and the mean was 3.44, showing a positive trend. Many students expressed high levels of satisfaction with the tool.

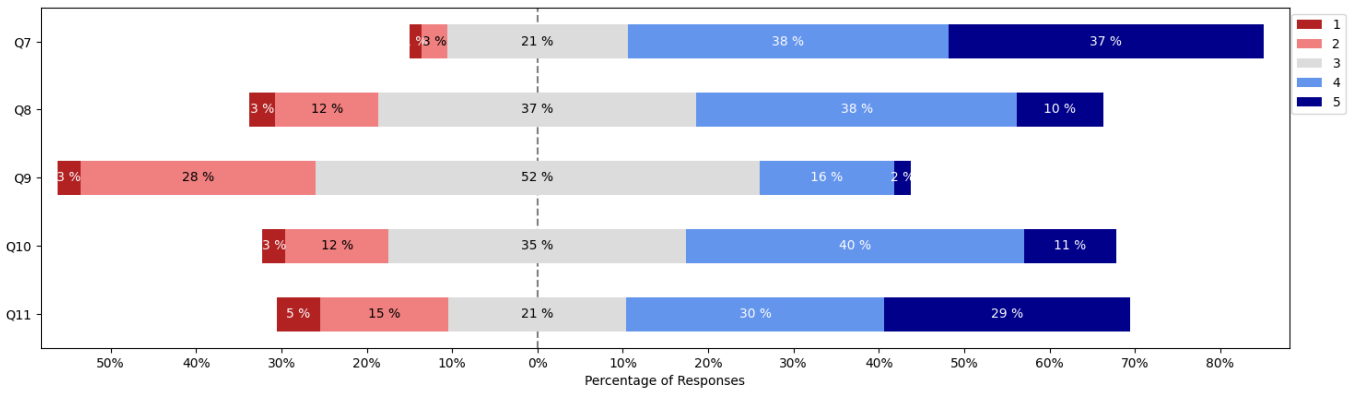


Fig. 2. Likert scale assessment of student impressions on using ChatGPT (Q7-Q11). The figure displays a diverging stacked bar chart representing student ratings on ease of use, assistance in skill improvement, accuracy and relevance of responses, overall satisfaction, and likelihood to recommend ChatGPT as a learning aid. Responses range from negative (1, dark red) to positive (5, dark blue), with a neutral central point (grey).

Lastly, the likelihood of recommending ChatGPT as a support tool to a programming novice (Q11) was predominantly positive, with a median of 4 and a mean of 3.63. A substantial portion of students indicated they are very likely to recommend ChatGPT for programming novices.

Across all questions, there is a substantial number of responses in the neutral category, reflecting a degree of ambivalence or moderate views about ChatGPT among the students.

2) *Qualitative results*: The qualitative analysis of the open questions further reflect upon students' perceptions towards ChatGPT when solving programming exercises in an introductory course. Table I summarizes the deductive-inductive categories of positive and negative aspects and experiences students reported in response to Q12 (n=261) and Q13 (n=259). The analysis resulted in 740 coding units with positive comments (Q12), and 682 coding units representing negative perspectives (Q13) – 1422 in sum.

The first column of the table represents the category label, and how often students referred to it in the data, both in the positive (\oplus), and negative (\ominus) sense. The second column provides the category definition, which is followed by a literal, representative example of a student response. Overall, we identified 20 categories in the responses to Q12 and Q13.

The initial seven categories of Table I represent students' perspectives on several different use patterns. Among them is the use as initial help tool when starting to solve a problem, the application as a search tool for dedicated questions, or for conceptual input. Many students referred to using ChatGPT for the generation of code and text (e.g., test cases, documentation, etc.), but noted both positive and negative aspects of that scenario. Debugging is another, mostly positively mentioned use case. ChatGPT was also mentioned as producing alternative perspectives and solutions, even though students disregarded the degree of variation of the replies. Another positively mentioned use case is that of the study buddy, meaning students use ChatGPT to generate individual, on-demand assistance.

The remaining 13 categories presented in Table I summarize students' perceptions of certain qualities or aspects related to

the use of ChatGPT. For example, students noted the response quality with a positive and negative connotation, e.g., due to its ability to rephrase things, but also w.r.t. its use of uncommon/unknown terms. Similarly, the chat history can help students, as they do not have to repeat themselves in a conversation. At the same time, 'remembering' the conversation can require students to restart a new chat. The tool's availability is mostly mentioned in a positive manner, with downtime being criticized. The ease of using ChatGPT comprises positive aspects, such as its intuitive handling, adapting to natural language, multilingual inputs, and even vague inputs. At the same time, students are under the impression that precise inputs are required.

ChatGPT's efficiency w.r.t. time was perceived as positive, due to the potential of saving time, and negative due to the need to verify its responses. Students further appreciated the large knowledge base, but criticized the biases it can reproduce. Another aspect concerns the perceived 'social' interaction with ChatGPT. On the one hand, the safe space without 'stupid' questions was appreciated. On the other hand, students noted that the tool does not comprehend emotions.

All of the aforementioned aspects were recognized in a sort of balanced manner, expressing both benefits for students, but also criticism. However, 6 of the 13 categories only refer to negative aspects students experience using ChatGPT. Among them are privacy concerns, ChatGPT's overconfidence despite producing incorrect results, hallucinations, and a lack of integrity (e.g., by citing non-existent sources). Moreover, students expressed the necessity to remain critical of the GenAI tool, and to always verify the generated responses. Some students (26) even mentioned the risk of depending too heavily on ChatGPT, causing them to not thoroughly attempt to solve programming exercises themselves anymore.

The last survey question (Q14) asked for additional remarks (n=136). Students mostly repeated the positive and negative aspects from Q12 and Q13. Nonetheless, we present some of the most interesting comments to highlight students' perspectives:

"AI (and how to use it) should be introduced as a core

TABLE I
CATEGORIES, DEFINITIONS, AND ANCHOR EXAMPLES REFLECTING STUDENTS' PERCEPTIONS (RQ2).

Category	Definition	Anchor Example(s)
Starting Point ⊕45	Refers to initial guidance and structure provided by ChatGPT, including code templates.	⊕“Provides at least an approach when one has no idea where to begin.”
Conceptual Input ⊕59	Refers to ChatGPT's ability to provide new concepts, ideas and inspiration, including suggestions for functions, libraries and hints.	⊕“If one suspects that there is a certain command or built-in function that most likely exists but does not know it.”
Study Buddy ⊕23	Refers to ChatGPT as companion-like support tool providing personalized guidance, tutoring, and interactive assistance.	⊕“Offers instant help with coding problems and advice on best practices, acting like an on-demand mentor.”
Code/Text Generation ⊕89 ⊖111	Refers to ChatGPT's ability to produce, transform, or finalize various forms of content, such as code, text, tests or documentation.	⊕“I can work well with the code that ChatGPT provides.” ⊖“More complex code is almost unusable.”
Debugging ⊕85 ⊖5	Refers to assisting with troubleshooting and refining code input by identifying and addressing compile, runtime, syntax and logical errors.	⊕“ChatGPT is very good at identifying minor errors in the code ” ⊖“When reviewing, ChatGPT itself creates mistakes.”
Alternative Perspectives and Solutions ⊕18 ⊖18	Refers to ChatGPT's capacity to offer students a range of opinions, perspectives, and approaches to programming tasks; includes providing alternative solutions and varied rephrasing of questions or answers.	⊕“A different perspective on the task (different code).” ⊖“Often provides incorrect answers or different answers to the same question upon repeated input.”
Search Tool ⊕29 ⊖1	Refers to use of ChatGPT as a resource for gathering information and finding answers, similar to search engines and online forums.	⊕“It shortens my time spent searching for information on the internet and helps me avoid wasting time looking for unnecessary information.” ⊖“The outputs often do not go beyond Google search results.”
Response Quality ⊕146 ⊖142	Refers to the level of accuracy, clarity, comprehensibility and usefulness as well as structure and format of ChatGPT's answers to students' questions.	⊕“If a task is not entirely clear, ChatGPT can rephrase it differently.” ⊖“Sometimes the content is not explained well, and strange terms and/or explanations are used.”
Chat History ⊕27 ⊖79	Refers to ChatGPT's ability to retain previous interactions, enabling users to revisit and correct prior queries and responses.	⊕“The program learns during the conversation [...] the context does not need to be re-explained with each question.” ⊖“ChatGPT often needs to be restarted because it remembers 'answers'.”
Availability ⊕84 ⊖35	Refers to prerequisites required to access ChatGPT (i.e., account/subscription) available servers, and to ChatGPT's response times.	⊕“It is quick [...] A large amount of code can be generated. It is free.” ⊖“ChatGPT is often down and unusable.”
Ease of Use ⊕67 ⊖119	Refers to the level of accessibility and user-friendliness; includes the ability to adapt to natural language inputs, and the structure, language and precision of instructions.	⊕“Even vaguely stated questions were explained thoroughly and precisely.” ⊖“The AI often fails when it is not provided with enough data. Precise formulation is necessary.”
Time Efficiency ⊕29 ⊖10	Refers to the potential of ChatGPT to save time by providing quick answers and reducing workload, despite having to verify responses.	⊕“Routine tasks are completed quickly.” ⊖“Slower for more complex problems than writing the code oneself.”
Knowledge Base ⊕31 ⊖29	Refers to range and scope of available information based on training data; includes versatility, relevance and to-dateness of the information.	⊕“ChatGPT is based on a large training dataset.” ⊖“It is biased on non-purely technical topics.”
Social Interaction ⊕8 ⊖9	Refers to the manner in which ChatGPT engages with students, focusing on the conversational aspects and the overall experience of communication.	⊕“With ChatGPT, there are no stupid questions; it is a safe space.” ⊖“The model may not understand emotions well and could provide inappropriate responses in emotional situations.”
Privacy Concerns ⊖5	Refers to the assumptions students may have about the confidentiality/privacy of their interactions with ChatGPT.	⊖“I am constantly concerned that my messages are read by third parties and always feel an underlying unease when using it.”
Overconfidence ⊖29	Refers to ChatGPT's presenting information with a high level of certainty, even when the data may be incorrect or questionable.	⊖“Questions can be just outright wrong, with the language model insisting it was correct.”
Hallucinations ⊖16	Refers to ChatGPT generating responses that include inaccurate or fabricated information, such as citing non-existent sources.	⊖“I once asked for literature recommendations. The responses seemed credible, but they were entirely fabricated.”
Lack of Integrity ⊖12	Refers to clarity and reliability of the provided information, including references to credible sources ensuring academic integrity.	⊖“Unverifiable or no information about sources.”
Criticizing GenAI ⊖36	Refers to students having to critically evaluate and verify ChatGPT's responses.	⊖“One must be able to judge whether the solution is correct or not. ChatGPT should only be used if one already has basic programming knowledge.”
GenAI dependency ⊖26	Refers to risk for students to depend too heavily on ChatGPT for programming exercises.	⊖“When I have a problem, I don't think about it for as long as I used to, I just go and ask a ChatGPT. It made me depend on it.”

module. If you know how to use it, you learn much faster and save a lot of energy.”

“If in the exercises the idea of how the code came up and the attempt to write it were more valued (instead of the result ‘the code runs well’), the process of doing the exercises might be more enjoyable.”

“I don't think it impacts my learning experience positively. Of course it helps to get answers fast and in my case mostly (if not all) right. But a lot of answers that stick to my head today are the ones I have researched very long.”

V. DISCUSSION

The analysis of the responses provided by students in the survey reveal that many students incorporate ChatGPT into their routines when solving programming tasks, utilizing it for a diverse range of problems. Some of these had been identified in prior work, e.g., debugging [3]. While many aspects of ChatGPT are perceived as helpful and useful by students, there is also notable criticism surrounding known issues such as hallucinations, overconfidence, and over-reliance.

Although some students voiced criticism regarding the occasional unavailability of ChatGPT, many valued the opportunity to receive immediate responses, along with appreciating the interactive nature of the platform. In addition, students positively highlighted the multilingual capabilities of ChatGPT and the ease of its use. They found the ability to seamlessly switch between their native language, German, and English to be particularly advantageous.

Many students expressed difficulties in understanding programming tasks and getting started with writing code. Hence, they valued the use of ChatGPT as starting point, similarly to the students surveyed in related work [41]. While some students indicated that ChatGPT provided satisfactory responses to straightforward prompts despite spelling or syntactic errors, a larger number of students expressed challenges in obtaining adequate responses requiring well-engineered prompts.

In terms of ChatGPT's utilization by programming novices, students voiced caution against over-reliance on these tools, which is similar to the findings in prior work [44], [45].

Conversely, other students recognize increasing expectations by programming educators and see ChatGPT as a means to counteract this. Additionally, there is acknowledgment of other models, like ChatGPT-4 or Copilot, which may offer enhanced features, e.g., improved effectiveness or support for image/file input. However, some students are concerned about the associated costs, which potentially increases the disparities in the equal access and affordability of education. So far, GenAI does not seem to have democratized education.

VI. LIMITATIONS

The students' responses proved sufficient to reveal their use patterns and perspectives on GenAI tools, such as ChatGPT. However, the limits of self-reporting (e.g., social desirability, exaggerations, omissions) apply to the present work. Moreover, as students knew they were surveyed, the observer's paradox [52] needs to be taken into account when interpreting the data. Another limitation refers to the course setting, the selected tasks, and the institution where the data was gathered (i.e., in only one country). Expanding or replicating this study in other areas is considered future work.

VII. CONCLUSIONS AND FUTURE WORK

The present research pursued the goal of gaining insights into the student perspective of using GenAI tools, such as ChatGPT, in the context of a curricular introductory programming course. To identify students' use patterns along with their perceptions of the tool, we designed an exercise sheet with programming tasks for novice learners at a German higher education institution. Students were asked to use ChatGPT-3.5 (freely available at the time, in Dec. 2023) without further instruction. After submitting their solutions to the programming exercises, students (n=298) filled out an online survey to reflect upon their use and impressions.

The results revealed that roughly half of the students use ChatGPT as often as every week, 18% even use it daily. We further identified a great diversity of students' use patterns and application scenarios, which is due to the sheer capacity of GenAI. For example, students use it to get started with their assignments, to generate text and code, for debugging, or as a search tool when having questions. At the same time, students' perceptions showed their ambivalence towards GenAI, as students seem to be well aware of its limitations and the potential (negative) impact on student learning, and equal opportunities. It is therefore crucial for educators to at least acknowledge GenAI tools in computing and programming education, and to provide guardrails and instructions for students on how to navigate through the GenAI revolution.

There are several pathways for future work. Among them is the triangulation of the present work with chat protocols from novice learners of programming with a GenAI tool to address the limitations of self-reporting [53]. Another avenue is the development of specialized GPT models incorporated into well-known educational environments, to offer students free, straightforward access. Such models could contain built-in guardrails to facilitate a guided use. Integrating prompt-

recommender-systems could also assist students in optimizing their queries. In the long run, it is essential to not only offer tools as resource but to integrate them into educational setting, evaluate them, and reflect upon their use.

APPENDIX

A. Survey Questions

- Q1 Programming Experience before Course.
- Q2 Did you use ChatGPT prior to this exercise for assignments?
- Q3 How often do you use ChatGPT?
- Q4 On average, how long do you engage with ChatGPT in a single session?
- Q5 Which platform do you primarily use for accessing ChatGPT?
- Q6 Please select all the tasks for which you used ChatGPT. (Check all that apply)
- Q7 How would you rate the ease of using ChatGPT? (Likert scale, 1 Very difficult - 5 Very easy)
- Q8 To what extent has ChatGPT helped in improving your programming skills or solving coding problems? (Likert scale, 1 Not at all - 5 Greatly improved)
- Q9 Rate the accuracy and relevance of the responses provided by ChatGPT. (Likert scale, 1 Very inaccurate - 5 Very accurate)
- Q10 How satisfied are you with your overall experience using ChatGPT for programming assistance? (Likert scale, 1 Not at all - 5 Very satisfied)
- Q11 How likely are you to recommend ChatGPT as a support tool to a programming novice? (Likert scale, 1 Not at all - 5 Highly likely)
- Q12 Please share three positive aspects or examples of your experience using ChatGPT. What did you find most valuable or beneficial? (open)
- Q13 Please share three negative aspects or examples of your experience using ChatGPT. What did you find challenging or difficult? (open)
- Q14 Is there anything else you would like to share about your experience using ChatGPT? (open)

REFERENCES

- [1] S. Amani, L. White, T. Balart, L. Arora, D. K. J. Shryock, D. K. Brumbelow, and D. K. L. Watson, "Generative AI Perceptions: A Survey to Measure the Perceptions of Faculty, Staff, and Students on Generative AI Tools in Academia," 2023.
- [2] R. Raman, S. Mandal, P. Das, T. Kaur, J. Sanjanasri, and P. Nedungadi, "University students as early adopters of ChatGPT: Innovation Diffusion Study," 2023.
- [3] J. Prather, P. Denny, J. Leinonen, B. A. Becker, I. Albluwi, M. Craig, H. Keuning, N. Kiesler, T. Kohn, A. Luxton-Reilly, S. MacNeil, A. Petersen, R. Pettit, B. N. Reeves, and J. Savelka, "The Robots Are Here: Navigating the Generative AI Revolution in Computing Education," in *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*, ser. ITICSE-WGR '23. New York: ACM, 2023, p. 108–159.
- [4] N. Kiesler, *Modeling Programming Competency: A Qualitative Analysis*. Cham: Springer International Publishing, 2024. [Online]. Available: <https://doi.org/10.1007/978-3-031-47148-3>
- [5] B. A. Becker, M. Craig, P. Denny, H. Keuning, N. Kiesler, J. Leinonen, A. Luxton-Reilly, J. Prather, and K. Quille, "Generative AI in Introductory Programming," 2023. [Online]. Available: <https://csed.acm.org/wp-content/uploads/2023/12/Generative-AI-Nov-2023-Version.pdf>
- [6] N. Kiesler, "Beyond the Textbook: Rethinking Students' Competencies in the LLM Era," *Generative AI: Implications for Teaching and Learning*, Uppsala, Sweden, 2023.
- [7] C. Geng, Y. Zhang, B. Pientka, and X. Si, "Can ChatGPT Pass An Introductory Level Functional Language Programming Course?" 2023.
- [8] N. Kiesler and D. Schiffner, "Large Language Models in Introductory Programming Education: ChatGPT's Performance and Implications for Assessments," 2023, 10.48550/arXiv.2308.08572.
- [9] J. Savelka, A. Agarwal, C. Bogart, and M. Sakr, "Large Language Models (GPT) Struggle to Answer Multiple-Choice Questions about Code," 2023.
- [10] J. Leinonen, A. Hellas, S. Sarsa, B. Reeves, P. Denny, J. Prather, and B. A. Becker, "Using Large Language Models to Enhance Programming Error Messages," in *Proc. SIGSE*. ACM, Mar. 2023.
- [11] S. Sarsa, P. Denny, A. Hellas, and J. Leinonen, "Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models," in *Proc. ICER*. ACM, Aug. 2022.

- [12] S. MacNeil, A. Tran, A. Hellas, J. Kim, S. Sarsa, P. Denny, S. Bernstein, and J. Leinonen, "Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book," in *Proc. SIGCSE TS*, 2023, p. 931–937.
- [13] J. Leinonen, P. Denny, S. MacNeil, S. Sarsa, S. Bernstein, J. Kim, A. Tran, and A. Hellas, "Comparing Code Explanations Created by Students and Large Language Models," in *Proc. ITiCSE*, 2023.
- [14] D. Bengtsson and A. Kaliff, "Assessment Accuracy of a Large Language Model on Programming Assignments," 2023. [Online]. Available: <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-331000>
- [15] N. Kiesler, D. Lohr, and H. Keuning, "Exploring the Potential of Large Language Models to Generate Formative Programming Feedback," in *2023 IEEE Frontiers in Education Conference (FIE)*, 2024, pp. 1–5.
- [16] I. Azaiz, O. Deckarm, and S. Strickroth, "AI-enhanced Auto-Correction of Programming Exercises: How Effective is GPT-3.5?" *International Journal of Engineering Pedagogy*, vol. 13, no. 8, pp. 67–83, 2023.
- [17] I. Azaiz, N. Kiesler, and S. Strickroth, "Feedback-Generation for Programming Exercises With GPT-4," in *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. New York: ACM, 2024, p. 31–37. [Online]. Available: <https://doi.org/10.1145/3649217.3653594>
- [18] L. Roest, H. Keuning, and J. Jeuring, "Next-Step Hint Generation for Introductory Programming Using Large Language Models," 2023.
- [19] J. Jeuring, H. Keuning, S. Marwan, D. Bouvier, C. Izu, N. Kiesler, T. Lehtinen, D. Lohr, A. Peterson, and S. Sarsa, "Towards Giving Timely Formative Feedback and Hints to Novice Programmers," in *Proceedings of the 2022 Working Group Reports on Innovation and Technology in Computer Science Education*. New York: ACM, 2022, p. 95–115.
- [20] N. Kiesler, "Investigating the Use and Effects of Feedback in CodingBat Exercises: An Exploratory Thinking Aloud Study," in *2023 Future of Educational Innovation-Workshop Series Data in Action*, 2023.
- [21] E. Al-Hossami, R. Bunescu, J. Smith, and R. Teehan, "Can Language Models Employ the Socratic Method? Experiments with Code Debugging," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. New York: ACM, 2024, p. 53–59.
- [22] I. Joshi, R. Budhiraja, H. Dev, J. Kadia, M. O. Ataulah, S. Mitra, H. D. Akolekar, and D. Kumar, "ChatGPT in the Classroom: An Analysis of Its Strengths and Weaknesses for Solving Undergraduate Computer Science Questions," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, ser. SIGCSE 2024. New York: ACM, 2024, p. 625–631.
- [23] J. Prather, J. Leinonen, N. Kiesler, J. G. Benario, S. Lau, S. MacNeil, N. Norouzi, S. Opel, V. Pettit, L. Porter, B. N. Reeves, J. Savelka, D. H. Smith, S. Strickroth, and D. Zingaro, "How Instructors Incorporate Generative AI into Teaching Computing," in *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 2*, ser. ITiCSE 2024. New York: ACM, 2024, p. 771–772.
- [24] M. Amoozadeh, D. Daniels, D. Nam, A. Kumar, S. Chen, M. Hilton, S. Srinivasa Ragavan, and M. A. Alipour, "Trust in Generative AI among Students: An exploratory study," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, ser. SIGCSE 2024. New York: ACM, 2024, p. 67–73.
- [25] M. P. Rogers, H. M. Hillberg, and C. L. Groves, "Attitudes Towards the Use (and Misuse) of ChatGPT: A Preliminary Study," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, ser. SIGCSE 2024. New York: ACM, 2024, p. 1147–1153.
- [26] R. Liu, C. Zenke, C. Liu, A. Holmes, P. Thornton, and D. J. Malan, "Teaching CS50 with AI: Leveraging Generative Artificial Intelligence in Computer Science Education," in *Proc. SIGCSE*. New York: ACM, 2024, p. 750–756.
- [27] V. Grande, N. Kiesler, and M. A. F. Rodriguez, "Student Perspectives on Using a Large Language Model (LLM) for an Assignment on Professional Ethics," in *Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education V. 2*, ser. ITiCSE 2024. New York: ACM, 2024.
- [28] S. MacNeil, J. Leinonen, P. Denny, N. Kiesler, A. Hellas, J. Prather, B. A. Becker, M. Wermelinger, and K. Reid, "Discussing the Changing Landscape of Generative AI in Computing Education," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*, ser. SIGCSE 2024. New York: ACM, 2024, p. 1916.
- [29] J. Finnie-Ansley, P. Denny, B. A. Becker, A. Luxton-Reilly, and J. Prather, "The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming," in *Proc. ACE*. New York: ACM, 2022, p. 10–19.
- [30] J. Finnie-Ansley, P. Denny, A. Luxton-Reilly, E. A. Santos, J. Prather, and B. A. Becker, "My AI Wants to Know if This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises," in *Proc. ACE*. New York: ACM, 2023, p. 97–104.
- [31] N. Parlante, "CodingBat," Online Publication, 2024. [Online]. Available: <https://codingbat.com/about.html>
- [32] S. S. Gill, M. Xu, P. Patros, H. Wu, R. Kaur, K. Kaur, S. Fuller, M. Singh, P. Arora, A. K. Parlikad, V. Stankovski, A. Abraham, S. K. Ghosh, H. Lutfiyya, S. S. Kanhere, R. Bahsoon, O. Rana, S. Dustdar, R. Sakellariou, S. Uhlig, and R. Buyya, "Transformative effects of ChatGPT on modern education: Emerging Era of AI Chatbots," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 19–23, 2024.
- [33] X. Zhai, "ChatGPT User Experience: Implications for Education," 2022.
- [34] C. Zastudil, M. Rogalska, C. Kapp, J. Vaughn, and S. MacNeil, "Generative AI in Computing Education: Perspectives of Students and Instructors," in *2023 IEEE Frontiers in Education Conference*, 2023, pp. 1–9.
- [35] T. Phung, J. Cambronero, S. Gulwani, T. Kohn, R. Majumdar, A. Singla, and G. Soares, "Generating High-Precision Feedback for Programming Syntax Errors using Large Language Models," 2023.
- [36] J. Zhang, J. Cambronero, S. Gulwani, V. Le, R. Piskac, G. Soares, and G. Verbruggen, "Repairing Bugs in Python Assignments Using Large Language Models," *arXiv preprint arXiv:2209.14876*, 2022.
- [37] S. Narciss, *Informatives Tutorielles Feedback: Entwicklungs- und Evaluationsprinzipien auf der Basis instruktionspsychologischer Erkenntnisse*. Münster: Waxmann Verlag, 2006.
- [38] A. Taylor, A. Vassar, J. Renzella, and H. Pearce, "dcc -help: Transforming the Role of the Compiler by Generating Context-Aware Error Explanations with Large Language Models," in *Proc. SIGCSE*. New York: ACM, 2024, p. 1314–1320.
- [39] M. Kazemitabaar, R. Ye, X. Wang, A. Z. Henley, P. Denny, M. Craig, and T. Grossman, "CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and Educator Needs," *arXiv preprint arXiv:2401.11314*, 2024.
- [40] J. Prather, B. N. Reeves, P. Denny, B. A. Becker, J. Leinonen, A. Luxton-Reilly, G. Powell, J. Finnie-Ansley, and E. A. Santos, "'It's Weird That It Knows What I Want': Usability and Interactions with Copilot for Novice Programmers," *ACM Trans. Comput.-Hum. Interact.*, 2023.
- [41] P. Vaithilingam, T. Zhang, and E. L. Glassman, "Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models." New York: ACM, 2022, pp. 1–7.
- [42] D. Jayagopal, J. Lubin, and S. E. Chasins, "Exploring the learnability of program synthesizers by novice programmers," in *Proc. ACM Symposium on User Interface Software and Technology*, 2022, pp. 1–15.
- [43] P. Denny, J. Leinonen, J. Prather, A. Luxton-Reilly, T. Amarouche, B. A. Becker, and B. N. Reeves, "Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize AI Code Generators," 2023.
- [44] S. Barke, M. B. James, and N. Polikarpova, "Grounded Copilot: How Programmers Interact with Code-Generating Models," *Proc. ACM Program. Lang.*, vol. 7, no. OOPSLA1, 4 2023.
- [45] Y. Xue, H. Chen, G. R. Bai, R. Tairas, and Y. Huang, "Does ChatGPT Help With Introductory Programming? An Experiment of Students Using ChatGPT in CS1," 2024. [Online]. Available: <https://yuhuang-lab.github.io/paper/ICSE-SEET2024.pdf>
- [46] N. Forman, J. Udvaros, and M. S. Avornicului, "ChatGPT: A new study tool shaping the future for high school students," *Intern. Journ. of Advanced Natural Sciences and Engin. Researches*, vol. 7, no. 4, 2023.
- [47] OpenAI, "Prompt Engineering," <https://platform.openai.com/docs/guides/prompt-engineering>, 2023, accessed: 2024-04-08.
- [48] P. Mayring, "Qualitative content analysis," *Forum: qualitative social research*, vol. 1, no. 2, 2000.
- [49] S.-P. Ballstaedt, H. Mandl, W. Schnotz, and S.-O. Tergan, *Texte verstehen, Texte gestalten*. München: Urban u. Schwarzenberg, 1981.
- [50] H. Mandl, *Zur Psychologie der Textverarbeitung: Ansätze, Befunde, Probleme*. München: Urban & Schwarzenberg, 1981.
- [51] A. Scholl and N. Kiesler, "Data: Analyzing Chat Protocols of Novice Programmers Solving Introductory Programming Tasks with ChatGPT," Jul 2024. [Online]. Available: osf.io/wbkqv
- [52] F. J. Roethlisberger and W. J. Dickson, *Management and the Worker*. Cambridge: Harvard University Press, 1939.
- [53] A. Scholl, D. Schiffner, and N. Kiesler, "Analyzing Chat Protocols of Novice Programmers Solving Introductory Programming Tasks with ChatGPT," 2024, 10.48550/arXiv.2405.19132.