eleed
e-learning
& education

Issue 16
2025

Students' Use of ChatGPT in an Introductory Programming Course: A Deep Dive into Chat Protocols and the Student Perspective

Andreas Scholl
andreas.scholl@th-nuernberg.de

Nuremberg Tech
Computer Science
Keßlerplatz 12
90489 Nuremberg, Germany

Daniel Schiffner
d.schiffner@dipf.de

DIPF | Leibniz Institute for Research and Information in Education
Information Centre Education
Rostocker Straße 6
60323 Frankfurt am Main, Germany

Natalie Kiesler
natalie.kiesler@th-nuernberg.de

Nuremberg Tech
Computer Science
Keßlerplatz 12
90489 Nuremberg, Germany

Abstract

Within just a few years, Generative AI (GenAI) and related tools proved their impact on higher education, including computing disciplines. Their performance and feedback capabilities are convincing, especially for introductory classes, e.g., CS1 or Introduction to Programming. It is, therefore, no surprise that students are using GenAI tools at a great scale. In this invited research paper, we investigate students' use and perception of ChatGPT in an Introductory Programming class at Goethe University, Frankfurt, during the winter term 2023/24. To address this goal, we asked students to solve programming exercises with the assistance of ChatGPT as part of their weekly introductory course exercises. 213 students submitted their chat protocols (with 2335 prompts in sum) as a data basis for this analysis. The data was analyzed w.r.t. the prompts, frequencies, the chats' progress, contents, and other use patterns. In addition, students were asked to provide information regarding their use of ChatGPT and their evaluation of the tool via an online survey (n=298). The chat protocols revealed a great variety of interactions and (follow-up) prompts, both potentially supportive and concerning. Students' responses to the survey added a diverse range of perceptions, indicating a wide adoption but also critical engagement. Learning about students' interactions with ChatGPT will help inform and align teaching practices and instructions for future introductory programming courses in higher education. Therefore, this work has implications for tool creators and educators who want to design pedagogical instruction or guardrails for students and help inform their reflected use of GenAI tools.

Keywords: e-learning; ChatGPT-3.5; large language models; generative AI; GenAI; students; interaction pattern; application; chat protocols; log data; student survey; higher education

1 Introduction

Computing education and related study programs have the goal of producing competent [Ra21] graduates who are proficient in programming, software development, and several other technologies [Cl20]. In addition, they are expected to work collaboratively and be responsible, proactive, inventive, and meticulous [Kl23]. A high-quality education with successful graduates, however, requires extensive resources, mentoring, and, for example, formative feedback for learners, especially in introductory programming classes [Je22; Lo24].

Novice learners of programming are known to experience several challenges in the process, which have been subject to extensive research for decades [Du86; Lu18; SS86]. Among them are cognitively demanding competencies [Ki20; Ki24a], such as problem understanding, designing and writing algorithms, debugging, and understanding error messages [Du86; ER16; Ki20; Lu18; SS86]). Educators' expectations of novice learners and what they can achieve in their first semester(s) seem to be too high and unrealistic [Ki22; Lu16; Lu18; WCL07]. Moreover, the student-educator ratio in introductory programming classes keeps increasing in German higher education institutions, thereby limiting resources to provide feedback and hints, and adequately address heterogeneous prior knowledge and diverse educational biographies [Ki24b; Pe16; SB22].

Fortunately, Generative AI (GenAI) tools based on Large Language Models (LLMs) have the potential to support novice learners of programming seeking help, if carefully used. GenAI tools can pass introductory programming tasks and courses [Ge23; KS23a; Sa23], enhance programming error messages [Le23b; Ma23; Sa22], generate exercises [Ja25], and formative feedback [ADS23; AKS24; Az25; BK23; JJ24; KLK24; LKK25; RKJ23] for learners. It is therefore not surprising that the emergence of GenAI triggered an extensive discussion within the computing education community regarding learning objectives, curricula, assessments, ethical questions [Be23; Pr23b], changing competencies and integration practices [JGK24; Ke24; Ki23; Ki25; Pr24; Pr25].

When GenAI tools initially emerged, most studies investigated them from the educator's perspective, hypothesizing about (students') application scenarios (e.g., [Al24; Jo24]). The student perspective (e.g., their trust or attitude towards GenAI [Am24; RHG24]), let alone their actual use in a classroom setting, has only been addressed in a few very recent studies (e.g., [Bi24b; GKR24; Li24b; Xu24]). Understanding students' use of GenAI tools and respective interactions, however, is crucial to support them and adequately instruct them regarding the critical and reflective application of GenAI in their studies.

To address this gap, the present work investigates how students in an introductory programming class interact with GenAI tools, such as ChatGPT, when doing their coursework. Specifically, it is guided by the **research questions** (RQs): *1) How do students chat with ChatGPT in the context of introductory programming course assignments? 2) How do students perceive the use of ChatGPT in an introductory programming course?* To answer these questions, an exercise sheet was developed as part of an introductory programming course at Goethe University in the winter term of 2023/24. Students were instructed to solve programming tasks with the assistance of ChatGPT-3.5 (i.e., the freely available version at the time, in December 2023), and to submit their chat protocols, which is the first data basis of this work. In addition, we conducted a survey to ask for the student perspective regarding their interaction, perceived usefulness, benefits, concerns, etc. The **contribution** of this paper is thus a collection and analysis of authentic student interactions with a GenAI tool (i.e., ChatGPT) in a curricular course setting, which is accompanied by students’ self-reported evaluation of the tool. These insights have implications for educators considering the use of GenAI in their introductory programming courses, as they can help inform the development of new pedagogical instruction to foster a conscious and critical use of GenAI tools.

2 Related Work

In 2019, a literature review of 146 articles on artificial intelligence applications in higher education [Za19] pointed out the rapidly increasing relevance of Artificial Intelligence (AI) in computing education and related fields. The review identified four main areas of interest: “adaptive systems and personalization, assessment and evaluation, profiling and prediction and intelligent tutoring systems” [Za19, p. 11]. Notably, Zawacki-Richter et al. recognized “the dramatic lack of critical reflection” [Za19, p. 21] of the challenges and risks of AI tools, including pedagogical and ethical considerations.

Five years later, in 2024, Generative AI (GenAI) and related tools have already taken the world by storm [Pr23b; Pr24], and implications for higher education have become the focus of the discourse in many disciplines, including computing education research [Ma24]. This development is due to the broad availability of GenAI tools, which gained popularity with the launch of OpenAI’s ChatGPT in late November 2022. Its impressive performance in, for example, introductory programming tasks and exams [Ge23; KS23a; Sa23] initiated discussions on LLMs’ potential and how it would affect computing curricula, learning objectives, but also teaching, learning, and assessing in general [Be23; Ki23; Pr23a; Pr23b]. Prather et al. further emphasize the fast pace of GenAI tools and their developments, indicating the need for ongoing research to keep up with recent technological advances. This is especially true if research data (e.g., for benchmarking these tools) are not available [KS23b; Pr23b].

General concerns and limitations of LLMs and GenAI tools comprise the accuracy and reliability in educational settings, as ChatGPT is susceptible to (re-)producing biased or inaccurate information [Gi24]. The tool’s knowledge base, which may lack recent information, can further lead to inaccuracies. Hallucinations add to this problem. Finally, it can bypass plagiarism detection tools. All of these aspects challenge the integrity of academic work and the reliability of AI as an educational tool, indicating the need for action by educational institutions [Gi24; Pr23b; Zh22a].

2.1 Related Work Revealing the Potential of GenAI Tools

In the context of introductory programming education, studies revealed the potential of GenAI tools for several application scenarios. Among them are the effective generation of code explanations [Ma23; Sa22], enhanced programming error messages [Le23b], but also the analysis of student code with the goal of fixing students’ errors [Ph23; Zh22b], or providing (elaborate) feedback [AKS24; KLK24]. All of these studies focus on students’ (potential) use of LLMs.

For example, MacNeil et al. investigated students’ perceptions towards automatically generated line-by-line code explanations by OpenAI’s Codex and GPT-3 as part of an e-book. A majority of students evaluated the generated explanations as helpful [Ma23]. Similarly, Leinonen et al. found that students rate code explanations generated by GPT-3 better on average than explanations from their peers. They also note that students showed no aversion towards LLM-generated feedback, and that students prefer line-by-line code explanations [Le23a]. It is therefore not surprising that LLMs are being integrated into educational tools and environments to generate novice-friendly explanations tailored to each error [Ta24], without revealing code solutions [Ka24], or providing guardrails [Li24a].

Related to that is the use of LLMs to analyze students’ solutions and fix errors, or provide various types of elaborated feedback [Na06; Sh08] for novice programmers [KJH18]. Phung et al. investigated the use of LLMs to fix syntax errors in Python programs and developed a technique to receive high-precision feedback. Other studies qualitatively explored the feedback generated by ChatGPT [ADS23; AKS24; KLK24]. Kiesler et al. characterized the feedback generated by ChatGPT-3.5 in response to authentic student solutions for introductory programming tasks. They found stylistic elements, textual explanations of the cause of errors and their fix, illustrating examples, meta-cognitive and motivational elements, but also misleading information, uncertainty in the model’s response, and requests for more information by the LLM. Azaiz et al. [ADS23] noted difficulties of GPT-3.5 with the formatting of its output, recognizing correct solutions, and hallucinated errors. Roest et al. [RKJ23] conclude that the feedback generated by the GPT-3.5-turbo model seems to lack sufficient detail toward the end of an assignment. Nonetheless, a recent qualitative evaluation of GPT4 Turbo’s feedback shows notable improvements, as the outputs are more structured, consistent, and always personalized [AKS24]. Moreover, by using certain prompts, it is possible to elicit certain types of feedback from ChatGPT [LKK25].

2.2 Student-centered Studies

While some of the aforementioned studies utilize authentic student data (e.g., students’ solutions as input to LLMs), others directly involve students to survey their perspective or usability of LLMs. One example is the study by Prather et al. [Pr23c] who investigated students’ use of GitHub Copilot in an introductory programming assignment. Their observations and interviews explore students' perceptions regarding the (usability) challenges and potential benefits of this technology, resulting in design implications. Similarly, Vaithilingam et al. explored the usability of LLMs and their code generation abilities from the student perspective, concluding that Copilot's design should be improved [VZG22]. Jayagopal et al. explored the learnability of program synthesizers (including Copilot) by observing and interviewing students, which resulted in a set of lessons learned regarding system design. In another usability study, Barke et al. presents a grounded theory analysis of programmers' interactions with Copilot, based on observations of 20 experienced programmers.

Even though many studies have investigated potential applications of LLMs by computing students, and specifically, novice programmers [AKS24; KLK24; Le23a; Ma23], they are conducted from an educator's perspective, meaning all use cases have been predefined. So, they are not necessarily authentic. The presented usability studies [BJP23; JLC22; Pr23c; VZG22] were based on observations and interviews and did not focus on novice programmers and their application of GenAI and related tools. Another related study on students' interactions [De23] emphasizes quantitative aspects and does not analyze follow-up prompts and interaction patterns.

A handful of survey-based studies investigated students' perceptions and experiences with ChatGPT in programming education. They highlight that students see the tool as beneficial for understanding programming concepts, but are concerned about the reliability of generated information and ethical implications, such as plagiarism, persist [AL24; Bi24a; HW24]. It was found that ChatGPT is particularly useful for grasping abstract concepts and adapting code but less so for detailed implementations. Mixed reactions were noted in areas such as exam preparation or practical exercises, indicating that its effectiveness depends on the context and individual student expectations [AL24; HW24].

Other studies have combined surveys with chat protocol analyses to explore detailed behavioral patterns and use cases in programming contexts. For example, ChatGPT has been found to reduce reliance on other programming tools, such as forums or search engines, while being perceived as helpful for conceptual clarification and task-solving [MCK24; Su24; Xu24]. Interaction data from chat protocols revealed both strengths, such as expansive programming knowledge, and challenges, including inaccurate code and limited reliability for diverse tasks [MCK24; Su24]. These studies also identified distinct interaction sequences and highlighted the potential of ChatGPT to support programming education, despite varied effectiveness depending on task type and student behavior.

Few studies looked at students' use of LLMs based on actual interaction data from a computing course. Liu et al. focused on students' use of ChatGPT-3.5 as virtual teaching assistants in a classroom setting, focusing on the tools' effectiveness [Li24b]. Grande et al. evaluated the student perspectives on using ChatGPT for an assignment on professional ethics, using only the students' submissions to the task as a data basis [GKR24].

To conclude, there is a lack of authentic interaction data of students using GenAI tools, i.e., complete chat protocols, and their (qualitative) evaluation in the context of an introductory programming class. Therefore, the goal was to gather and analyze such data and complement them with students' perceptions and self-reported evaluation of such a tool.

3 Methodology

To answer the RQs *1) How do students chat with ChatGPT in the context of introductory programming course assignments? and 2) How do students perceive the use of ChatGPT in an introductory programming course?*, the study leverages empirical data collected from students enrolled in an introductory programming course (see also [SK24c; SSK24]). Students were asked to complete a series of programming exercises with the assistance of ChatGPT-3.5 and to submit their chat protocols as part of the assignment. These protocols are the first data source. The second data source is student responses to an accompanying online survey. In this section, we introduce the course context, instructions students received, and the selected tasks. Then we present the methodology applied for data gathering and analysis.

3.1 Course Context

The context of this study was an introductory programming course for first-year computing students (n=790) at Goethe University Frankfurt (Germany) during the winter term 2023/24. The majority were computer science students, with some individuals from other majors who pursue computer science as a minor. The course is designed for novice learners or programmers, hence, there are no prerequisites to participate. The class was accompanied by a Moodle course with learning materials. It comprised a 2-hour lecture every week for all 790 students and 2-hour tutorial sessions in groups of 20-30 students. Important components of the tutorial are the weekly or bi-weekly exercise sheets with programming tasks. Students are awarded points for (mostly individually) submitting their solutions, which contribute to the exam at the end of the semester.

3.2 Exercise Sheet and Selected Tasks

The exercise sheet addressed the concept of *recursion, functions, lists, conditionals, string manipulation, and documentation*. The exercise comprised two main tasks, while each one included sub-tasks. The first task, consisting of four sub-tasks, presented 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. Students were asked to read and interpret the given code snippets to determine the output of the code, the number of function calls, and the implemented type of recursion.

The second task required students to implement a function that determines the number of “happy strings” within all sub-strings of a given string (i.e., a string that can either be rearranged into (or already is) a repetition of some string). To solve this task, students were required to (2a) test a given string for the “happy” property (e.g., 20230322 can be rearranged into 02320232, which is a repetition of 0232). Task (2b) required students to test all possible sub-strings of the given string (e.g., to find pairs of integers meeting some conditions). Students were encouraged to adopt a recursive approach by awarding additional points.

Before the integration into the exercise sheet, the authors evaluated ChatGPT-3.5's capability to solve the selected tasks. For task 1, ChatGPT demonstrated a high success rate in identifying the correct functions but frequently miscalculated the number of calls, often overlooking the base case. Task 2 proved to be more challenging for ChatGPT-3.5. Although the responses included the necessary steps for problem-solving, they never included a correct solution (we tried 10 regenerations). This may have been due to a degree of ambiguity, and, for example, the uncommon term “happy string” in the task description.

3.3 RQ1: Data Gathering and Analysis of Chat Protocols

For the present study, a specific exercise sheet was developed for the tutorials in the week of December 6, 2023. Students were expected to submit their solutions two weeks later, group work was not permitted. Students were further 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. However, no specific instructions were provided on how to interact with ChatGPT-3.5, except for a reference to OpenAI's guide on prompt engineering [Op23]. This approach was deliberately chosen to avoid influencing students' interactions with ChatGPT-3.5. The exercise sheet was available in both German and English to accommodate the diverse student body of the course. Using ChatGPT for this exercise sheet was voluntary, but two extra points were offered as an incentive. Moreover, students were introduced to the present study's objective and the process during the lecture preceding the tutorial.

To evaluate students' interactions with ChatGPT-3.5, and how they seek assistance from the GenAI tool, the collected chat protocols were quantitatively and qualitatively analyzed. For the quantification of students' use patterns, the following aspects were investigated:

- *Number of prompts per student:* The total count of prompts given to ChatGPT-3.5, offering insights into the extent of interaction from students with the GenAI tool for the given tasks.
- *Number of words per prompt:* Identifying the verbosity or conciseness of students' prompts, to help characterize students' descriptions of the problem or task (and its complexity).
- *Follow-up interactions per student:* This characteristic aims to capture the iterative nature of problem-solving interactions with ChatGPT-3.5. It is supposed to help identify students' use patterns.

In addition, the chat protocols were **qualitatively** analyzed [Ma00] by two of the authors to identify the type of follow-up interactions, the contents or issues addressed by the students' prompts, and the overall (interaction) strategy students seem to pursue or apply.

- *Solution requests (SR):* Determines whether the student's prompt explicitly seeks a direct solution, shedding light on the intention behind the query and the expected outcome of the interaction (e.g., if only the task description is prompted). Frequencies emphasize the extent to which students apply this kind of request.
- *Type of follow-up interaction:* Categorizes whether a prompt is a standalone query (STA), a follow-up to a previous prompt (PRE), a response to a ChatGPT answer (RES), or a correction of a previous response (COR). We further add the frequencies of these types. Each prompt was categorized into only one of these four types.
- *Issue or problem-solving step addressed by the prompt:* This classification identifies the issue addressed by the prompt. These categories include common problem-solving steps and well-known issues for novice learners based on the literature

[Du86; ER16; Ki20; Lu18; SS86]: problem understanding (PU), conceptual understanding (CU), code generation (CG), debugging (DE), runtime analysis (RA), syntax/style (SY), documentation (DO), test cases (TC), and other categories (OT). It should be noted that a single prompt can reflect multiple issues within the problem-solving process.

- *Categories describing students' interactions:* Describes how students navigate and modify their prompts based on the generated output. Categories reflecting this interaction pattern were inductively built based on the material.

3.4 RQ2: Survey Development and Analysis

Guided by RQ2 and prior work, we designed the student survey. The goal was to gather students' self-reported use of ChatGPT when working on the given exercise sheet. So, we asked students how often, how long, and for which purposes they used ChatGPT in the problem-solving process. The answer options were informed by related work and the outlined application scenarios [ADS23; Le23b; Ma23; Ph23; Pr23b; VZG22].

Moreover, we addressed students' perception of ChatGPT, for example, regarding its usability, following prior work: *ease of use, perceived skills gain, accuracy and relevance of the responses*, and *user satisfaction* [De23; JLC22; Pr23c; VZG22]. Via open-ended questions, we also gathered students' thoughts and reflections when using the tool in a curricular course setting (both positive and negative).

Regarding demographics, we only addressed students' prior programming experience. This decision was due to the increasing diversity and heterogeneity of students' educational backgrounds when entering university and introductory courses. The survey questions are available in the Appendix.

We received 298 valid student responses to the survey. Closed questions (Q1-Q11) were quantitatively analyzed. The open questions (Q12-Q14) were addressed in alignment with qualitative research methods. Students' self-reported use patterns were discussed in Q1-Q6. We asked for programming experience before the introductory course in years, and whether students had used ChatGPT before the given exercise. Moreover, we asked about their usage frequencies and duration, how ChatGPT was accessed, and for which tasks or problems it was consulted in the context of solving the exercise sheet. Students' perceptions and experiences were the focus of questions Q7-Q14, where questions 7 to 11 referred to the ease of use and adoption criteria. The evaluation of these questions is based on a 5-point Likert scale given as answer options. Responses to the three remaining open questions were qualitatively analyzed [Ma00]. It should be noted that the open responses contained multiple aspects. For example, Q12 and Q13 requested three positive or negative aspects of using the GenAI tool. Hence, each meaningful element was treated as a coding unit and coded once. The students' full response to an open-ended survey question was treated as a context unit.

For the qualitative analysis of open-ended responses, we used deductive categories rooted in the literature, e.g., regarding ease of use, code explanations, use as a study buddy, or for debugging. Yet, we also developed new, inductive categories based on the material by following the psychology of text processing steps [Ba81; Ma81]. Responses were summarized, paraphrased, and abstracted to construct new categories. The inductive category development was iterative, and the initial categories were applied to a small proportion of the material (approx. 10%), before revising and extending the category scheme. This procedure was repeated several times until all responses were coded, and categories were finalized. Two coders discussed the edge cases until reaching a consensus.

4 Results of the Chat Protocols - Students' Use (RQ1)

A total of 360 students completed the exercise sheet. Of these, 305 students engaged with ChatGPT-3.5 as part of the exercise. Due to the absence of the required template usage or submission in non-processable file formats (e.g., unstructured Word files or alike), the data from 92 students could not be included in the final analysis. Therefore, the final dataset comprises 213 students, providing a collection of 2335 prompts for detailed examination. In total, we received 1668 German and 426 English prompts. (The data is available online [SK24a].)

Students engaged in approximately 10.96 prompts on average. The median during all interactions is 7 prompts per student. Based on these numbers, we categorize students into three groups, as shown in Table 1: group A, comprising individuals who use 0-5 prompts; group B, including those who use 6-11 prompts; and group C, consisting of students who use 12 or more prompts. The distinction yields almost equal-sized groups.

Range of prompts per student	No. of Students	No. of Prompts	Avg word count	Solution Requests (SR)	Follow-Up Interactions			
					STA	PRE	RES	COR
A: 0 to 5	66	262	73.74	213	172	57	25	8
B: 6 to 11	79	612	51.49	416	240	205	111	56
C: 12 and more	68	1461	40.46	603	358	462	322	319

Tab. 1: Quantification of students' interactions with ChatGPT-3.5, including some of the qualitative aspects of their follow-up interactions.

Regarding the evaluation of word counts per prompt, it should be noted that the number of words for task 1 alone is 146, and 175 for task 2. This partially explains the decrease in the average word count from group A to group B by 30%, followed by a further decline from group B to group C by 21% to 40.46 words on average per prompt. Furthermore, the number of solution requests (SR) shows a declining trend across the groups. In group A, approximately 81% of prompts are SR, which decreases to 68% in group B and 41% in group C (see Table 1).

Range of prompts per student	No. of Prompts	Problem-solving category								
		PU	CU	CG	DE	RA	SY	DO	TC	OT
A: 0 to 5	262	198	25	34	16	40	3	24	26	3
B: 6 to 11	612	303	87	121	101	55	10	83	58	10
C: 12 and more	1461	497	185	273	354	115	36	186	143	29

Tab. 2: Distribution of students' prompts across the various problem-solving categories (multiple categories per student and prompt).

The nature of follow-up interactions (FIs) also varies among the groups (see Table 1). In group A, the majority of interactions had no follow-up (STA, in 172 cases). In many cases, the prompts resembled the task descriptions, which had been used as input. At the same time, 90 instances of FIs were observed, comprising 57 follow-ups to a previous prompt (PRE), 25 responses to a generated answer by ChatGPT (RES), and 8 corrections of responses (COR), indicating a correction rate of approximately 3%.

Group B reveals an increase of FIs, with 372 instances of FIs, and somewhat fewer instances (240) without an FI (STA). Notably, there is a greater number of students referring back to their own prompts (PRE, 205 cases), while 56 corrections of a previous response (COR) were identified (9%). In contrast, group C demonstrates a significant increase of FIs, with the vast majority of follow-ups (1103 is sum). In group C, the correction rate (22%) was particularly high, with 319 correction instances (COR).

The analysis of student prompts relating to programming problem-solving categories reveals several trends, summarized in Table 2. For example, the proportion of initial prompts aimed at understanding the problem (PU), including solution requests, is smaller in longer conversations. In group A, these prompts constitute 76% of the interactions, dropping to 50% in group B, and further declining to 34% in group C. Conversely, there is an increase of prompts requesting help in debugging (DE) in longer conversations. This category sees a rise from 6% in group A to 17% in group B, and a further increase to 24% in group C. The number of prompts referring to runtime analyses (RA) is higher for group A, accounting for 15% of interactions. This may be due to fewer prompts in group A and runtime analysis being an explicit part of Task 1. In terms of concept understanding (CU), code generation (CG), and documentation (DO), there is a slight increase from group A to group B and C, while the percentage difference between group B and C is minimal. Additionally, the analysis shows a relatively stable percentage of prompts related to syntax/style (SY) from group A to C, and a relatively consistent ratio for test case prompts (TC) to overall prompts observed across group A, B, and C (about 10%).

After qualitatively analyzing the prompts, we were able to identify two common patterns of interactions with the GenAI tool: (1) the *Task Description Prompts* pattern, and (2) the *Prompts in Own Words* pattern. For (1), students seem to use the given task description at least as part of their initial prompt. In the other pattern (2), students rephrase the task description or immediately focus on specific parts of the problem statement. We provide a summary of both pattern and sub-pattern in Table 3 and Table 4, whereas the example ID refers to the student ID in the chat protocols. The order of the prompting patterns in the tables is supposed to reflect on the increasing specificity. Table 3 summarizes respective use pattern for *Task Description Prompts*, showing different approaches to generate a solution using ChatGPT after having used the task description as an initial prompt.

Example ID	Follow-ups after the initial task description prompt:
1012	No or few additional instructions.
1044	Giving direct orders / requests including style, documentation or corrections.
1199	Short, but extensive prompting for explanations to gain understanding.
1135	Specific prompts requesting explanations and corrections.
1070	Testing of initially generated code by the LLM. Providing incorrect (console) output followed by "Correct the code" instructions. No additional input or adaptation.
1195	Testing of initially generated code by the LLM. Providing incorrect (console) output followed by "Correct the code" instructions, resulting in disappointment.
1006	Testing of initially generated code by the LLM. Providing incorrect (console) output followed by "Correct the code" instructions. Followed by instruction to restart, and adaptation of instruction.
1156	Requesting (conceptual) explanations. Follow-up to create text and not bullet points.

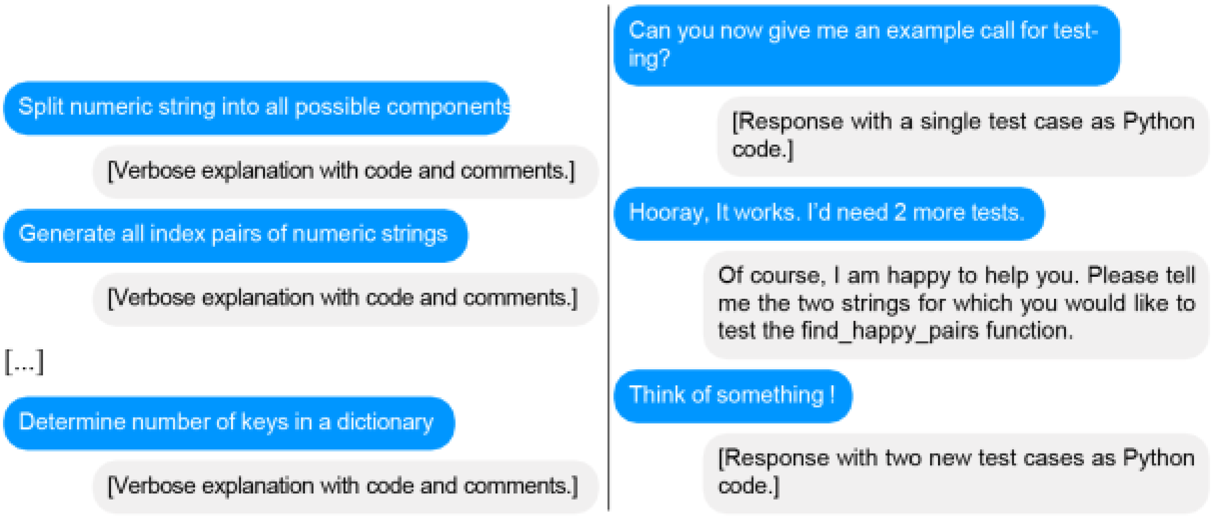
Tab. 3: Examples of the Task Description Prompts-patterns encountered in the dataset.

For the *Prompt in Own Words* pattern (summarized in Table 4), students exhibit different approaches to achieve the correct solution or responses to their questions. These patterns seem more directed towards a specific aspect of the problem or occurred only after the student had created a solution.

Example ID	Prompts and follow-ups in students' own words
1014	Demanding explanations for specific concepts, e.g. indices, dictionaries.
1058	Adding own solution and asking specific questions.
1106	Providing additional task constraints. Task description is provided as a follow-up. Requests specific aspects, e.g., test cases, and documentation, while providing examples.

Tab. 4: Examples of Prompts in Own Words-patterns found in the dataset.

In addition to these two patterns, we observed some other interesting aspects of students' interactions. This includes switching the language from German to English as soon as the GenAI tool produced English outputs. Some prompts started in English, but students asked to create documentation in German. Also, as part of the OT categorization, we found interjections, e.g., "Hooray", "Thank You!", which resemble human-to-human communication (ID 1037). In Figure 1, we present two exemplary excerpts from students and their ChatGPT interactions. Figure 1a illustrates that the student did not write complete sentences, but assigned explicit tasks to be executed. Figure 1b shows a human-to-human interaction with the tool.



Split numeric string into all possible components

[Verbose explanation with code and comments.]

Generate all index pairs of numeric strings

[Verbose explanation with code and comments.]

[...]

Determine number of keys in a dictionary

[Verbose explanation with code and comments.]

Can you now give me an example call for testing?

[Response with a single test case as Python code.]

Hooray, It works. I’d need 2 more tests.

Of course, I am happy to help you. Please tell me the two strings for which you would like to test the find_happy_pairs function.

Think of something !

[Response with two new test cases as Python code.]

(a) Excerpt from chat protocol 1014 with brief and direct commands.

(b) Excerpt from chat protocol 1037 resembling human-to-human interaction.

Fig. 1: Excerpts of two example chat protocols illustrating different interaction patterns (translated into English by the authors).

5 Results of the Student Survey - Students’ Perspectives (RQ2)

This section addresses the second RQ, namely, how students perceive the use of ChatGPT in an introductory programming course. In the following, we present the results of the 298 students who responded to the survey. We present both quantitative and qualitative results (data is available online [SK24b]).

5.1 Quantitative Results

Survey questions Q1 to Q11 were mandatory for the submission of the survey, so we had 298 responses each for these questions. With this high number of responses, students provided significant insights into their perspectives and their self-reported use of ChatGPT-3.5.

The surveyed students were mostly programming novices (Q1). 34% of them had no programming experience at all, 43% had limited experience of less than one year, 17% had one to two years of programming experience before enrolling in the course, and 6% had more than 3 years of programming experience. We also asked about students’ use of ChatGPT before solving the exercise sheet (Q2). 84% of the 298 students reported to have used ChatGPT for assignments, while 16% negated that question. This means that the majority of students had little to no programming experience, but most of them had previously used ChatGPT for coursework.

Students further indicated the frequency of using ChatGPT (Q3). About half of the students (52%) work with ChatGPT weekly, 18% even daily. 6% reported a monthly use, while 16% use it rarely, and only 8% of students have never used ChatGPT. In Q4, we asked for the average duration of the engagement with the GenAI tool. 43% of students use it for less than 15 minutes, 38% use it for 15-30 minutes. The remaining 16% use it for 30-60 minutes, or more than 60 minutes (3%). In addition (Q5), we found that students primarily use the ChatGPT interface (95%). Only 2%, respectively 3% use the messaging App integration or the API integration.

In response to Q6 (multiple-choice), students selected all tasks for which they used ChatGPT when working on the given programming problems from a predefined list of answer options. We received 223 selections for *problem understanding*, 178 for *conceptual understanding*, 176 for *code generation*, 134 for *debugging*, 102 for *producing documentation*, 102 for *test cases*, 90 for *correcting syntax*, and 35 for *runtime analysis*. An additional open input field was used by 52 students. They added the following tasks for which they used ChatGPT: 13 for *coding*, 9 to get *general explanations*, 8 as a *writing assistant*, 7 for *private use*, 4 as a *study buddy*, 4 used it as a *search tool*, and 3 as a *starting point* when programming.

Table 5 represents the distribution of student responses to the 5-point Likert-scale questions Q7 to Q11 (n=298 each). These questions focused on students' perspectives and impressions when 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).

Question		**1**	**2**	**3**	**4**	**5**	
Q7: How would you rate the ease of using ChatGPT?	Very Difficult	1%	3%	21%	38%	37%	Very Easy
Q8: To what extent has ChatGPT helped in improving your programming skills or solving coding problems?	Not helped at all	3%	12%	37%	38%	10%	Helped very well
Q9: Rate the accuracy and relevance of the responses provided by ChatGPT	Very Inaccurate	3%	28%	52%	16%	2%	Very Accurate
Q10: How satisfied are you with your overall experience using ChatGPT for programming assistance?	Very Dissatisfied	3%	12%	35%	40%	11%	Very Satisfied
Q11: How likely are you to recommend ChatGPT as a support tool to a programming novice?	Very Unlikely	5%	15%	21%	30%	29%	Very Likely

Tab. 5: Likert scale assessment of student impressions on using ChatGPT. Percentages represent the distribution of responses from negative (1) to positive (5). Colors vary from red (disagreement) to blue (agreement) to represent increasing agreement.

In Q7, we asked for ChatGPT's ease of ease from students' perspective. The median and mean of the responses were 4 and 4.06, respectively, on a 5-point-Likert scale ranging from "very difficult" to "very easy". Thus, the majority of students indicated that ChatGPT is easy or very easy to use (see Table 5). In Q8, students rated the extent to which ChatGPT has helped improve programming skills or solve coding problems (again on a 5-point-Likert scale). The median was 3 and the mean was 3.40. Hence, students tended to perceive ChatGPT as helpful or very helpful. It should be noted though that 37% remain undecided (see Table 5). Regarding the accuracy and relevance of ChatGPT's responses (Q9), opinions varied more, with a median of 3 and a mean of 2.87. These numbers indicate that many students recognize the deficits of the output concerning accuracy and relevance. (see Table 5). When asked about the overall satisfaction with ChatGPT for programming assistance

(Q10), the median of student responses was 4 and the mean was 3.44, which is a positive trend. Many students expressed high levels of satisfaction with the tool (see Table 5). In Q11, we asked students about the likelihood of recommending ChatGPT as a support tool. Responses were mostly positive, with a median of 4 and a mean of 3.63. 59% of students were likely or very likely to recommend ChatGPT for programming novices (see Table 5).

5.2 Qualitative Results

In the following, we present the responses to the open-ended questions (Q12-Q14). These questions requested students to express positive and negative aspects of their experiences using ChatGPT for their coursework in introductory programming (Q12 and Q13), as well as other comments students wanted to make (Q14).

In Table 6, we summarize the deductive-inductive categories of positive and negative aspects in response to Q12 (n=261) and Q13 (n=259). These are based on the 740 coding units (i.e., one coding unit represents one meaning) containing positive aspects (Q12) and 682 coding units with negative aspects (Q13). Hence, each response contained multiple aspects. The first column of Table 6 represents the category label, which is followed by the category definition. In the next column, we provide the number of coding units (#Pos.) positively representing these categories and provide a positive anchor example. Likewise, the column #Neg. represents the number of coding units pointing out the negative perspective, followed by a negative anchor example. It should be noted that for some categories students described both positive and negative perspectives.

The initial 7 categories displayed in Table 6 reflect students' use of ChatGPT. For example, they use it as a help when getting started with a problem, and for retrieving conceptual input, but also as a study buddy or for generating code and text, such as test cases and documentation. Debugging is another use case, and so is the generation of alternative perspectives or solutions. Some students used ChatGPT as a search tool, as they found it easier to use other search engines.

The subsequent 13 categories in Table 6 summarize students' perceptions of qualities or characteristics related to the use of ChatGPT. It should be noted that many of these characteristics are evaluated both positively and negatively. Among them are the response quality, the availability of the chat history, the general availability of the tool, how easy it is to use, the time-wise efficiency, its knowledge base, and social interaction. All of these qualities were critically received by students, and we provide positive and negative anchor examples for these two contrary perspectives in Table 6. Interestingly, positive and negative comments on the response quality, knowledge base, and social interaction are approximately even. Regarding the ease of use and the available chat history, students' responses were more negative citing, for example, the need to ask multiple times before receiving the desired answer.

Category	Definition	#Pos.	Positive Anchor Example	#Neg.	Negative Anchor Example
Starting Point	Denotes the initial direction and guidance offered by ChatGPT, such as code templates.	45	“Especially when you have no idea how to approach a task, ChatGPT is incredibly helpful.”	-	-
Conceptual Input	Describes ChatGPT’s capacity to offer fresh ideas, concepts, and inspiration, including recommendations for functions, libraries, and tips.	59	“You can have entire concepts explained, something that is very difficult to achieve through self-study.”	-	-
Study Buddy	Describes ChatGPT as a supportive companion tool offering personalized guidance, tutoring, and interactive help.	23	“Chatgpt can teach me as a patient teacher.”	-	-
Code and Text Generation	Describes ChatGPT’s capability to create, modify, or complete different types of content, including code, text, tests, and documentation.	89	“ChatGPT can write working code for simpler tasks. For more complex tasks, individual code fragments from the AI can serve as inspiration or building blocks for your own code.”	111	“The AI often cannot generate code based on a question, or the code does not work.”
Debugging	Describes ChatGPT’s role in troubleshooting and improving code by detecting and resolving compile, runtime, syntax, and logical errors.	85	“When I encounter an issue where I can’t resolve an error message, ChatGPT can be very helpful.”	5	“When reviewing, ChatGPT itself makes mistakes.”
Alternative Perspectives and Solutions	Highlights ChatGPT’s ability to present students with diverse opinions, viewpoints, and strategies for programming tasks; includes offering alternative solutions and rephrasing questions or answers in different ways.	18	“Very handy at getting another opinion on what an exercise expects me to do”	18	“Often, the proposed solution is the only option; ChatGPT cannot find alternative solutions.”
Search Tool	Describes ChatGPT as a tool for retrieving information and locating answers, much like search engines or online forums.	29	“I don’t have to spend a long time searching the web myself; instead, I can have ChatGPT search for assistance.”	1	“The outputs often do not go beyond a summary of Google search results.”
Response Quality	Refers to the accuracy, clarity, comprehensibility, and usefulness of ChatGPT’s answers, as well as their structure and format.	146	“Provides clear, concise explanations of complex programming concepts, aiding in quick learning and comprehension.”	142	“Sometimes the content is not well explained, or strange terms and/or explanations are used.”
Chat History	Describes ChatGPT’s ability to remember past interactions, allowing users to review and modify previous queries and responses.	27	“The program learns during the conversation and can respond increasingly faster. This means you don’t have to explain the context with every new question.”	79	“’Sorry, my mistake’ loop.”
Availability	Refers to the requirements for accessing ChatGPT (e.g., account/ subscription), the availability of servers, and ChatGPT’s response times.	84	“Always ready to answer questions.”	35	“Sometimes the servers are overloaded at night.”
Ease of Use	Refers to how accessible and user-friendly ChatGPT is; includes its ability to understand natural language inputs, as well as the structure, clarity, and precision of instructions.	67	“Simple and easy-to-use interface.”	119	“You usually have to ask multiple times before it really gives the answer you are looking for.”

Time Efficiency	Refers to ChatGPT’s ability to save time by delivering quick answers and reducing workload, even though responses may need verification.	29	“I think ChatGPT can really save you time if used correctly.”	10	“Sometimes it is even faster to just program it yourself.”
Knowledge Base	Refers to the breadth and depth of information available from the training data; includes the versatility, relevance, and currency of the information.	31	“ChatGPT can be used for a wide variety of things.”	29	“The AI often fails because it is not given enough data.”
Social Interaction	Refers to how ChatGPT interacts with students, emphasizing the conversational aspects and the overall communication experience.	8	“It can engage in conversations, answer questions, and even tell stories. This makes it an interesting and entertaining companion.”	9	“It expresses itself very technically/precisely. You can tell it is a machine.”
Privacy Concerns	Refers to students’ perceptions regarding the confidentiality and privacy of their interactions with ChatGPT.	-	-	5	“Posing some information security risks.”
Overconfidence	Refers to ChatGPT presenting information with a high degree of certainty, even when the data may be inaccurate or questionable.	-	-	29	“Tendency to Generate Plausible-sounding but Incorrect Answers”
Hallucinations	Refers to ChatGPT generating responses that contain inaccurate or fabricated information, such as referencing non-existent sources.	-	-	16	“ChatGPT always argues in a way that sounds convincing, even when the solution is wrong. It does not indicate how confident it is in its answer.”
Lack of Integrity	Refers to the clarity and reliability of the information provided, including the absence of credible sources that ensure academic integrity.	-	-	12	“You quickly get a plagiarism issue because ChatGPT crawls the web, just like I do when I search for a solution and check the first results.”
Criticizing GenAI	Refers to students needing to critically assess and verify ChatGPT’s responses.	-	-	36	“For the same questions, you occasionally get different answers, each with different results and approaches.”
GenAI dependency	Refers to the risk of students becoming overly reliant on ChatGPT for programming exercises.	-	-	26	“Relying too heavily on ChatGPT for problem-solving can potentially stifle one’s and problem-solving skills.”

Tab. 6: Categories, definitions, and anchor examples reflecting students’ positive and negative perceptions of ChatGPT in response to Q12 and Q13.

Among these 13 categories are 6 aspects that were perceived exclusively negative. They comprise privacy concerns, overconfident responses, hallucinations, lack of integrity, the need to criticize the tool, and the risk of depending on the GenAI tool.

Q14 asked for additional remarks; 136 students replied. Although they mostly repeated the positive and negative aspects mentioned in Q12 and Q13, students highlighted a few interesting experiences and insights, such as:

> Its ability to answer in multiple ways can simplify and enhance the learning experience, catering to different styles of understanding and preferences."
>
> "It is important to use ChatGPT as a support tool to help you learn even more, essentially as a learning partner. You must not be tempted to take the easy way out."
>
> "If you let ChatGPT do everything, it can quickly backfire. You don't learn anything, and at some point, you become completely dependent on ChatGPT.

6 Discussion

In terms of RQ1, the analysis of students' chat protocols indicates diverse interactions with GenAI tools, such as ChatGPT, when being applied as part of an introductory programming course in higher education. For example, students engaged differently with ChatGPT in terms of the number of prompts used, their length, and regarding (the content of) follow-up interactions.

One group of students (n=66) only used 0-5 prompts while working on the selected tasks. This group may not have been thoroughly engaged in the interaction with ChatGPT. There is also a large number of solution requests (SR) present in this group. This may be an indicator for students looking for a quick solution to the given problems.

The second group of students (n=79) used ChatGPT differently. For example, we see an increasing number of prompts in this group. Still, many students requested the correct solution (SR) immediately. What is interesting though is that there were many follow-up interactions in that group. This might indicate that either the initial response of ChatGPT was not considered good enough, or that the generated response fostered an interest in (an aspect of) the topic or problem, or the urge to generate a better or different solution. The increasing number of follow-ups on previous prompts (PRE) and responses to a ChatGPT answer (RES) disambiguate this group from students who only submitted 0-5 prompts.

The third group of students (n=68) used the GenAI tool most frequently but for different purposes. For example, a smaller percentage of immediate solution requests was identified. Especially due to the high number of corrections (COR), students seemed to be more reflective of the generated answers when trying to generate valid solutions. Especially interesting is the increase in responses to a ChatGPT answer (RES), which shows that students used the tool as a "learning partner" instead of a simple query or search tool. This is also supported by the high number of prompts requesting help for problem understanding (PU) and debugging (DE).

The qualitative data analysis further reveals a difference in the categorization of problem-solving steps, the identified follow-up interactions, and prompting patterns (*Task Description Prompts and Prompts in Own Words*). While the former are well known (e.g., as knowledge on how to proceed), especially in the context of feedback [KJH18; KLK24; Lo24], the latter represent entirely new categories aiming to describe students' interactions and prompting patterns with ChatGPT. Similar patterns have been found in a recent study [Su24]. Educators can use these patterns as a basis for designing instructions on how to (not) use LLMs and related tools, and to reflect on their limitations and potential benefits. For example, Lohr et al. have shown how to generate specific feedback types [LKK25].

To answer our first research question *How do students chat with ChatGPT in the context of introductory programming course assignments?*, we conclude that there is a certain variety and range of students' applications of the tool. There are at least three very different behaviors present. One seems to focus on problem-solving by generating a single prompt to generate the best answer possible. Other students revealed a more "peer-like" interaction with the GenAI tool. They used more follow-up prompts, focused on problem-solving step-by-step, tried to use the tool to identify and correct errors, and communicated respectively (we thus observed signs of anthropomorphism towards ChatGPT in this study). The third recognized use pattern is more straightforward and focuses on code generation, but utilizes the option of having an interactive tool. We assume that there are even more facets to these behaviors.

Building on the varying ways students interacted with ChatGPT during their programming tasks, the analysis highlights not only differences in engagement but also the broader perceptions of the tool's usefulness.

The survey analysis (n=298) in response to RQ2 (*How do students perceive the use of ChatGPT in an introductory programming course?*) indicates that many students integrate ChatGPT into their daily practices when tackling programming tasks, applying it to a wide variety of problems. Several of the mentioned applications were also identified in earlier studies, such as the role of GenAI tools in debugging [AL24; Pr23b]. While students generally view ChatGPT as a valuable tool offering significant benefits, they are also aware of several challenges, which include hallucinations, overconfidence, compromising academic integrity, and possibly depending on the tool. These observations align with findings from recent studies [Bi24a] and show that students do critically reflect on GenAI tools and using them.

Although some students expressed frustration with the occasional unavailability of ChatGPT, many appreciated the immediate responses it provides and the interactive nature of the tool. Furthermore, the multilingual capabilities of ChatGPT were seen as an advantage, with students particularly appreciating the ease of switching between German and English. This is especially useful in German CS classes, as technical terms are usually used in English, while classes are generally taught in German.

We also recognized that several students faced difficulties in understanding programming tasks and getting started with programming. In these cases, ChatGPT is considered a helpful starting point, which is a confirmation of earlier findings [MCK24; VZG22]. While some students were satisfied with the answers from ChatGPT despite minor errors, a larger group of students expressed frustration, as useful responses required well-crafted and precise prompts, especially for more complex tasks. This finding has also been observed in similar studies [HW24; Su24; Xu24] and confirms that students need support from educators on how to utilize GenAI tools more effectively to receive high-quality, and correct responses.

When considering the use of ChatGPT by programming novices, the students highlighted concerns about becoming too dependent on such tools, a sentiment echoed in related work [BJP23; Xu24]. At the same time, some students see ChatGPT as a way to meet educators' expanding expectations. This student remark is interesting, as it shows that some students perceive a need to use these tools to overcome challenges they face as part of their classes. There is also recognition of other advanced models, such as ChatGPT-4 or Copilot, which promise improved features. Yet, students expressed concern about the costs associated with these tools, as they may exacerbate inequalities in access to education. It is, however, unclear if the paid versions provide additional benefits and how to address these inequalities in formal educational settings in general.

Regarding RQ2, we further found that overall, students perceive ChatGPT as a useful and versatile tool with many positive aspects. In their perspective, it offers support across a wide range of tasks, from debugging to understanding complex concepts. Students appreciate its interactive nature, multilingual capabilities, and ability to provide quick responses 24/7. They also proved to look at GenAI tools from a critical lens and raised concerns about, e.g., hallucinations, overconfidence, or the potential for over-reliance. Educators (and tool creators) need to address these concerns accordingly.

The findings from both research questions reveal similarities between students' diverse behavioral patterns and their perceptions of ChatGPT's usefulness and risks. In our unsupervised scenario, students exhibited various approaches to interacting with ChatGPT. They also demonstrated a wide range of opinions about the tool's benefits and limitations. One takeaway is related to the next steps, and how to enhance teaching with the knowledge from the present study. This can be done in several ways, ranging from designing a “study buddy” interface for ChatGPT to using it as a support tool for debugging in programming classes. Students' self-guided use of ChatGPT showed that they have developed some strategies and recognize numerous benefits and limitations. Yet, leaving students alone with GenAI tools still seems like the worst choice. Initial guidance and providing knowledge about the tools are likely to add even more beneficial use while avoiding harm and limitations.

7 Limitations

A limitation of this study is the context of the data collection, which was a single class taking place at one university. Even though the high number of students participating in the study strengthens its representative character, the results may not be transferable to other educational contexts. Another limitation is that due to their knowledge of the study, students may have interacted differently with the tool than they would normally have in the context of an assignment, which is known as the observer's paradox [RD39]. Likewise, students may have submitted socially desirable responses, exaggerated or omitted various aspects as part of their survey responses. We also did not connect the chat protocols of individual students to their survey responses. Conducting a pseudonymized study in which chat protocols, survey results, and perhaps course outcomes are triangulated could be an interesting avenue for future work.

8 Conclusions and Future Work

This study aimed to explore how students engage with GenAI tools, like ChatGPT, in the context of an introductory programming course. To investigate both students' use patterns and self-reported perceptions of the tool, we designed a set of programming exercises for novice learners at a German higher education institution. Students were encouraged to use ChatGPT 3.5 (freely available at the time, in December 2023) without specific guidance. Students' chat protocols were quantitatively and qualitatively analyzed to characterize students' use and interaction patterns. The high participation rate (n=213 chat protocols) allows us to draw several conclusions. For example, the results reveal three groups of students interacting differently with the tool. Among these groups are students seeking an immediate solution, but also students seeking help with problem understanding, conceptual knowledge, debugging, syntax error or style, and documentation. Some students also used the chat more extensively than others, thereby resembling a "study buddy" or "virtual tutor" that is available around the clock. The chat protocols further revealed new patterns of interactions, e.g., students starting with *Task Description Prompts* as input, and specifying their requests later, and those using *Prompts in Own Words* with more specific requests. However, there is a huge diversity in students' use of ChatGPT, which may be due to the lack of previous instruction on how to use the tool.

Additionally, we conducted an online survey to gather the student perspective (n=298 responses). The findings indicate that around half of the students used ChatGPT at least once a week, with 18% reporting daily use, which aligns with similar recent studies [AL24; Bi24a]. Students confirmed a wide range of application scenarios, including using ChatGPT for the first steps of the problem-solving process, code and text generation, debugging, and as an information resource. At the same time, students expressed mixed perceptions of GenAI tools, recognizing both their benefits and their limitations. Concerns included the potential negative impact on learning, over-reliance, and the constant need to evaluate every output.

Implications for teaching and learning are manifold. For example, the diversity of use patterns reflects the heterogeneity of students' needs for support, and feedback. GenAI tools have the potential to offer that degree of personalization [AKS24]. Regardless, educators need to provide the necessary tools and advice on how to use them successfully, e.g., by providing prompt patterns [Wh23] or prompts resulting in specific feedback types [LKK25]. Moreover, it is important to communicate the tools' limitations and potential pitfalls to students so they do not end up frustrated or accused of cheating. The results underline the importance of educators addressing the role of GenAI tools in programming education, and guiding students toward informed decisions regarding the use of AI technologies (see also [Pr24]). The computing education community also needs to continue their discussion on the competencies required for successful graduates, and the role of GenAI tools in that process [JGK24; Ki23].

As our approach has been executed with a general LLM (ChatGPT-3.5), it would be interesting to see how a specialized AI tool for coding would increase its support for learners as part of future work. Such a tool could be integrated within common learning platforms to ensure access (but not necessarily accessibility). Specialized models could include built-in safeguards to promote responsible use and features like prompt recommendations (e.g., [LKK25]) to help students refine their queries. Ultimately, the effective incorporation of GenAI tools into educational contexts requires not only offering them as a resource but also rigorously evaluating their impact and fostering thoughtful reflection upon their use. Moreover, additional replication studies at other institutions or countries would further strengthen the results of this work and other recent studies [Bi24a; MCK24; Su24].

Appendix: Survey Questions

Q1 Programming Experience before Course.

- ○ None
- ○ Limited (<1 Year)
- ○ Moderate (1-2 Years)
- ○ Advanced (3+ Years)

Q2 Did you use ChatGPT prior to this exercise for assignments?

- ○ Yes
- ○ No

Q3 How often do you use ChatGPT?

- ○ Never
- ○ Daily
- ○ Weekly
- ○ Monthly
- ○ Rarely

Q4 On average, how long do you engage with ChatGPT in a single session?

- ○ Less than 15 minutes
- ○ 15-30 minutes
- ○ 30-60 minutes
- ○ More than 60 minutes

Q5 Which platform do you primarily use for accessing ChatGPT?

- ○ Web Interface (as required in exercise)
- ○ API Integration (e.g. IDE Plugins like CodeGPT)
- ○ Messaging App Integration (e.g. Chat Bots like MyAI)

Q6 Please select all the tasks for which you used ChatGPT. (Check all that apply)

- □ Problem Understanding
- □ Conceptual Understanding
- □ Code Generation
- □ Debugging
- □ Runtime Analysis
- □ Syntax
- □ Documentation
- □ Test Cases
- □ Here you can add tasks that were not listed as possible answers in the previous question. (open question)

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 - 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

[ADS23] Azaiz, I.; Deckarm, O.; Strickroth, S.: AI-enhanced Auto-Correction of Programming Exercises: How Effective is GPT-3.5? In: International Journal of Engineering Pedagogy (iJEP), 13 (8), pp. 67–83, 2023, DOI: 10.3991/ijep.v13i8.45621 (last check 2025-04-28).

[AKS24] Azaiz, I.; Kiesler, N.; Strickroth, S.: Feedback-Generation for Programming Exercises With GPT-4, 2024, DOI: 10.48550/arXiv.2403.04449, arXiv: 2403.04449 [cs.AI] (last check 2025-04-28).

[AL24] Andersen-Kiel, N.; Linos, P.: Using ChatGPT in Undergraduate Computer Science and Software Engineering Courses: A Students' Perspective. In: 2024 IEEE Frontiers in Education Conference (FIE). 2024, https://www.computer.org/csdl/proceedings-article/fie/2024/10892934/24Enxb94bsY (last check 2025-04-28).

[Al24] Al-Hossami, E.; Bunescu, R.; Smith, J.; Teehan, R.: 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. SIGCSE 2024, ACM, Portland, OR, USA, pp. 53–59, 2024, DOI: 10.1145/3626252.3630799 (last check 2025-04-28).

[Am24] Amoozadeh, M.; Daniels, D.; Nam, D.; Kumar, A.; Chen, S.; Hilton, M.; Srinivasa Ragavan, S.; Alipour, M. A.: Trust in Generative AI among Students: An exploratory study. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1. ACM, New York, pp. 67–73, 2024, DOI: 10.1145/3626252.3630842 (last check 2025-04-28).

[Az25] Azaiz, I.; Kiesler, N.; Strickroth, S.; Zhang, A.: Open, Small, Rigmarole – Evaluating Llama 3.2 3B's Feedback for Programming Exercises. In: International Journal of Engineering Pedagogy (iJEP; eISSN: 2192-4880). Preprint available, 2025, DOI: 10.48550/arXiv.2504.01054, arXiv: 2504.01054 [cs.CY] (last check 2025-04-28).

[Ba81] Ballstaedt, S.-P.; Mandl, H.; Schnotz, W.; Tergan, S.-.-O.: Texte verstehen, Texte gestalten. Urban u. Schwarzenberg, München, 1981.

[Be23] Becker, B. A.; Craig, M.; Denny, P.; Keuning, H.; Kiesler, N.; Leinonen, J.; Luxton- Reilly, A.; Prather, J.; Quille, K.: Generative AI in Introductory Programming. 2023, url: https://csed.acm.org/wp-content/uploads/2023/12/Generative-AI-Nov-2023-Version.pdf (last check 2025-04-28).

[Bi24a] Bikanga Ada, M.: It Helps with Crap Lecturers and Their Low Effort: Investigating Computer Science Students' Perceptions of Using ChatGPT for Learning. In: Education Sciences 14 (10), 2024, ISSN: 2227-7102, DOI: 10.3390/educsci14101106, url: https://www.mdpi.com/2227-7102/14/10/1106 (last check 2025-04-28).

[Bi24b] Bikanga Ada, M.: It Helps with Crap Lecturers and Their Low Effort: Investigating Computer Science Students' Perceptions of Using ChatGPT for Learning. In: Education Sciences 14 (10), 2024, ISSN: 2227-7102, DOI: 10.3390/educsci14101106, url: https://www.mdpi.com/2227-7102/14/10/1106 (last check 2025-04-28).

[BJP23] Barke, S.; James, M. B.; Polikarpova, N.: Grounded Copilot: How Programmers Interact with Code-Generating Models. Proc. ACM Program. Lang. 7 (OOPSLA1), 2023, DOI: 10.1145/3586030 (last check 2025-04-28).

[BK23] Bengtsson, D.; Kaliff, A.: Assessment Accuracy of a Large Language Model on Program- ming Assignments, 2023, url: https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-331000 (last check 2025-04-28).

[Cl20] Clear, A.; Parrish, A.; Impagliazzo, J.; Wang, P.; Ciancarini, P.; Cuadros-Vargas, E.; Frezza, S.; Gal-Ezer, J.; Pears, A.; Takada, S.; Topi, H.; van der Veer, G.; Vichare, A.; Waguespack, L.; Zhang, M.: Computing Curricula 2020: Paradigms for Future Computing Curricula, tech. rep., New York: ACM/IEEE, 2020, DOI https://doi.org/10.1145/3467967 (last check 2025-04-28).

[De23] Denny, P.; Leinonen, J.; Prather, J.; Luxton-Reilly, A.; Amarouche, T.; Becker, B. A.; Reeves, B. N.: Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize AI Code Generators, 2023, arXiv: 2307.16364 [cs.HC] https://doi.org/10.48550/arXiv.2307.16364 (last check 2025-04-28).

[Du86] Du Boulay, B.: Some difficulties of learning to program. In: Journal of Educational Computing Research 2 (1), pp. 57–73, 1986, DOI: 10.2190/3LFX-9RRF-67T8-UVK9 (last check 2025-04-28).

[ER16] Ebert, M.; Ring, M.: A presentation framework for programming in programing lectures. In: Proc. EDUCON. IEEE, pp. 369–374, 2016, DOI 10.1109/EDUCON.2016.7474580 (last check 2025-04-28).

[Ge23] Geng, C.; Zhang, Y.; Pientka, B.; Si, X.: Can ChatGPT Pass An Introductory Level Functional Language Programming Course?, 2023, arXiv: 2305.02230 [cs.CY], https://doi.org/10.48550/arXiv.2305.02230 (last check 2025-04-28).

[Gi24] Gill, S. S.; Xu, M.; Patros, P.; Wu, H.; Kaur, R.; Kaur, K.; Fuller, S.; Singh, M.; Arora, P.; Parlikad, A. K.; Stankovski, V.; Abraham, A.; Ghosh, S. K.; Lutfiyya, H.; Kanhere, S. S.; Bahsoon, R.; Rana, O.; Dustdar, S.; Sakellariou, R.; Uhlig, S.; Buyya, R.: Transformative effects of ChatGPT on modern education: Emerging Era of AI Chatbots. In: Internet of Things and Cyber-Physical Systems 4, pp. 19–23, 2024, DOI: 10.1016/j.iotcps.2023.06.002 (last check 2025-04-28).

[GKR24] Grande, V.; Kiesler, N.; Rodriguez, M. A. F.: 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. ITiCSE 2024, ACM, Milan, Italy, 2024, DOI: 10.1145/3649217.3653624 (last check 2025-04-28).

[HW24] Haindl, P.; Weinberger, G.: Students’ Experiences of Using ChatGPT in an Undergraduate Programming Course. IEEE Access 12, pp. 43519–43529, 2024, DOI: 10.1109/ACCESS. 2024.3380909 (last check 2025-04-28).

[Ja25] Jacobs, S.; Peters, H.; Jaschke, S.; Kiesler, N.: Unlimited Practice Opportunities: Automated Generation of Comprehensive, Personalized Programming Tasks. In: Proceedings of the 2025 on Innovation and Technology in Computer Science Education V. 1. ITiCSE 2025, Preprint available: https://doi.org/10.48550/arXiv.2503.11704 (last check 2025-04-28), Association for Computing Machinery, Nijmegen, 2025, DOI: 10.1145/3724363.3729089 (not available 2025-04-28).

[Je22] Jeuring, J.; Keuning, H.; Marwan, S.; Bouvier, D.; Izu, C.; Kiesler, N.; Lehtinen, T.; Lohr, D.; Peterson, A.; Sarsa, S.: 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. ITiCSE-WGR ’22, ACM, Dublin, Ireland, pp. 95–115, 2022, DOI: 10.1145/3571785.3574124 (last check 2025-04-28).

[JGK24] Jeuring, J.; Groot, R.; Keuning, H.: What Skills Do You Need When Developing Software Using ChatGPT? (Discussion Paper). In: Proceedings of the 23rd Koli Calling International Conference on Computing Education Research. Koli Calling ’23, Association for Computing Machinery, New York, NY, USA, pp. 1–6, 2024, ISBN: 9798400716539, DOI: 10.1145/3631802.3631807 (last check 2025-04-28).

[JJ24] Jacobs, S.; Jaschke, S.: Evaluating the Application of Large Language Models to Generate Feedback in Programming Education. In: 2024 IEEE Global Engineering Education Conference (EDUCON). Pp. 1–5, 2024, DOI: 10.1109/EDUCON60312.2024.10578838 (last check 2025-04-28).

[JLC22] Jayagopal, D.; Lubin, J.; Chasins, S. E.: Exploring the learnability of program synthesizers by novice programmers. In: Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology, pp. 1–15, 2022, https://doi.org/10.1145/3526113.3545659 (last check 2025-04-28).

[Jo24] Joshi, I.; Budhiraja, R.; Dev, H.; Kadia, J.; Ataullah, M. O.; Mitra, S.; Akolekar, H. D.; Kumar, D.: ChatGPT in the Classroom: An Analysis of Its Strengths and Weaknesses for Solving Undergraduate Computer Science Questions. In: Proc. of the 55th ACM Technical Symposium on Computer Science Education V. 1. ACM, New York, pp. 625–631, 2024, DOI: 10.1145/3626252.3630803 (last check 2025-04-28).

[Ka24] Kazemitabaar, M.; Ye, R.; Wang, X.; Henley, A. Z.; Denny, P.; Craig, M.; Grossman, T.: CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and Educator Needs. arXiv preprint arXiv:2401.11314, 2024, CHI '24: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, Article No.: 650, Pages 1 – 20, https://doi.org/10.1145/3613904.3642773 (last check 2025-04-28).

[Ke24] Keuning, H.; Luxton-Reilly, A.; Ott, C.; Petersen, A.; Kiesler, N.: Goodbye Hello World - Research Questions for a Future CS1 Curriculum. In: Proceedings of the 24th Koli Calling International Conference on Computing Education Research. Koli Calling ’24, Association for Computing Machinery, 2024, ISBN: 9798400710384, DOI: 10.1145/3699538.3699591 (last check 2025-04-28).

[Ki20] Kiesler, N.: Towards a Competence Model for the Novice Programmer Using Bloom’s Revised Taxonomy – An Empirical Approach. In: Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education. ITiCSE ’20, ACM, Trondheim, Norway, pp. 459–465, 2020, DOI: 10.1145/3341525.3387419 (last check 2025-04-28).

[Ki22] Kiesler, N.: Kompetenzförderung in der Programmierausbildung durch Modellierung von Kompetenzen und informativem Feedback, Fachbereich Informatik und Mathematik, Dissertation, Frankfurt am Main: Johann Wolfgang Goethe-Universität, 2022.

[KI23] Kiesler, N.; Impagliazzo, J.: Industry's Expectations of Graduate Dispositions. In: 2023 IEEE Frontiers in Education Conference (FIE). Pp. 1–5, 2023, DOI: 10.1109/FIE58773.2023.10343406 (last check 2025-04-28).

[Ki23] Kiesler, N.: Beyond the Textbook: Rethinking Students' Competencies in the LLM Era, Generative AI: Implications for Teaching and Learning, Uppsala, Sweden, 2023, DOI: 10.13140/RG.2.2.28355.37922/1 (last check 2025-04-28).

[Ki24a] Kiesler, N.: Modeling Programming Competency: A Qualitative Analysis. Springer International Publishing, Cham, 2024, ISBN: 978-3-031-47148-3.

[Ki24b] Kiesler, N.; Scholz, I.; Albrecht, J.; Stappert, F.; Wienkop, U.: Novice Learners of Programming and Generative AI - Prior Knowledge Matters. In: Proceedings of the 24th Koli Calling International Conference on Computing Education Research. Koli Calling '24, Association for Computing Machinery, Koli, Finland, 2024, ISBN: 979-8-4007-1038- 4/24/11, DOI: 10.1145/3699538.3699580 (last check 2025-04-28).

[Ki25] Kiesler, N.; Smith, J.; Leinonen, J.; Fox, A.; MacNeil, S.; Ihantola, P.: The Role of Generative AI in Software Student CollaborAItion, accepted at ITiCSE 2025: https://doi.org/10.1145/3724363.3729040, Preprint available: 2025, arXiv: 2501.14084 [cs.SE], url: https://arxiv.org/abs/2501.14084 (last check 2025-04-28).

[KJH18] Keuning, H.; Jeuring, J.; Heeren, B.: A Systematic Literature Review of Automated Feedback Generation for Programming Exercises. In: ACM Trans. Comput. Educ. 19 (1), 2018, DOI: 10.1145/3231711 (last check 2025-04-28).

[KLK24] Kiesler, N.; Lohr, D.; Keuning, H.: Exploring the Potential of Large Language Models to Generate Formative Programming Feedback. In: 2023 IEEE Frontiers in Education Conference (FIE). Pp. 1–5, 2024, DOI: 10.1109/FIE58773.2023.10343457 (last check 2025-04-28).

[KS23a] Kiesler, N.; Schiffner, D.: Large Language Models in Introductory Programming Education: ChatGPT's Performance and Implications for Assessments, 2023, DOI: 10.48550/arXiv.2308.08572, arXiv: 2308.08572 [cs.SE] (last check 2025-04-28).

[KS23b] Kiesler, N.; Schiffner, D.: Why We Need Open Data in Computer Science Education Research. In: Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1. ITiCSE 2023, ACM, Turku, Finland, pp. 348–353, 2023, DOI: 10.1145/3587102.3588860 (last check 2025-04-29).

[Le23a] Leinonen, J.; Denny, P.; MacNeil, S.; Sarsa, S.; Bernstein, S.; Kim, J.; Tran, A.; Hellas, A.: Comparing Code Explanations Created by Students and Large Language Models. In: Proc. ITiCSE. Pp. 124–130, 2023, DOI: 10.1145/3587102.3588785 (last check 2025-04-29).

[Le23b] Leinonen, J.; Hellas, A.; Sarsa, S.; Reeves, B.; Denny, P.; Prather, J.; Becker, B. A.: Using Large Language Models to Enhance Programming Error Messages. In: Proc. SIGCSE. ACM, 2023, DOI: 10.1145/3545945.3569770 (last check 2025-04-29).

[Li24a] Liffiton, M.; Sheese, B. E.; Savelka, J.; Denny, P.: CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes. In: Proceedings of the 23rd Koli Calling International Conference on Computing Education Research. Koli Calling ’23, ACM, Koli, Finland, 2024, DOI: 10.1145/3631802.3631830 (last check 2025-04-29).

[Li24b] Liu, R.; Zenke, C.; Liu, C.; Holmes, A.; Thornton, P.; Malan, D. J.: Teaching CS50 with AI: Leveraging Generative Artificial Intelligence in Computer Science Education. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1. SIGCSE 2024, ACM, New York, pp. 750–756, 2024, DOI: 10.1145/3626252.3630938 (last check 2025-04-29).

[LKK25] Lohr, D.; Keuning, H.; Kiesler, N.: You’re (Not) My Type – Can LLMs Generate Feedback of Specific Types for Introductory Programming Tasks? In: Journal of Computer Assisted Learning, 2025, DOI: 10.1111/jcal.13107 (last check 2025-04-29).

[Lo24] Lohr, D.; Kiesler, N.; Keuning, H.; Jeuring, J.: “Let Them Try to Figure It Out First” – Reasons Why Experts (Do Not) Provide Feedback to Novice Programmers. In: Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education V. 1. ITiCSE 2024, ACM, Milan, Italy, 2024, DOI: 10.1145/3649217.3653530 (last check 2025-04-29).

[Lu16] Luxton-Reilly, A.: Learning to Program is Easy. In: Proc. ITiCSE. Pp. 284–289, 2016, DOI: 10.1145/2899415.2899432 (last check 2025-04-29).

[Lu18] Luxton-Reilly, A.; Simon; Albluwi, I.; Becker, B. A.; Giannakos, M.; Kumar, A. N.; Ott, L.; Paterson, J.; Scott, M. J.; Sheard, J.; Szabo, C.: Introductory Programming: A Systematic Literature Review. In: Proc. ITiCSE. ACM, New York, pp. 55–106, 2018, DOI: 10.1145/3293881.3295779 (last check 2025-04-29).

[Ma00] Mayring, P.: Qualitative content analysis forum qualitative sozialforschung. In: Forum: qualitative social research 1 (2), 2000, DOI: https://doi.org/10.17169/fqs-1.2.1089 (last check 2025-04-29).

[Ma23] MacNeil, S.; Tran, A.; Hellas, A.; Kim, J.; Sarsa, S.; Denny, P.; Bernstein, S.; Leinonen, J.: Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book. In: Proc. SIGCSE TS. Pp. 931–937, 2023, DOI: 10.1145/3545945.3569785 (last check 2025-04-29).

[Ma24] MacNeil, S.; Leinonen, J.; Denny, P.; Kiesler, N.; Hellas, A.; Prather, J.; Becker, B. A.; Wermelinger, M.; Reid, K.: Discussing the Changing Landscape of Generative AI in Computing Education. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2. ACM, New York, p. 1916, 2024, DOI: 10.1145/3626253.3635369 (last check 2025-04-29).

[Ma81] Mandl, H.: Zur Psychologie der Textverarbeitung: Ansätze, Befunde, Probleme. Urban & Schwarzenberg, München, 1981.

[MCK24] Ma, B.; Chen, L.; Konomi, S.: Enhancing Programming Education with ChatGPT: A Case Study on Student Perceptions and Interactions in a Python Course, 2024, arXiv: 2403.15472 [cs.CY], url: https://arxiv.org/abs/2403.15472, DOI: https://doi.org/10.48550/ arXiv.2403.15472 (last check 2025-04-29). In: Olney, Andrew M.; Chounta, Irene-Angelic; Liu, Zitao; Santos, Olga C.; Bittencourt, Ig Ibert (Ed.): Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky. 25th International Conference, AIED 2024, Recife, Brazil, July 8–12, 2024, Proceedings, Part I, p. 113-126, Springer, Cham, 2024, DOI: https://doi.org/10.1007/978-3-031-64315-6 (last check 2025-04-29).

[Na06] Narciss, S.: Informatives Tutorielles Feedback: Entwicklungs- und Evaluationsprinzipien auf der Basis instruktionspsychologischer Erkenntnisse. Waxmann Verlag, Münster, 2006.

[Op23] OpenAI: Prompt Engineering, https://platform.openai.com/docs/guides/prompt-engineering , 2023 (last check 2025-04-29).

[Pe16] Petersen, A.; Craig, M.; Campbell, J.; Tafliovich, A.: Revisiting why students drop CS1 In: Proc. Koli Calling. Pp. 71–80, 2016, DOI: 10.1145/2999541.2999552 (last check 2025-04-29).

[Ph23] Phung, T.; Cambronero, J.; Gulwani, S.; Kohn, T.; Majumdar, R.; Singla, A.; Soares, G.: Generating High-Precision Feedback for Programming Syntax Errors using Large Language Models, 2023, arXiv: 2302.04662 [cs.PL], DOI: https://doi.org/10.48550/arXiv.2302.04662 (last check 2025-04-29). In: Proceedings of the 16th International Conference on Educational Data Mining, EDM 2023, Bangalore, India, https://educationaldatamining.org/edm2023/proceedings/2023.EDM-short-papers.37/2023.EDM-short-papers.37.pdf (last check 2025-04-29).

[Pr23a] Prather, J.; Denny, P.; Leinonen, J.; Becker, B. A.; Albluwi, I.; Caspersen, M. E.; Craig, M.; Keuning, H.; Kiesler, N.; Kohn, T.; Luxton-Reilly, A.; MacNeil, S.; Petersen, A.; Pettit, R.; Reeves, B. N.; Savelka, J.: Transformed by Transformers: Navigating the AI Coding Revolution for Computing Education: An ITiCSE Working Group Conducted by Humans. In: Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2. ACM, New York, pp. 561–562, 2023, DOI: 10.1145/3587103.3594206 (last check 2025-04-29).

[Pr23b] Prather, J.; Denny, P.; Leinonen, J.; Becker, B. A.; Albluwi, I.; Craig, M.; Keuning, H.; Kiesler, N.; Kohn, T.; Luxton-Reilly, A.; MacNeil, S.; Petersen, A.; Pettit, R.; Reeves, B. N.; Savelka, J.: 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. ACM, New York, pp. 108–159, 2023, DOI: 10.1145/3623762.3633499 (last check 2025-04-29).

[Pr23c] Prather, J.; Reeves, B. N.; Denny, P.; Becker, B. A.; Leinonen, J.; Luxton-Reilly, A.; Powell, G.; Finnie-Ansley, J.; Santos, E. A.: “It’s Weird That It Knows What I Want”: Usability and Interactions with Copilot for Novice Programmers. ACM Trans. Comput.- Hum. Interact. 2023, DOI: 10.1145/3617367 (last check 2025-04-29).

[Pr24] Prather, J.; Leinonen, J.; Kiesler, N.; Benario, J. G.; Lau, S.; MacNeil, S.; Norouzi, N.; Opel, S.; Pettit, V.; Porter, L.; Reeves, B. N.; Savelka, J.; Smith, D. H.; Strickroth, S.; Zingaro, D.: How Instructors Incorporate Generative AI into Teaching Computing. In: Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education V. 2. Pp. 771–772, 2024, ISBN: 9798400706035, DOI: 10.1145/3649405.3659534 (last check 2025-04-29).

[Pr25] Prather, J.; Leinonen, J.; Kiesler, N.; Gorson Benario, J.; Lau, S.; MacNeil, S.; Norouzi, N.; Opel, S.; Pettit, V.; Porter, L.; Reeves, B. N.; Savelka, J.; Smith, D. H.; Strickroth, S.; Zingaro, D.: Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools. In: 2024 Working Group Reports on Innovation and Technology in Computer Science Education. ITiCSE 2024, Association for Computing Machinery, Milan, Italy, pp. 300–338, 2025, isbn: 9798400712081, doi: 10.1145/3689187.3709614 (last check 2025-04-29).

[Ra21] Raj, R.; Sabin, M.; Impagliazzo, J.; Bowers, D.; Daniels, M.; Hermans, F.; Kiesler, N.; Kumar, A. N.; MacKellar, B.; McCauley, R.; Nabi, S. W.; Oudshoorn, M.: Professional Competencies in Computing Education: Pedagogies and Assessment. In: Proceedings of the 2021 Working Group Reports on Innovation and Technology in Computer Science Education. ACM, New York, pp. 133–161, 2021, DOI: 10.1145/3502870.3506570 (last check 2025-04-29).

[RD39] Roethlisberger, F. J.; Dickson, W. J.: Management and the Worker. Harvard University Press, Cambridge, 1939.

[RHG24] Rogers, M. P.; Hillberg, H. M.; Groves, C. L.: 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. SIGCSE 2024, ACM, Portland, OR, USA, pp. 1147– 1153, 2024, DOI: 10.1145/3626252.3630784 (last check 2025-04-29).

[RKJ23] Roest, L.; Keuning, H.; Jeuring, J.: Next-Step Hint Generation for Introductory Program- ming Using Large Language Models, 2023, arXiv: 2312.10055 [cs.CY], DOI: https://doi.org/10.48550/arXiv.2312.10055 (last check 2025-04-29). In: ACE '24: Proceedings of the 26th Australasian Computing Education Conference, p. 144 - 153, DOI: https://doi.org/10.1145/3636243.363625 (last check 2025-04-29).

[Sa22] Sarsa, S.; Denny, P.; Hellas, A.; Leinonen, J.: Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. In: Proc. ICER. ACM, 2022, DOI: 10.1145/3501385.3543957 (last check 2025-04-29).

[Sa23] Savelka, J.; Agarwal, A.; Bogart, C.; Sakr, M.: Large Language Models (GPT) Struggle to Answer Multiple-Choice Questions about Code, 2023, arXiv: 2303.08033 [cs.CL], DOI: https://doi.org/10.48550/arXiv.2303.08033 (last check 2025-04-29). https://www.scitepress.org/Papers/2023/119969/119969.pdf (last check 2025-04-29).

[SB22] Strickroth, S.; Bry, F.: The Future of Higher Education is Social and Personalized! Experience Report and Perspectives. In: Proceedings of the 14th International Conference on Computer Supported Education – Vol. 1: CSEDU, pp. 389–396, 2022, DOI: 10.5220/0011087700003182 (last check 2025-04-29).

[Sh08] Shute, V. J.: Focus on formative feedback. In: Review of Educational Research 78 (1), 2008, p. 153-189, https://www.jstor.org/stable/40071124 (last check 2025-04-29).

[SK24a] Scholl, A.; Kiesler, N.: Data: Analyzing Chat Protocols of Novice Programmers Solving Introductory Programming Tasks with ChatGPT, 2024, DOI: 10.17605/OSF.IO/WBKQV, url: osf.io/wbkqv (last check 2025-04-29).

[SK24b] Scholl, A.; Kiesler, N.: Data: How Novice Programmers Use and Experience ChatGPT when Solving Programming Exercises in an Introductory Course, 2024, DOI: 10.17605/OSF.IO/6EN4Z, url: osf.io/6en4z (last check 2025-04-29).

[SK24c] Scholl, A.; Kiesler, N.: How Novice Programmers Use and Experience ChatGPT when Solving Programming Exercises in an Introductory Course. In: 2024 IEEE Frontiers in Education Conference (FIE). IEEE Computer Society, Los Alamitos, CA, USA, pp. 1–9, 2024, DOI: 10.1109/FIE61694.2024.10893442 (last check 2025-04-29).

[SS86] Spohrer, J. C.; Soloway, E.: Novice mistakes: Are the folk wisdoms correct? In: Communications of the ACM 29 (7), pp. 624–632, 1986, DOI: 10.1145/6138.6145 (last check 2025-04-29).

[SSK24] Scholl, A.; Schiffner, D.; Kiesler, N.: Analyzing Chat Protocols of Novice Programmers Solving Introductory Programming Tasks with ChatGPT. In: Schulz, S.; Kiesler, N. (Eds.): P356 - DELFI 2024 - Die 22. Fachtagung Bildungstechnologien, p. 63–79, 2024, DOI: 10.18420/delfi2024_05 (last check 2025-04-29).

[Su24] Sun, D.; Cao, Y.; Xu, F.; Zhu, C.: Exploring the Role of ChatGPT in Undergraduate Programming Education: A Fine-Grained Analysis of Students' Behaviors and Inquiry Patterns. In: 2024 IEEE Frontiers in Education Conference (FIE), 2024 (not available at 2025-04-29).

[Ta24] Taylor, A.; Vassar, A.; Renzella, J.; Pearce, H.: dcc –help: Transforming the Role of the Compiler by Generating Context-Aware Error Explanations with Large Language Models. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1. ACM, New York, pp. 1314–1320, 2024, DOI: 10.1145/3626252.3630822 (last check 2025-04-29).

[VZG22] Vaithilingam, P.; Zhang, T.; Glassman, E. L.: Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. In: CHI EA '22: CHI Conference on Human Factors in Computing Systems Extended Abstracts, Article No.: 332, ACM, New York, pp. 1–7, 2022, DOI: https://doi.org/10.1145/3491101.3519665 (last check 2025-04-29).

[WCL07] Whalley, J.; Clear, T.; Lister, R.: The many ways of the Bracelet project. BACIT, 2007, https://opus.lib.uts.edu.au/handle/10453/5747 (last check 2025-04-29).

[Wh23] White, J.; Fu, Q.; Hays, S.; Sandborn, M.; Olea, C.; Gilbert, H.; Elnashar, A.; Spencer-Smith, J.; Schmidt, D. C.: A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT, 2023, arXiv: 2302.11382 [cs.SE], https://doi.org/10.48550/arXiv.2302.11382 (last check 2025-04-29). In: PLoP '23: Proceedings of the 30th Conference on Pattern Languages of Programs, Article No.: 5, p. 1 – 31, https://dl.acm.org/doi/10.5555/3721041.3721046 (last check 2025-04-29).

[Xu24] Xue, Y.; Chen, H.; Bai, G. R.; Tairas, R.; Huang, Y.: Does ChatGPT Help With Introductory Programming? An Experiment of Students Using ChatGPT in CS1. In: Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training. ICSE-SEET '24, Association for Computing Machinery, Lisbon, Portugal, pp. 331–341, 2024, DOI: 10.1145/3639474.3640076 (last check 2025-04-29).

[Za19] Zawacki-Richter, O.; Marin, V. I.; Bond, M.; Gouverneur, F.: Systematic review of research on artificial intelligence applications in higher education–where are the educators? In: International Journal of Educational Technology in Higher Education 16 (1), pp. 1–27, 2019, DOI: https://doi.org/10.1186/s41239-019-0171-0 (last check 2025-04-29).

[Zh22a] Zhai, X.: ChatGPT User Experience: Implications for Education. 2022, DOI: http://dx.doi.org/10.2139/ssrn.4312418 (last check 2025-04-29).

[Zh22b] Zhang, J.; Cambronero, J.; Gulwani, S.; Le, V.; Piskac, R.; Soares, G.; Verbruggen, G.: Repairing Bugs in Python Assignments Using Large Language Models. arXiv preprint arXiv:2209.14876, 2022, DOI: https://doi.org/10.48550/arXiv.2209.14876 (last check 2025-04-29).