



Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning

Ramazan Yilmaz^{*}, Fatma Gizem Karaoglan Yilmaz

Faculty of Science, Department of Computer Technology & Information Systems, Bartın University, Bartın, Turkey

ARTICLE INFO

Keywords:

Generative artificial intelligence
ChatGPT
Programming
Programming learning
Student opinions

ABSTRACT

With the diversification of generative artificial intelligence (AI) applications, the interest in their use in every segment and field of society in recent years has been increasing rapidly. One of these areas is programming learning and program writing processes. One of the generative AI tools used for this purpose is ChatGPT. The use of ChatGPT in program writing processes has become widespread, and this tool has a certain potential in the programming process. However, when the literature is examined, research results related to using ChatGPT for this purpose have yet to be found. The existing literature has a gap that requires exploration. This study aims to analyze the students' perspectives on using ChatGPT in the field of programming and programming learning. The study encompassed a cohort of 41 undergraduate students enrolled in a public university's Computer Technology and Information Systems department. The research was carried out within the scope of the Object-Oriented Programming II course for eight weeks. Throughout the research process, students were given project assignments related to the course every week, and they were asked to use ChatGPT while solving them. The research data was collected using a form consisting of open-ended questions and analyzed through content analysis. The research findings revealed both the advantages and disadvantages of ChatGPT usage, as perceived by the students. The students stated that the main benefits of using ChatGPT in programming learning are providing fast and mostly correct answers to questions, improving thinking skills, facilitating debugging, and increasing self-confidence. On the other hand, the main limitations of using ChatGPT in programming education were getting students used to laziness, being unable to answer some questions, or giving incomplete/incorrect answers, causing professional anxiety in students. Based on the results of the research, it can be said that it would be useful to integrate generative AI tools into programming courses considering the advantages they provide in programming teaching. However, appropriate measures should be taken regarding the limitations it brings. Based on the research findings, several recommendations were proposed regarding the integration of ChatGPT into lessons.

1. Introduction

Augmented Intelligence is a concept used to combine human and artificial intelligence. Augmented Intelligence allows people to make more effective and efficient decisions by working with artificial intelligence technologies. This approach aims to manage and use artificial intelligence systems by humans. In this way, people can gain more confidence and accuracy in decision-making by leveraging artificial intelligence technologies. By combining artificial intelligence and human intelligence. Augmented intelligence can complement each other's shortcomings and achieve better results. Today, especially with artificial intelligence language models such as ChatGPT finding more place in

people's daily lives, the concept of augmented intelligence has started to be talked about in many areas. One of these areas is the programming and software development sector.

Programming education has made significant progress in the last decade. While previously programming was considered a skill that a small group of people should know, today, it has become an essential requirement in many industries. Programming is now a fundamental tool for solving complex problems and providing innovative solutions in many fields, including the healthcare, finance, and transportation sectors. Therefore, programming education is vital for anyone who wants to be successful in many areas, especially in the business world (Nouri et al., 2020).

^{*} Corresponding author.

E-mail addresses: ramazanyilmaz067@gmail.com (R. Yilmaz), gkaraoglan@gmail.com (F.G. Karaoglan Yilmaz).

Despite the increasing interest in programming education, learning programming is generally a challenging and complex process for most people (Tsai, 2019; Yusoff et al., 2020). For this reason, many people who start the process of learning programming may give up learning programming before completing this process successfully. The reasons for this include the need for support and guidance during the program writing process, difficulties in debugging the errors in the program they have written, etc. (Gomes & Mendes, 2014; Sun & Hsu, 2019). Today, generative AI tools such as ChatGPT can provide solutions to the problems that individuals face in programming by giving fast and mostly correct answers. It can almost offer them support and guidance. This study aims to reveal the advantages and limitations of using ChatGPT in the programming learning process from the student's perspective. Upon reviewing the existing literature, no study investigates student views on benefiting ChatGPT in programming learning. In this respect, the research makes a new and original contribution to the literature.

2. Background and literature

2.1. Importance of programming education

Programming education is an important source of skills and knowledge for students today. With today's rapidly developing technology and increasing digitalization in the business world, it can provide a great advantage for future business opportunities (Qian & Lehman, 2017; Topalli & Cagiltay, 2018). Today, rapid developments in technology and digitalization in the business world have made programming education an important source of skills and knowledge for students and a necessary feature to survive in a competitive job market. Learning programming skills equips students with analytical thinking, problem-solving, logical reasoning, and algorithmic thinking skills, enabling them to learn essential skills useful in any area of their lives. These skills can positively affect students' academic achievement, performance at work, and personal development (Agbo et al., 2019; Tikva & Tambouris, 2021). Therefore, programming education is necessary to prepare students for future job opportunities and can increase their professional success. Programming education is more important than providing students with job opportunities and skills. Learning to program helps students develop a deeper understanding of technology and better grasp the inner workings of today's digital world. In addition, the programming learning process can also improve students' creativity and ability to express themselves (Noh & Lee, 2020; Romero et al., 2017). In this way, students consume technology and learn how it works and how they can use it. Programming education can also increase students' self-confidence, allowing them to develop their own projects and generate innovative ideas (Dorotea et al., 2021). Therefore, programming education can play a critical role in their future careers by increasing students' awareness of technology and improving their creativity and ability to express themselves.

Programming training is offered at many educational institutions around the world as a core part of the computer science curriculum. This course introduces students to basic programming concepts, programming languages such as Python, Java, and C++. Students gain the ability to write simple programs using programming tools and learn the steps in the problem-solving process through programming. As programming education progresses, students move towards more complex programming concepts, such as algorithms, data structures, and programming paradigms. This course provides students with a strong foundation in programming principles and problem-solving techniques and increases their ability to develop software for solving real-world problems (Gordon et al., 2022; Siegfried et al., 2021). Students also gain the skills to test, debug and maintain software through programming education. This course helps students understand software development processes for different application areas and prepares them for a successful career in technology.

2.2. Programming learning environments

In recent years, there has been a serious increase in programming languages and tools. This has made it easier for people to learn and use software application development skills (Lindberg et al., 2019; Zinovieva et al., 2021). This trend, especially combined with the increase in popularity of online learning platforms, has made it easier for people to access programming education. For example, platforms such as Codecademy, Udemy, and Coursera offer a wide variety of programming courses from beginner to advanced (Diaz et al., 2021; Oktavia et al., 2018; Sharov et al., 2021). Therefore, the ease of access to programming education tools and resources has increased the number of interested persons.

Programming learning tools and environments aim to make the programming learning process of students more effective and enjoyable. These resources are designed to help students understand programming concepts more easily. Visual programming tools can help students make their programming code easier and more understandable (Sahay et al., 2020). Interactive environments can help students identify and correct their mistakes by allowing them immediate feedback as they write their code. In addition, these resources are designed to support students' self-learning processes (Fagerlund et al., 2021; Saito et al., 2017). In this way, students can progress at their own pace while learning programming and have learning experiences that suit their learning styles (Chen et al., 2023; Tian et al., 2023; Yilmaz & Yilmaz, 2023). There are various programming tools for example *Scratch*, *Alice*, *Code*, *Blockly*, *Code.org*, *App Inventor*, *Micro:bit*, *Greenfoot*, *Logo*, *Codecademy*, *Thimble*, *Lightbot*, *Tynker*, *Blockly Games*, *App Lab*, *Snap!*, *CodeCombat*, *MakeCode*, *Swift Playgrounds*, *Repl.it*, and *Unity* which can be used in programming learning. Their purposes may differ from each other. These programming tools and environments are designed to help students learn programming concepts in a fun and engaging way. Using these resources, students can improve their coding skills and better understand programming concepts.

2.3. ChatGPT as a programming tool

ChatGPT is different from existing programming environments and tools. Because ChatGPT is an artificial intelligence language model that can interact with people using a natural language. Therefore, even people without programming knowledge can easily solve programming problems with ChatGPT (Surameery & Shakor, 2023; Yilmaz & Yilmaz, 2023). Existing programming tools generally consist of software development environments, programming languages, libraries, tools, and other related components (Carver et al., 2007; Ray et al., 2014). These tools usually focus on a specific programming language and require programming knowledge. For example, a Python IDE can only be used for the Python programming language, and the user must be familiar with the basics of Python. However, there are cross IDE that allow develop several languages, for example, Flutter (several platforms) or Microsoft Visual Studio (several languages such as C++, C#, Visual Basic .NET, F#, Java, Python, Ruby). ChatGPT, on the other hand, uses natural language processing and machine learning technologies to understand and correctly respond to commands given by the user in a natural language. ChatGPT has been developed to understand the user's needs and respond accordingly, without using syntax and concepts specific to programming languages (OpenAI, 2023). Therefore, ChatGPT offers a different approach to programming education for both students and teachers (Jalil et al., 2023; Yilmaz & Yilmaz, 2023).

ChatGPT offers different advantages over other programming tools. Here are some advantages that ChatGPT can provide to those who want to learn programming. *Using Natural Language*: ChatGPT can interact with people using natural language processing technologies. This allows even people with no programming knowledge to solve programming problems via ChatGPT. *Easy Access*: ChatGPT can be accessed from any device with an internet connection. Users do not need to install any

special software or tools. *Quick Response*: ChatGPT provides immediate feedback to users by providing quick response. This can speed up the learning process and help students understand better. *Personalized Learning*: ChatGPT can provide users with customized learning experience. It can provide users with learning materials, practices, or examples, follow the learning process and provide customized feedback. *Multi-Language Support*: ChatGPT is available in many different languages. This allows those who want to learn programming to interact with ChatGPT in their native language. *Unlimited Resources*: ChatGPT can provide users with unlimited programming resources using the internet's vast resources. These resources include learning materials, programming examples, online courses, apps, and others. *Clear Explanations*: ChatGPT provides students with clear explanations about programming topics. It helps students to understand the topics better and allows them to spend less time. *Examples and Applications*: ChatGPT provides students with programming examples and applications, enabling them to use theoretical knowledge in practice. This helps students better understand the topics and have a better learning experience. *Inquiry and Search*: ChatGPT allows students to ask questions and search about programming topics. This helps students research topics they are curious about and learn more about. *Debugging and Feedback*: ChatGPT helps students identify and fix programming errors. It also provides feedback to students and can make suggestions for better programming practice. *Advanced Topics*: ChatGPT helps students advance to advanced levels in programming. Students can reach more advanced levels in programming. Research shows that generative AI tools such as ChatGPT also increase students' computational thinking, programming self-efficacy, and motivation (Yilmaz & Yilmaz, 2023). According to Chen et al. (2023), programming training supported by tools such as ChatGPT can effectively improve students' programming skills by providing benefits such as code explanations and debugging. Similarly, Jalil et al. (2023) state that software testing training using ChatGPT can be more effective for learners. In addition, researchers state that ChatGPT can be used effectively in debugging programming and assisting programmers (Surameery & Shakor, 2023; Tian et al., 2023). Some of the advantages of ChatGPT are that it provides fast and mostly correct answers, can be accessed at any time and place, and is easy to access (Kasneji et al., 2023; Lo, 2023; Qureshi, 2023; Tlili et al., 2023). All these are some of ChatGPT's benefits in the programming learning process.

There are also limitations to the use of ChatGPT in programming. Because ChatGPT is a language model, it needs some of the features of traditional programming tools and environments. Some of the limitations are as follows. *Unstructured Learning*: ChatGPT teaches students about a particular programming language or topic in an unstructured way. This can cause students to need help managing the learning process and achieving their learning goals. *Lack of Applications and Need for Secondary Tools and Environments*: ChatGPT does not provide environments or tools for programming applications. Students must use a separate tool or environment for coding in programming languages. *Limited Data Structures and Algorithms*: ChatGPT does not support all data structures and algorithms used in programming languages. Therefore, when students want to learn about a particular data structure or algorithm they want to know, they may need to use different resources or tools. *Lack of User Interface*: ChatGPT does not support interfaces (GUI) used in programming languages. This means students may need to use another tool or environment when they want to develop GUI-based applications in programming languages. Rahman and Watanobe (2023) state in their research that although ChatGPT has several advantages in learning programming, it has limitations, such as a lack of common sense, possible biases, difficulty in complex reasoning, and inability to process visual information. The researchers also note that the ethical implications of ChatGPT (e.g., bias and discrimination, privacy and security, misuse of technology, accountability, transparency, and social impact) are complex and multifaceted.

2.4. Purpose and importance of the research

While ChatGPT offers many advantages to students in terms of programming learning, it also has several limitations in terms of programming learning. Upon reviewing the existing literature, research has yet to be found that examines students' views on the use of ChatGPT in the programming learning process. Knowing the students' views on using ChatGPT in programming learning can provide educators and curriculum developers with many perspectives, especially regarding the integration of ChatGPT into the course. The purpose of this research carried out from this point of view, is to examine the students' opinions about using ChatGPT for programming learning purposes. The findings of this research will provide valuable insights into the integration of generative AI tools like ChatGPT into programming courses. These results will contribute to a better understanding of the potential benefits and implications of utilizing such tools in educational settings, offering guidance for educators and curriculum developers in effectively incorporating them into programming instruction. In line with the main purpose of the research, the following research questions were sought to be answered.

- 1 What are the students' views on the benefits/advantages of using ChatGPT in the programming learning process?
- 2 What are the students' opinions about the limitations/disadvantages of using ChatGPT in the programming learning process?

3. Method

The following section details the research model, study group, data collection tool and data analysis.

3.1. Research model

This study examines the opinions of students who use the ChatGPT tool for programming learning purposes about the use of ChatGPT. For this reason, the case study method was used in the research. A case study is a detailed examination and analysis of an individual, group, organization, or event. These studies delve deeper into a specific topic, explain a phenomenon, or understand what works or doesn't work in a particular situation. This study aimed to examine the students' opinions regarding the use of ChatGPT in the object-oriented programming II course within the scope of the case study. Thus, it was tried to determine the situation of the students regarding the use of ChatGPT in the programming course. Both quantitative and qualitative research approaches were utilized in the study. The researchers created a questionnaire to collect quantitative data and a form consisting of open-ended questions to obtain qualitative data.

3.2. Participants and process

The research was conducted on 41 s-year undergraduate students studying object-oriented programming II at a state university in Turkey's computer technology and information systems department. The study included 33 male and 8 female students who voluntarily participated and responded to the data collection tool. The students who took part in the study were aged between 19 and 25. The students took Introduction to Programming and Algorithms I and II courses in the first year of the undergraduate program and the Object-Oriented Programming I course in the fall semester of the second year of the undergraduate program. From this point of view, the students participating in the research are experienced in programming.

Within the research scope, it aimed to integrate ChatGPT into the course. For this purpose, the object-oriented programming II course was taught for eight weeks in the spring semester. The course was taught in a computer laboratory environment. At the beginning of the course, the instructor explained each week's topic theoretically to the students.

During this lecture process, documents such as presentations and e-books were also utilized. This took approximately 1 h of the course. The students spent the remaining 2 h doing the application assignments related to that week's topic that the instructor projected on the screen. The application assignments were related to that week's topic and involved writing programs using the information that the students had learned in the theoretical course. For example, the following assignment was given on array and collection.

"Market application will be developed using array and collection. You need to write a program to keep the list of products in the market and perform different operations. Requirements;

- **Product Class:**
Create a "Product" class. Use the appropriate data types to store the following properties for each grocery product: Product Name
Manufacturer name
Price information
Production date
It must contain the necessary getter and setter methods.
- **Market Class:** Create a "Market" class. This class should use an array or a suitable collection to store the list of market products. Add the following functionalities:
Add a new product
Removing a product
List all products
Search for a product by manufacturer
Search for a product by production date
- **Test Application:** Write a test application. This application should add, remove and search products using the market class.
Present a menu to the user and allow them to select different actions.
After the user selects an action, perform the action by taking the relevant inputs and show the results to the user.
The program should continue to run until the user logs out.
- **Tip:** You can use a suitable collection like ArrayList or HashMap to store the products.
You can use the Scanner class to get input from the user."

As shown in the example, students are given laboratory assignments based on the information related to the subject learned each week within a scenario. The purpose of the laboratory assignments is to give the students the ability to apply what they have learned about the theoretical topic of that week. In this direction, the instructor presents a problem situation to the student within a scenario. It was stated that the student could benefit from ChatGPT while performing the programming task related to this problem. At the beginning of the research process, that is, at the beginning of the academic term, the instructor explained the ChatGPT tool and how it can be used within the scope of programming education to the students. Students used the ChatGPT tool for various purposes while doing their weekly laboratory assignments.

Students do their weekly laboratory assignments individually using the computers in the computer laboratory and ChatGPT. After completing the assignment, students send the assignment to the course e-mail address. The instructor then evaluates the correctness of the assignment and gives the student a score. The assignments are explained to the students by the instructor after the theoretical lecture. Therefore, the same assignments were given to all students in the class using the same methodology. In other words, the laboratory assignments and the way the course is implemented are similar for all students. In this way, it was aimed to control the factors that could threaten the validity and reliability of the research. The research continued for eight weeks. Students benefited from ChatGPT in the process of doing different application assignments related to the course every week. At the end of the eight-week period, the students' views on the use of ChatGPT in programming education became clear. At the end of this process, the data collection tool developed by the researchers was applied to the students. The data collection tool was prepared in a web-based

environment and the link was sent to the students via e-mail. The students answered the questions in the data collection tool by clicking on the link. By answering open-ended questions, students' opinions were obtained to reveal the benefits and limitations of using ChatGPT in the programming course from the student's perspective.

3.3. Data collection tools

The data collection process involved using a questionnaire and a form consisting of open-ended questions, both designed by the researchers. The questionnaire and semi-structured form questions were prepared by reviewing the literature. Then, it was submitted to the evaluation of five experts in the field of educational technology. In line with the opinions of the experts, the questions were tried to be prepared as short, simple, and clear as possible in order to be suitable for the research and to facilitate students' responses. Students' demographic characteristics were assessed using questions included in the questionnaire. These questions are about age and gender. Questions in the form were developed to determine students' viewpoints on the use of ChatGPT for programming learning purposes. For this purpose, two open-ended questions were included in the semi-structured form. These questions were; a) "What have been the benefits of using ChatGPT while doing the application assignments in the programming course? Explain." b) "What were the limitations/disadvantages of using ChatGPT while doing the application assignments in the programming course? Explain." After the questions were included in the questionnaire form and the semi-structured form was created, the researchers created the questionnaire electronically using Google Forms. Then, during the data collection process, the link to the online form was sent to the students in the class and the students were asked to answer the form using the computers in the laboratory. The results of the content analysis made in line with the questions in the form and the answers given to these questions are given in the findings section of the article.

3.4. Data analysis

The qualitative data obtained from the 41 student participants were subjected to analysis through the content analysis method. This method allowed for the identification of key themes, patterns, and insights within the collected data, enabling a comprehensive exploration and interpretation of the students' experiences and perspectives. Two different researchers took part in the analysis of the qualitative data. Based on the students' responses to each research question, the researchers extracted sub-themes related to the advantages of using ChatGPT and sub-themes related to the limitations/disadvantages of using ChatGPT. Then, the two researchers came together to examine the sub-themes and clarified the sub-themes by reaching a consensus. These sub-themes were written on the student opinion form and guided the researchers in terms of classification in analyzing the students' opinions. After determining the sub-themes, each researcher examined the students' opinions to determine the number of codes (frequency). The use of ChatGPT for programming learning was examined by gathering the opinions of students through a form consisting of open-ended questions. For research reliability purposes, the qualitative data collected from the analysis process were coded by one researcher and cross-checked by a second encoder. The agreement between the two coders was assessed by calculating the percentage of codes they both assigned to the same category (Miles & Huberman, 1994). The coding reliability was 95%, indicating a strong agreement between the two coders. The remaining 5% was resolved through discussion and consensus. After reviewing the students' written explanations, it was apparent that the coding inconsistency arose from some of the responses being classified under various sub-themes. By means of a questionnaire, the study obtained quantitative data from the participants about their gender, age, and other relevant details. Descriptive statistics, particularly the mean, were employed to analyze the gathered data. While interpreting the

qualitative data in the research, quantitative data were used in order to better determine the study group, reveal demographic information and create a student profile.

These interventions were made to ensure the validity, reliability, and generalizability of the study. During participant selection for the research, inclusion criteria were applied to identify students who actively utilized the ChatGPT tool as part of their programming learning process. Thus, it is aimed that the students have a detailed view of the questions in the context. Two researchers coded the students' opinions separately to assess the research's reliability. The reliability coefficient between the two coders was then computed. Experts in educational technologies reviewed the research questions to ensure their validity.

4. Findings

4.1. Opinions on the benefits of using ChatGPT in the learning programming process

Students' views on the benefits of using ChatGPT in the programming learning process and doing application assignments were analyzed. Student opinions are given in Table 1.

Students stated that the use of ChatGPT in the programming process provides many advantages. It is noted that the most essential benefits of ChatGPT in this process are that it responds quickly and effectively to questions, reduces time loss in researching the solution to problems, helps in debugging, gives ideas for solving complex problems, and provides information on details about the subject. Students state that the answers given by ChatGPT make the students think and think about how to integrate the given solution into the project and that it will improve the students' thinking skills. In addition, students see ChatGPT as a teacher and state that they use it to learn the details of the issues that need to be understood about the subject. Some of the student opinions are as follows.

S3: "I can easily learn any information I need. It works great for me in the coding world."

S11: "It is both easier and time-saving to find the answer in one place, rather than constantly researching from different sites."

S13: "ChatGPT can write a desired program partially correctly or completely if the desired program part is described correctly. For this reason, it provides great convenience in programming. Likewise, ChatGPT helps in roadmap when it comes to programming education."

S24: "I can get fast and accurate answers to my questions about software development. It allows me to spend my time practicing coding instead of looking for time and resources to find answers to my questions."

S35: "ChatGPT can answer questions quickly and effectively, thanks to its natural language processing technology. ChatGPT is available at any hour,

Table 1
Opinions on the benefits of using ChatGPT.

Sub-Themes	f
Respond to questions quickly and effectively	39
Preventing wasting time while doing research	38
Help with debugging	35
Supporting the development of thinking skills	34
Providing the opportunity to reach the solution of complex problems	33
Providing simple and understandable teaching of incomprehensible subjects	32
Contributing to improving our programming skills	31
Giving information about the details about the curious subject	28
Being available 24/7	27
Confidence in programming	25
Helping me see alternative solutions to the problem	23
Guiding/giving ideas to the student in the process of making the projects	22
Presenting the solution of problems from different disciplines such as mathematics and physics, which are needed in the program writing process.	19
Providing interaction in our own language thanks to natural language processing technology	18
Demonstrating more accurate and appropriate writing of codes	17
Providing help when I can't continue while writing code	12

any day. Users can ask their questions at any time and get immediate answers. ChatGPT is available in many languages. Users can ask questions and get answers in their native language."

4.2. Views on the disadvantages/limitations of using ChatGPT in the learning programming process

Students' views on the disadvantages/limitations of using ChatGPT in the programming learning process and doing application assignments were analyzed. Student opinions are given in Table 2.

While some of the students stated that there were no disadvantages/limitations of using ChatGPT in the programming process, some students stated that there were various disadvantages/limitations. First of all, one of the points stated by the students is related to the fact that the use of ChatGPT leads the student to be lazy/easy-going and causes occupational anxiety. Some students stated that ChatGPT may not always give correct answers and that they needed more knowledge to answer some questions. A small number of students stated that ChatGPT has aspects for improvement as a programming learning tool/environment. Some students stated that the use of ChatGPT could weaken students' thinking skills and make them dependent on themselves. Some of the student opinions are as follows.

S12: "ChatGPT may not always give correct answers. In particular, it may make mistakes when responding to questions that are misunderstood or contain missing information."

S16: "It is a disadvantage that it negatively affects the algorithmic thinking ability by taking the easy way out."

S23: "I think that it will remove the efficiency of professions. I think that after a while there will be no need for people. In the future, it may even be able to finish coding a website."

S34: "The use of ChatGPT in programming can easily familiarize someone who is just starting to learn programming. In the past, people who entered the programming sector were preferred in the sector because they matured with the experience they gained from the mistakes they made. It is very difficult to make mistakes nowadays because instead of us, there is an artificial intelligence (model) that has already learned (trained) from these mistakes."

S38: "It is not a real programming environment. Therefore, it may be insufficient to enable students to gain real programming experience."

5. Discussion and conclusion

This research was carried out on university students to determine the advantages and limitations of using ChatGPT in the programming learning and program writing processes. While students state that the use of ChatGPT in program writing processes provides many benefits, they also state that it includes some limitations. Students use ChatGPT to get suggestions for solving complex problems in the process of writing a program, learning the subjects they do not know, understanding, and obtaining information about the details of a subject. ChatGPT created a positive perception in the eyes of the students as it gave quick and often effective answers to the questions of the students. However, some students state that ChatGPT sometimes needs to give correct answers or

Table 2
Opinions on disadvantages/limitations of using ChatGPT.

Sub-Themes	f
Can lead the programmer to be lazy/easy	16
Causing occupational anxiety	15
It does not have any disadvantages/limitations	13
May not always give correct answers	12
Not having enough information to answer some questions	9
Sometimes his/her answers are not as desired	9
Lack of a real programming training environment/tool	8
Negatively affect the development of thinking skills	7
May increase self-dependence	5

answer questions. This may be related to the lack of answers to these questions in ChatGPT's existing data sources. However, this may also be related to how students should ask questions to ChatGPT. At this point, the prompt comes to the fore.

A prompt is a sentence or short paragraph that forms the beginning of a task or text to be completed by the language model using artificial intelligence technology. For example, "What are the advantages and disadvantages of using the K-Nearest Neighbor (KNN) Algorithm to classify people by gender?" prompt could help a machine learning model generate text about using the KNN algorithm to classify people by gender. When using an artificial intelligence model such as ChatGPT, it is important to type the correct prompt because the model tries to generate text based on the given prompt (Liu et al., 2023; Reynolds & McDonnell, 2021). Typing the correct prompt can help the model produce the correct results. For example, a false prompt such as "Classification of a human by gender" instead of "Benefits of using the KNN algorithm in classifying people by gender" may cause ChatGPT to respond incorrectly. Also, choosing a prompt can help the model provide helpful or interesting information for the user. For example, the "How to become a better programmer?" prompt can direct ChatGPT to provide relevant and helpful suggestions. Depending on ChatGPT's answers to this question, the user can get in-depth information on the subject by asking more detailed questions (for example, the details of each answer) based on the answers given. When the literature is examined, it has been revealed that prompt type and prompt typing significantly affect individuals' outputs such as writing performance, resource usage strategies, and monitoring strategies (Nückles et al., 2020; Shi et al., 2020). For this reason, it is very important today that teachers who want to integrate artificial intelligence language models such as ChatGPT into their lessons provide their students with "prompt literacy" skills.

The majority of participants in the study reported that incorporating ChatGPT into their programming endeavors enhances their cognitive abilities. By reducing the time spent on coding tasks, students are able to allocate more time towards engaging in algorithmic thinking processes for effective problem-solving. Students think about which questions to ask ChatGPT, how to integrate the answers they receive into their projects, and how they can complete the project holistically by bringing together small code blocks. This can improve students' thinking skills. On the other hand, some students state that the use of ChatGPT will negatively affect their thinking skills, because the answer to the desired question can be obtained from ChatGPT and therefore they cannot develop their thinking skills. This may be true for answers to questions with short and simple solutions. For this reason, it will be useful for teachers to give complex and modular projects to the students, and to ensure that the student tries to reach the whole from modular answers, at the point of development of thinking skills (computational thinking, creativity, algorithmic thinking, cooperation, critical thinking, and problem-solving). When the literature is examined, it has been revealed that complex and unstructured problems are useful in developing students' computational thinking skills (Bai et al., 2021; Jiang et al., 2022).

When the literature is examined, it is seen that research results are confirming this view of the students. In the study conducted by Yilmaz & Karaoglan Yilmaz (2023), it was concluded that the computational thinking, programming self-efficacy, and motivation of the students who received ChatGPT support during the programming learning process were higher than the students who did not receive this support. According to the results of Yilmaz & Karaoglan Yilmaz (2023) study, the use of ChatGPT led to a significant increase in students' creativity, algorithmic thinking, cooperativity, critical thinking, and problem-solving scores. These findings of the study are consistent with the results of our research. Based on the results of our study, students stated that using ChatGPT in the programming process effectively improved their thinking skills. Huang et al. (2023) conducted a study that demonstrated the positive impact of providing AI-based personalized advice and guidance to learners, leading to improved motivation and learning outcomes. Consistent with these findings, our research

concludes that the incorporation of ChatGPT in the programming process enhances students' self-confidence, learning motivation, and code-writing skills. In a study by Qureshi (2023), the performance of an experimental group, which utilized ChatGPT, was compared to a control group that did not use ChatGPT in laboratory assignments for a data structures and algorithms course. The results revealed that the experimental group students who employed ChatGPT obtained higher scores, indicating a clear advantage in their academic performance. Still, there needed to be more consistency and accuracy in the submitted code that affected the overall performance. In this respect, the results of our study are similar to the literature. The research findings suggest that the utilization of generative AI tools, including ChatGPT, in program writing and programming learning processes yields overall benefits. These tools have shown promise in enhancing various aspects of the learning experience, such as improving self-confidence, motivation, code-writing skills, and academic performance. Therefore, incorporating generative AI tools into programming education can be regarded as a valuable approach to enhancing the learning outcomes and effectiveness of programming courses. Therefore, it is useful to integrate these tools into courses. On the other hand, students should also take into account that the codes produced by these tools may be faulty. For this reason, it should be explained to students that they should be made aware of this and that the results produced by generative AI tools should be checked and verified by themselves. Students can reconstruct the outputs produced by generative AI tools and adapt them in their own way. Once again, it is crucial to emphasize the importance of evaluating the ethical appropriateness of results generated by generative AI tools like ChatGPT. This evaluation is closely linked to the development of AI literacy skills among students. It becomes essential to equip students with the knowledge and critical thinking abilities necessary to navigate the ethical considerations and potential biases associated with AI-generated outputs. By providing students with these skills, educators can empower them to make informed decisions and engage responsibly with generative AI tools, fostering a more ethical and responsible use of AI technologies in their academic and professional pursuits. (Kong et al., 2021; Laupichler et al., 2022; Wang et al., 2023). Thus, the integration of generative AI tools such as ChatGPT into courses can be done correctly and effectively. It should be noted that this research has some limitations. First, the research was conducted within the Object Oriented Programming II course scope. Similarities and differences in the results can be examined by examining student opinions in different programming courses (e.g. visual programming, web programming, robotic programming, etc.). The research was conducted on university students. Similar studies can be conducted on students from different levels (e.g. middle school, high school, etc.) to examine student opinions. The implementation period of this research is limited to eight weeks. In future studies, longitudinal studies can be conducted to analyze the opinions of students who use ChatGPT for longer. Differences in the results depending on the duration of use can be determined. To determine the effect of generative AI tools such as ChatGPT on students' learning processes and outcomes, it would be useful to plan experimental studies, especially with experimental and control groups.

Ethical approval

All procedures performed in these studies were in accordance with the APA ethical guidelines, the ethical standards of the institutional research committee, and the 1964 Helsinki declaration and its later amendments.

Code availability

The authors used AMOS functions for their statistical analyses.

Informed consent to participate

All participants gave full informed consent to participate.

Consent for publication

All participants gave consent for their data to be used in publication.

Data availability

The authors are willing to share their data, analytics methods, and study materials with other researchers upon request.

Declaration of competing interest

The authors have no conflicting or competing interests to declare.

References

- Agbo, F. J., Oyeler, S. S., Suhonen, J., & Adewumi, S. (2019, November). A systematic review of computational thinking approach for programming education in higher education institutions. In *Proceedings of the 19th koli calling international conference on computing education research* (pp. 1–10).
- Bai, H., Wang, X., & Zhao, L. (2021). Effects of the problem-oriented learning model on middle school students' computational thinking skills in a python course. *Frontiers in Psychology, 12*, Article 771221.
- Carver, J. C., Kendall, R. P., Squires, S. E., & Post, D. E. (2007, May). Software development environments for scientific and engineering software: A series of case studies. In *29th international conference on software engineering (ICSE'07)* (pp. 550–559). Ieee.
- Chen, E., Huang, R., Chen, H. S., Tseng, Y. H., & Li, L. Y. (2023). *GPTutor: A ChatGPT-powered programming tool for code explanation*. arXiv preprint arXiv:2305.01863.
- Dorotea, N., Piedade, J., & Pedro, A. (2021). Mapping K-12 computer science teacher's interest, self-confidence, and knowledge about the use of educational robotics to teach. *Education Sciences, 11*(8), 443.
- Fagerlund, J., Häkkinen, P., Vesinenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education, 29*(1), 12–28.
- Gomes, A., & Mendes, A. (2014). A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. In *2014 IEEE frontiers in education conference (FIE) proceedings* (pp. 1–8). IEEE.
- Gordon, C., Lysecky, R., & Vahid, F. (2022). Programming learners struggle as much in Python as in C++ or Java. In *2022 ASEE annual conference & exposition*.
- Jalil, S., Rafi, S., LaToza, T. D., Moran, K., & Lam, W. (2023, April). Chatgpt and software testing education: Promises & perils. In *2023 IEEE international conference on software testing, verification and validation workshops (ICSTW)* (pp. 4130–4137). IEEE.
- Jiang, S., Qian, Y., Tang, H., Yalcinkaya, R., Rosé, C. P., Chao, J., & Finzer, W. (2022). Examining computational thinking processes in modeling unstructured data. *Education and Information Technologies, 1*–25. <https://doi.org/10.1007/s10639-022-11355-3>
- Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., ... Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences, 103*, Article 102274.
- Kong, S. C., Cheung, W. M. Y., & Zhang, G. (2021). Evaluation of an artificial intelligence literacy course for university students with diverse study backgrounds. *Computers and Education: Artificial Intelligence, 2*, Article 100026.
- Laupichler, M. C., Aster, A., Schirch, J., & Raupach, T. (2022). Artificial intelligence literacy in higher and adult education: A scoping literature review. *Computers and Education: Artificial Intelligence, 3*, Article 100101.
- Lindberg, R. S., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology, 50*(4), 1979–1995.
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys, 55*(9), 1–35.
- Lo, C. K. (2023). What is the impact of ChatGPT on education? A rapid review of the literature. *Education Sciences, 13*(4), 410.
- Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. Sage Publications.
- Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research & Development, 68*, 463–484.
- Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry, 11*(1), 1–17.
- Nückles, M., Roelle, J., Glogger-Frey, I., Waldeyer, J., & Renkl, A. (2020). The self-regulation-view in writing-to-learn: Using journal writing to optimize cognitive load in self-regulated learning. *Educational Psychology Review, 32*, 1089–1126.
- Oktavia, T., Prabowo, H., & Supangkat, S. H. (2018). The comparison of MOOC (massive open online course) platforms of edx and coursera (study case: Student of programming courses). In *2018 international conference on information management and technology (ICIMTech)* (pp. 339–344). IEEE.
- OpenAI. (2023). *Introducing ChatGPT*. <https://openai.com/blog/chatgpt>.
- Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education, 18*(1), 1–24.
- Qureshi, B. (2023). *Exploring the use of chatgpt as a tool for learning and assessment in undergraduate computer science curriculum: Opportunities and challenges*. arXiv preprint arXiv:2304.11214.
- Rahman, M. M., & Watanobe, Y. (2023). ChatGPT for education and research: Opportunities, threats, and strategies. *Applied Sciences, 13*(9), 5783.
- Ray, B., Posnett, D., Filkov, V., & Devanbu, P. (2014). A large scale study of programming languages and code quality in github. In *Proceedings of the 22nd ACM SIGSOFT international symposium on foundations of software engineering* (pp. 155–165).
- Reynolds, L., & McDonnell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended abstracts of the 2021 CHI conference on human factors in computing systems* (pp. 1–7).
- Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education, 14*(1), 1–15.
- Sahay, A., Indamutsa, A., Di Ruscio, D., & Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms. In *2020 46th euromicro conference on software engineering and advanced applications (SEAA)* (pp. 171–178). IEEE.
- Saito, D., Sasaki, A., Washizaki, H., Fukazawa, Y., & Muto, Y. (2017). Program learning for beginners: Survey and taxonomy of programming learning tools. In *2017 IEEE 9th international conference on engineering education (ICEED)* (pp. 137–142). IEEE.
- Sharov, S., Kolmakova, V., Sharova, T., & Pavlenko, A. (2021). Analysis of MOOC on programming for IT specialist training. *TEM Journal: Technology, Education, Management, Informatics, 10*(4), 1884–1894.
- Shi, B., Huang, L., & Lu, X. (2020). Effect of prompt type on test-takers' writing performance and writing strategy use in the continuation task. *Language Testing, 37*(3), 361–388.
- Siegfried, R. M., Herbert-Berger, K. G., Leune, K., & Siegfried, J. P. (2021). Trends of commonly used programming languages in CS1 and CS2 learning. In *2021 16th international conference on computer science & education (ICCSE)* (pp. 407–412). IEEE.
- Sun, J. C. Y., & Hsu, K. Y. C. (2019). A smart eye-tracking feedback scaffolding approach to improving students' learning self-efficacy and performance in a C programming course. *Computers in Human Behavior, 95*, 66–72.
- Surameery, N. M. S., & Shakor, M. Y. (2023). Use chatgpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering (IJITC), 3*(1), 17–22.
- Tian, H., Lu, W., Li, T. O., Tang, X., Cheung, S. C., Klein, J., & Bissyandé, T. F. (2023). *Is ChatGPT the ultimate programming assistant—How far is it?* arXiv preprint arXiv:2304.11938.
- Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature review. *Computers & Education, 162*, Article 104083.
- Tlili, A., Shehata, B., Adarkwah, M. A., Bozkurt, A., Hickey, D. T., Huang, R., & Agyemang, B. (2023). What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education. *Smart Learning Environments, 10*(1), 15.
- Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education, 120*, 64–74.
- Tsai, C. Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior, 95*, 224–232.
- Wang, B., Rau, P. L. P., & Yuan, T. (2023). Measuring user competence in using artificial intelligence: Validity and reliability of artificial intelligence literacy scale. *Behaviour & Information Technology, 42*(9), 1324–1337.
- Yilmaz, R., & Yilmaz, F. G. K. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence, 4*, Article 100147.
- Yusoff, K. M., Ashaari, N. S., Wook, T. S. M. T., & Ali, N. M. (2020). Analysis on the requirements of computational thinking skills to overcome the difficulties in learning programming. *International Journal of Advanced Computer Science and Applications, 11*(3), 244–253.
- Zinovieva, I. S., Artemchuk, V. O., Iatsyshyn, A. V., Popov, O. O., Kovach, V. O., Iatsyshyn, A. V., ... Radchenko, O. V. (2021, March). The use of online coding platforms as additional distance tools in programming education. In *Journal of physics: Conference series* (Vol. 1840)IOP Publishing, Article 012029, 1.