Leveraging ChatGPT for Adaptive Learning through Personalized Prompt-based Instruction: A CS1 Education Case Study

SHARAREH ALIPOUR*, Tehran Institute for Advanced Studies(TeIAS), Iran
MOHAMMAD ABOLNEJADIAN* and KAMYAR TAEB*, Sharif University of Technology, Iran

In this research paper, we introduced high school students to introductory programming with Python using a custom learning platform that leverages ChatGPT to generate personalized learning materials based on each student's educational background. The platform features topics and subtopics, each supported by prompts for Explanation, Example, Exercise, and Exercise Solution, with a context-setting prompt tailored to individual students' backgrounds while respecting their privacy.

The case study brought up compelling insights. Students exhibited heightened engagement, and the lecturers transitioned from being traditional instructors teaching content to becoming mentors who guide students on what to do next, clarifying misunderstandings and addressing potential questions. Furthermore, students gained hands-on programming experience during the learning process, eliminating the traditional post-class experimentation phase.

This innovative approach not only enhances introductory programming education but also suggests a broader application of Large Language Models for personalized learning across diverse fields, providing tailored instruction, and fostering engagement.

CCS Concepts: • **Human-centered computing** → **Empirical studies in HCI**; • **Applied computing** → **Education**.

Additional Key Words and Phrases: LLM, ChatGPT, CS1, Introductory Programming, Course Design, Learning Platform, Prompt Engineering

ACM Reference Format:
Sharareh Alipour, Mohammad Abolnejadian, and Kamyar Taeb. 2023. Leveraging ChatGPT for Adaptive Learning through Personalized Prompt-based Instruction: A CS1 Education Case Study. In . ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Education is a field that is constantly changing due to societal transitions, technological developments, and changes in pedagogical philosophies. How we approach teaching and learning has changed in recent years due to the fusion of cutting-edge technologies and contemporary educational frameworks [2].

Our pursuit to further improve education while leveraging technology began with a vision: to enhance students' classroom engagement, provide hands-on learning experiences, and deliver personalized learning materials tailored to individual educational backgrounds. We recognized the value of maintaining a consistent framework and curriculum while accommodating the diverse needs of our students. In pursuit of this vision, we turned to the power of artificial intelligence (AI) and Large Language Models (LLMs).

Our primary objective was to take advantage of AI and LLMs' capabilities to generate text-based educational materials that adapt to each student's proficiency level and educational background. At the heart of this endeavor was

*All authors contributed equally to this research.

Manuscript submitted to ACM

ChatGPT[16], created by OpenAI, a powerful tool capable of providing educational content[1] enriched with code snippets tailored to the unique needs of each learner[11].

Moreover, This AI model offers several benefits in an educational context. Firstly, it helps students enhance their writing skills by providing suggestions, corrections, and examples of good writing [24]. Secondly, it supports teachers in saving time and reducing workload by generating various educational materials, including lesson plans, activities, projects, quizzes, and grading rubrics [6]. Additionally, it enables personalized and adaptive learning experiences, allowing students to learn at their own pace and level [21]. Lastly, it contributes to expanding students' knowledge and vocabulary by introducing them to new topics and words [8].

ChatGPT is used in various ways by students and instructors[28]. Our research explores using ChatGPT as a teacher who assists lecturers for a Python programming course during class time.

We present three main contributions: first, we design a course curriculum that integrates ChatGPT as a support tool for students and instructors during class time; second, we develop a platform that acts as a medium between students and their AI instructor, giving them their personalized educational material based on the pre-embedded prompts baked into it; finally, we conduct an experiment to compare our course's learning outcomes and satisfaction with conventional programming courses. We analyze the data from our experiment and discuss the benefits and drawbacks of our approach.

2 RELATED WORK

The use of AI-powered tools is growing in both online and face-to-face learning environments. These tools can benefit students in various ways, such as providing them with feedback on their performance, customizing their learning journey, and offering them intelligent tutors that assist them. These tools also help instructors monitor student activity and adapt their teaching accordingly [27]. Moreover, some of these tools employ augmented reality to enable instructors to observe student learning, metacognition, and behaviour in real-time in the classroom [10]. Explore AI's support for instructors in various aspects, including student reviews, grading, feedback, intelligent tutoring systems, and enhancing pedagogical practices and student experiences through AI-powered VR for experiential learning. [3].

Researchers have shown how these models can assist students in various tasks, such as generating code [22, 25], explaining code [13, 15, 17], and creating programming assignments [7, 23]. Investigations have been conducted on how novice programmers can use OpenAI's GPT-3, a large language model (LLM), to answer their programming questions [9].

The research explores the significant impacts of large language models (LLMs) on CS education. The research highlights the potential benefits of using LLMs as teaching aids, such as helping students to code faster and learn programming concepts from AI-generated explanations [14].

A study examined how Copilot performs on CS1 programming problems and how prompt engineering affects its efficacy [5]. A Professor at the University of Pennsylvania used AI extensively in classes by instructing students to employ AI tools for various tasks [18].

There are other studies which explored how AI assistants influence student learning, such as their roles in solving and generating CS problems and providing feedback to students [12, 25].

3 METHODOLOGY

Our research consists of two major phases we present in this paper: course design and experiment.

ChatGPT has both positive and negative impacts on education, mentioned in section 1, which we need to take into account for students and instructors. The existence and capabilities of ChatGPT enable us to design courses that can benefit from it. The traditional teaching methods need to evolve in the future, and as a first step, we have developed a course that uses ChatGPT as a teaching assistant in the classroom. We have designed a Python programming course for beginners. The details of our course and how we use ChatGPT are presented in Section 4.

Next, we conduct an experiment to evaluate how our designed course performs compared to the traditional methods. In our experiments, we teach Python programming courses using our new design and a traditional technique for two groups of students. In Section 5, we present the details of our experiment and how it has been done.

In Section 5.3, we discuss our new method from the perspectives of students and instructors. We also collect the opinions and suggestions of the students. In doing so, we follow the approach of Zastudil et al. [27], where they interviewed 18 people and asked them how they envision using generative AI in computing education, both positively and negatively. We also inquired about their experiences in this class by giving them a questionnaire to gather reliable participant feedback.

We also analyze the outcome of our experimental results, such as the student's grades, the duration of classes, and some other data. Our experiment indicates that the new proposed course design benefits both students and instructors.

4 COURSE DESIGN

The course structure is a novel strategy intended to improve learning by delivering individualized and interactive explanations, examples, and exercises. This section provides a step-by-step breakdown of the design and development of the course, stressing the essential elements and approaches used to guarantee successful pedagogical outcomes.

4.1 Syllabus Extraction and Expansion

The curriculum of prestigious colleges (e.g., MIT [4] and University of Washington [20]) known for their computer science programs served as the model for creating a thorough syllabus, which served as the course's foundation. This initial syllabus served as a framework that was later improved upon and expanded to include more thorough subsections within each topic. Our syllabus is divided into three sections: the first part section is Introduction, Comments, and Variables; the second section is the Condition Statement and Loop Statement; and the third section is Function and Library.

4.2 Prompts Generation

To effectively harness the full potential of ChatGPT, prompts, the input text received by the model, need to be carefully crafted in order to produce desired responses [19]. As for our educational purpose, there are several targets that we want to achieve from ChatGPT, namely adaptive learning, personalized examples and exercises, and individualized instruction [28].

To achieve this goal, within each subsection, a triad of prompts was designed to encompass various dimensions of learning: explanation, example, and exercise. Each prompt was created with the associated topic in mind and was intended to lead students through different programming techniques.

The explanation prompt sought concise yet thorough explanations of the essential concepts. The syntax is a significant aspect to cover in an introductory programming course; Therefore, we added a part to the prompt describing each new concept's structure in Python. An example of an explanation prompt designed to align with the defined purposes can be seen in Figure 1.

```
# Prompts from function section

## Explanation prompt
Explain the concept of functions to me and describe their usages. Also, explain the
structure of functions in Python and explain functions with and without arguments.
## Example prompt
Give me a real-world example of using functions.
## Exercise prompt
Give me an exercise of working with functions in Python
```

Fig. 1. Three prompts from the platform, extracted from the introduction subsection of the Functions section.

The example prompt tries to illustrate the application of these ideas through real-world instances, aiding in visualization and comprehension so students can get a better feel for where they can use this concept.

The Exercise prompt presented challenges reinforcing the learning and fostering practical skills acquisition. For each exercise, the prompt would mention not to give out the answer so that each student could think and solve the movement for themself first, and then another prompt was designed to give out the solution to the earlier exercise.

4.3 Personalization

A central aspiration of this research and course was to provide a tailored and individualized learning experience for each participating student. This was accomplished by generating an initial prompt to set the context for the model, leveraging the Persona Pattern [26], telling the model to act as an introductory programming teacher who teaches Python while explaining the student's related information to the model so that it could give the best answer based on the target learner. This prompt was generated with participants' privacy in mind. No personal information, such as name and location is disclosed with the AI model in this prompt. It is worth mentioning that no unnecessary information was gathered from the participants in the first place, as mentioned in Section 5.1.

This initial prompt's first part was establishing a contextual framework for the subsequent interactions. By telling the model to act as an introductory programming teacher who teaches Python, we set the stage for the model to present itself as a teacher who would be on point with their questions.

A template-based methodology was employed to infuse the personalization layer into the responses. This template was designed to accommodate various parameters related to each participant, such as age, grade, GPA, educational background, and familiarity with programming, particularly in the Python language. These templates would then be given to course participants to fill out, giving the AI model precise information about their academic backgrounds and programming skills. By incorporating these personalized insights, the AI-generated responses were fine-tuned to each participant's unique profile, rendering the instructional content more relevant and relatable.

Combining the initial context-setting prompt, using the Persona Pattern, and the participant-specific templates yielded a personalization strategy that elevated the learning experience. This template and an example of this template filled can be seen in Figure 2.

To better see how this approach helps us to give personalized material to each student, one can refer to Figure 3. Students A and B represent two ends of the spectrum that were in our minds designing this course. Student A is

```
## Initial Prompt Template
Act as an introductory programming teacher who teaches Python. I am a AGE year old student
in GRADE grade from a school with SCHOOL utilities and a CITY prestige city. My GPA is
GPA out of 20, and my math score is MATH out of 20. I got a IQ out of 100 scores on an IQ
test. I PROGRAMMING. Tailor your answers to the specific background that I provided.


## Example filled initial prompt
Act as an introductory programming teacher who teaches Python. I am a 17-year-old student
in eleventh grade from a school with high utilities and a low-prestige city. My GPA is
18.93 out of 20, and my math score is 19.34 out of 20. I got an 85 out of 100 scores on an
IQ test. I don't know programming concepts, neither do I know Python programming. Tailor
your answers to the specific background that I provided.
```

Fig. 2. The initial prompt template and an example of the template, filled by an imaginary student.

from a well-developed city, studies in a decent school, has a high GPA and has prior Python programming knowledge. On the other hand, student B represents the students who may need more help in the course. He or she is from an underdeveloped city, studies in an underprivileged school, does not have a high GPA, and lacks prior Python programming knowledge. By examining ChatGPT's answers to each of these students, we can observe that not only does it try to explain more thoroughly to Student B, but its code snippets are also more advanced for Student A, having multiple arguments and doing a mathematical operation in the function. Moreover, we can see that the Persona Pattern works too, as ChatGPT tries to explain like a lecturer, starting from basic concepts, elaborating on them with examples, and concluding by mentioning their usages.

4.4 Interactive Learning Platform

The next phase of course development involved creating an intuitive platform that allowed students to interact seamlessly with ChatGPT. This platform served as the bridge between students and their AI lecturers. It features a dashboard where students can access predefined prompts based on their selected learning topics. However, it is important to note that the platform operates in a read-only mode, meaning that students cannot ask free-form questions to ChatGPT. To achieve the goal of prompting ChatGPT, we leveraged OpenAI's API using their gpt-3.5-turbo language model. Thanks to ChatGPT's multilingual support, our platform already supports two languages, namely English and Persian. So, in order to generate content in the proper language, each prompt is translated into the mentioned languages.

In a real-world classroom, there is a contextual relevance between each question and answer that goes back and forth between the student and the lecturer. Providing contextual relevance throughout the students' engagement with the material should be an essential part to keep in mind for this platform. To achieve this contextual relevance, the venue was designed to keep track of each topic's specific context. By contextualizing the prompts, the platform enhanced the model's understanding and improved the answers' quality.

To visually represent the platform's functionality, a screenshot of the implemented interface is showcased in Figure 4. Additionally, the platform has been implemented, deployed, and can be accessed at **learnprogrammingwithgpt.com**.

Student A

```
Sure! In Python, a function is a block of code
     that performs a specific task. It helps
     organize your code and make it reusable.
Here's how you can define a function in Python:

def function_name(parameters):
    # code block
- `def` is the keyword used to declare a
     function.
- `function_name` is the name you choose for
     your function.
- `parameters` are optional values that you can
      pass to the function, if needed.

Now, let's say you want to create a function to
      multiply two numbers. Here's an example:

def multiply(num1, num2):
    result = num1 * num2
    return result

In this example, the function name is `multiply
     ` and it takes two parameters, `num1` and
     `num2`.

To use the function, you simply call it by its
     name and provide the required arguments:

result = multiply(5, 6)
print(result)

The output will be `30`, as the function `
     multiply` returns the product of 5 and 6.
     You can also assign the result to a
     variable, as shown in the example above.
```

Student B

```
Defining a function in Python allows you to
     group a sequence of statements together
     and give it a name so that you can reuse
     it later in your code. To define a
     function, you need to use the `def`
     keyword followed by the name you want to
     give to your function, followed by
     parentheses `()`. You can also include
     parameters inside the parentheses if your
     function requires any inputs. Finally, you
      need to end the function definition with
     a colon `:`. Here's an example:

# Function definition
def greet(name):
    print("Hello, " + name + "!")

# Using the function
greet("Alice") # Output: Hello, Alice!
greet("Bob") # Output: Hello, Bob!

In the above example, we defined a function
     called `greet` that takes one parameter `
     name`. Inside the function, it prints a
     greeting message with the name provided.
     When we call the function using `greet("
     Alice")`, it will print "Hello, Alice!".

You can use a function by calling it with the
     function name followed by parentheses `()`.
      If the function requires any input, you
     need to provide that in the parentheses.
     The function will execute the code inside
     its block and may return a value if
     specified.
```

Fig. 3. Comparing two outputs for the explanation prompt in defining a function subsection. Student A has a high GPA, is from a high-prestige city and an equipped school. He or she is also familiar with Python programming to some extent. Conversely, Student B is from a poor city and an under-resourced school. He or she is new to programming and does not have any prior coding knowledge.

4.5 Teacher Role and Student Engagement

In this proposed educational paradigm, the role of the teacher shifted from a traditional lecturer to a facilitator of student-centred learning. With ChatGPT taking on the burden of providing personalized explanations and examples, the

Fig. 4. This illustration captures a scenario where a student has chosen to receive an explanation of the "Complex Conditional Structures" topic. The corresponding response generated by the AI model demonstrates how the platform enables students to engage in topic-specific inquiries and receive relevant, contextualized answers and a way to use codes given by the model. The platform can be accessed at learnprogrammingwithgpt.com

teacher's role transformed into that of a guide and mentor. The teacher will be prepared to answer particular questions, elaborate on complex subjects, and provide extra insights during the learning process.

This method ensured that teachers were available for more individualized and nuanced help while maximizing student engagement and allowing students to participate in the learning process actively.

In conclusion, the course's design thoughtfully combines pre-designed prompts, an initial prompt contextualizing the language model as an introductory programming teacher, learner-specific information, and the teacher's evolving role from lecturer to mentor. This innovative combination not only tries to give out a tailored learning experience but also creates a robust framework for redefining CS1 education and presents a promising way to increase the effectiveness and engagement of teaching.

4.6 Evaluation

In this evaluation, we utilized quizzes as a fundamental programming concept as a test to measure the effectiveness of our teaching approach in imparting essential programming concepts to the students. The quizzes, administered at the end of each section, allowed us to assess the students understanding comprehensively. Analyzing quiz responses provided quantitative insights into student performance. Comparative analysis between students taught through our innovative ChatGPT-based method and those taught through traditional methods provided a perspective on the efficacy of our approach.

5 EXPERIMENT

To empirically evaluate the effectiveness of our proposed method, we designed a comprehensive experiment that compared the outcomes of our adaptive ChatGPT-assisted course with traditional instruction. This section describes the design of the investigation, participant demographics, course implementation, and the evaluation strategy used to determine the effectiveness of both teaching approaches.

5.1 Experiment Setup and Data Collection

To initiate this approach, a comprehensive student profiling process was employed. To gather the essential information, which would be needed to tailor the course based on their background, critical data such as math scores, GPA, prior programming experience, age, and IQ test results were gathered by questionnaire. This data was instrumental in customizing the template prompt, as discussed in section 4.

The selection process for student participation in the study was conducted based on their math scores, average subject scores, and IQ quiz results. It is important to note that students voluntarily provided their data and participated in the experiment. They were assigned to either the traditional teaching group (Group A) or the ChatGPT-assisted teaching group (Group B) to balance these attributes across both groups, ensuring fairness and impartiality. The ethical aspect of data collection was strictly adhered to, and the participants' confidentiality and consent were respected throughout the study. The statistical data of the participants is presented in Table 1. Our standard deviation for all participant's ages is about 2.669.

Table 1. Student's data.

	Group A	**Group B**
Number of participant	36	49
Average age	≈ 15	≈ 16
Average GPA (/20)	≈ 19	≈19.9
Average math grade (/20)	≈19	≈19.9
Lowest score GPA (/20)	16	14.5

5.2 Course Implementation and Platform Introduction

As mentioned earlier, The experiment encompassed two classes, each utilizing a different teaching approach. Group A adhered to the conventional teaching model, with a teacher employing slides and lectures to convey the concepts. Group B experienced the instructional methodology detailed in section 4, where students interacted with the ChatGPT-assisted platform to learn introductory programming concepts. Groups A and B had different instructors. All the classes were taught identical course content based on the syllabus, guaranteeing uniformity in the material covered.

The ChatGPT-assisted class began with an introduction to the implemented platform, accessible at learnprogrammingwithgpt.com. The initial page of the platform expects an ID to enter the venue. Each student received a unique ID before the class, which they could use to access the forum. This ensured that each student started learning with personalized content aligned with their academic background and personal information.

The experiment consisted of three sessions, each spanning 45 minutes. After each session, a quiz is administered to all the classes. These tests assess students' knowledge of the corresponding sections' content and ability to solve problems and code. Both groups of students were not allowed to use ChatGPT during the quiz.

Ultimately, we also asked the students to complete a questionnaire about the course and share their feedback. The main questions in this questionnaire are as follows. The first part of the questionnaire is for both groups.

(1) How satisfied were you with the class? (1-5)
(2) How confident are you that you can solve a basic programming question right now? (1-5)

Table 2. The first part of the questionnaire is for both groups.

Question	T-test
(1) Class Satisfaction	0.018295
(2) Confidence in Basic Programming	0.67668

The second part of the questionnaire is asked from group B.

(1) How much did ChatGPT help you in transferring the content? (a lot, not much, not at all)
(2) will you continue to learn more advanced programming materials by self-study after this class?
(3) Do you think ChatGPT helped you learn more content during the class(Compared to traditional methods)? (yes, It made no difference, no)
(4) Do you think using ChatGPT made the content you learned in this class personalized for you? (yes, no)
(5) Do you prefer using ChatGPT over traditional methods? (yes, no)
(6) Will you use ChatGPT for learning more advanced programming materials in the future? (yes, no)
(7) What are the benefits and drawbacks of interacting with ChatGPT in this class?
(8) Write your criticisms and suggestions for improving programming education with ChatGPT in this class.

Table 3. The average answers of the students of group B for the second part of questions.

Question	Group B
(1) Helpfulness of ChatGPT in Transferring Content	a lot
(2) Intention to Continue Self-Study	yes
(3) Effectiveness of ChatGPT in Learning Content	yes
(4) Personalization of Content with ChatGPT	yes
(5) Preference for ChatGPT over Traditional Methods	yes
(6) Willingness to Use ChatGPT for Advanced Learning	yes

Table 4. Summarize some of the feedback on the benefits and drawbacks of interacting with ChatGPT in the class, along with their criticisms and suggestions for improving programming education using ChatGPT for Q7 and Q8.

Student's comment
The class was valuable and efficient, with proper examples and explanations.
It was effective at advanced levels and could be used to find questions.
ChatGPT provided complete comments and teacher explanations, making the class more organized and efficient.
The quiz in between classes provided good examples and could have improved learning.
The class was organized, with a teacher's basic explanation followed by GPT chat answers and detailed analysis.
Although it had a limited teaching method, it was okay and helped students learn programming skills quickly and easily.

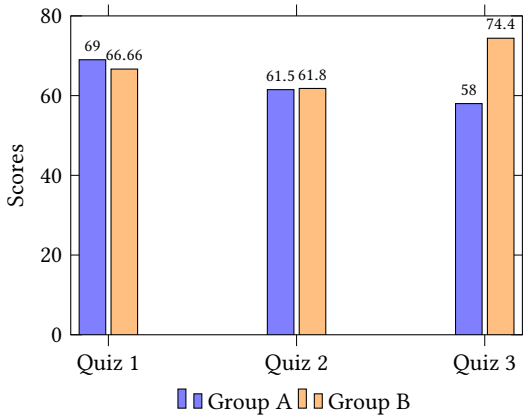

Fig. 5. Comparison of Quiz Scores for Group A and Group B

5.3 Analysis of the experimental results

The average score of students in each quiz is plotted in Figure 5. Our proposed teaching method provides several advantages based on our experiment and feedback from both students and the instructor. Firstly, students showed higher engagement and involvement in the learning process. Secondly, the instructor had more time available to address student questions directly. Additionally, we personalized examples for each student, ensuring tailored learning experiences, which were concise and aided in understanding the explained concepts.

6 REMARKS AND FUTURE WORK

This is a preliminary version of our course design, which can be enhanced with more feedback from the students and instructors by conducting more trials. Furthermore, our platform can be expanded to adapt new features, such as in-app registration, so we can roll out this platform to more users and get more user feedback. Also, to further improve on our material personalization goal, the prompts can be crafted even more, which requires further prompt engineering research in this area.

While this study primarily focuses on high school-aged students, there is a vast potential to extend this teaching method to learners of different age groups, so one can analyze if our desired parameters will be achieved in other age groups as well. Utilizing our ChatGPT-based platform for teaching Python programming suggests its applicability in many educational contexts. Future work could explore integrating this interactive AI tool into various courses. We call for future researchers to expand our questionnaire with additional constructs to improve its generalizability.

It's important to make education accessible to a global audience. Leveraging ChatGPT to support multiple languages and local programming communities can help achieve this goal. By customizing the interface and content to meet the needs of learners from different linguistic and cultural backgrounds, programming education using ChatGPT can be made inclusive and globally applicable.

With the support of ChatGPT, our platform not only accommodates two different languages at the moment but also offers the flexibility to customize both the interface and content, making our educational approach more accessible to a wide global audience of students. Our prompt-based approach enables this seamless customization, ensuring an inclusive learning experience for all.

REFERENCES

[1] David Baidoo-anu and Leticia Owusu Ansah. 2023. Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning. *Journal of AI* 7, 1 (2023), 52 – 62.

[2] Brett A. Becker and Keith Quille. 2019. 50 Years of CS1 at SIGCSE: A Review of the Evolution of Introductory Programming Education Research. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education, SIGCSE 2019, Minneapolis, MN, USA, February 27 - March 02, 2019*, Elizabeth K. Hawthorne, Manuel A. Pérez-Quiñones, Sarah Heckman, and Jian Zhang (Eds.). ACM, 338–344. https://doi.org/10.1145/3287324.3287432

[3] Lijia Chen, Pingping Chen, and Zhijian Lin. 2020. Artificial intelligence in education: A review. *Ieee Access* 8 (2020), 75264–75278.

[4] MIT Open Courseware. 2023. *Introduction To Computer Science And Programming In Python, Syllabus.* Retrieved October 10, 2023 from https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/pages/syllabus/

[5] Paul Denny, Viraj Kumar, and Nasser Giacaman. 2023. Conversing with copilot: Exploring prompt engineering for solving cs1 problems using natural language. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 1136–1142.

[6] Cem Dilmegani. 2023. *ChatGPT Education Use Cases, Benefits Challenges in 2023.* Retrieved August 10, 2023 from https://research.aimultiple.com/chatgpt-education/

[7] James Finnie-Ansley, Paul Denny, Brett A Becker, Andrew Luxton-Reilly, and James Prather. 2022. The robots are coming: Exploring the implications of openai codex on introductory programming. In *Proceedings of the 24th Australasian Computing Education Conference*. 10–19.

[8] Priyanka Gupta. 2023. *What is ChatGPT? What is its Applications in Education?* Retrieved January 31, 2023 from https://www.edtechreview.in/trends-insights/trends/what-is-chatgpt-applications-in-education/

[9] Arto Hellas, Juho Leinonen, Sami Sarsa, Charles Koutcheme, Lilja Kujanpää, and Juha Sorva. 2023. Exploring the Responses of Large Language Models to Beginner Programmers' Help Requests. *arXiv preprint arXiv:2306.05715* (2023).

[10] Kenneth Holstein, Bruce M McLaren, and Vincent Aleven. 2018. Student learning benefits of a mixed-reality teacher awareness tool in AI-enhanced classrooms. In *Artificial Intelligence in Education: 19th International Conference, AIED 2018, London, UK, June 27–30, 2018, Proceedings, Part I 19*. Springer, 154–168.

[11] Lorraine Jacques. 2023. Teaching CS-101 at the Dawn of ChatGPT. *ACM Inroads* 14, 2 (may 2023), 40–46. https://doi.org/10.1145/3595634

[12] Charles Koutcheme. 2022. Towards Open Natural Language Feedback Generation for Novice Programmers using Large Language Models. In *Proceedings of the 22nd Koli Calling International Conference on Computing Education Research*. 1–2.

[13] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 2023. Comparing code explanations created by students and large language models. *arXiv preprint arXiv:2304.03938* (2023).

[14] Stephen MacNeil, Joanne Kim, Juho Leinonen, Paul Denny, Seth Bernstein, Brett A Becker, Michel Wermelinger, Arto Hellas, Andrew Tran, Sami Sarsa, et al. 2023. The Implications of Large Language Models for CS Teachers and Students. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, Vol. 2.

[15] Stephen MacNeil, Andrew Tran, Arto Hellas, Joanne Kim, Sami Sarsa, Paul Denny, Seth Bernstein, and Juho Leinonen. 2023. Experiences from using code explanations generated by large language models in a web software development e-book. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 931–937.

[16] Stephen MacNeil, Andrew Tran, Dan Mogil, Seth Bernstein, Erin Ross, and Ziheng Huang. 2022. Generating Diverse Code Explanations using the GPT-3 Large Language Model. In *ICER 2022: ACM Conference on International Computing Education Research, Lugano and Virtual Event Switzerland, August 7 - 11, 2022, Volume 2*, Jan Vahrenhold, Kathi Fisler, Matthias Hauswirth, and Diana Franklin (Eds.). ACM, 37–39. https://doi.org/10.1145/3501709.3544280

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

[18] Ethan Mollick. 2023. *My class required AI. Here's what I've learned so far.* Retrieved Ferbruary 17, 2023 from https://www.oneusefulthing.org/p/my-class-required-ai-heres-what-ive

[19] Alexandra Neagu. 2023. *How can large language models and prompt engineering be leveraged in Computer Science education?: Systematic literature review.* Retrieved June 28, 2023 from http://resolver.tudelft.nl/uuid:f2d50c24-6f10-4f82-9f69-7edff5ea44ba

[20] University of Washington. 2023. *Syllabus for CSE 140: Data Programming.* Retrieved October 10, 2023 from https://courses.cs.washington.edu/courses/cse140/13wi/

[21] Aiswarya PM. 2023. *ChatGPT Education Use Cases, Benefits Challenges in 2023.* Retrieved April 15, 2023 from https://www.analyticsinsight.net/chatgpt-in-education-sector-benefits-and-challenges/

[22] Ben Puryear and Gina Sprint. 2022. Github copilot in the classroom: learning to code with AI assistance. *Journal of Computing Sciences in Colleges* 38, 1 (2022), 37–47.

[23] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1*. 27–43.

[24] John Villasenor. 2023. *How ChatGPT Can Improve Education, Not Threaten It.* Retrieved Ferbruary 10, 2023 from https://www.scientificamerican.com/article/how-chatgpt-can-improve-education-not-threaten-it/

[25] Michel Wermelinger. 2023. Using GitHub Copilot to solve simple programming problems. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 172–178.

[26] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. *CoRR* abs/2302.11382 (2023). https://doi.org/10.48550/arXiv.2302.11382 arXiv:2302.11382

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

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