

Niemann, Sonja; Schmid, Ute

## For What Tasks and Purpose Do Computer Science Students Use Code Generators : Preliminary Results of an Online-Study

In:

Schmid, Ute; Leidner, Jochen L.; Kohlhase, Michael; Wolter, Diedrich (Eds.), Proceedings of the Second Workshop on Artificial Intelligence for Artificial Intelligence Education (AI4AI Learning 2024), Bamberg: University of Bamberg Press, p. 68-80. 2025. DOI: 10.20378/irb-107661

### Bookpart - Published Version

DOI of the Article: 10.20378/irb-108889

Date of Publication: 07.07.2025

### Legal Notice:

This work is protected by copyright and/or the indication of a licence. You are free to use this work in any way permitted by the copyright and/or the licence that applies to your usage. For other uses, you must obtain permission from the rights-holder(s).

This document is made available under the **Creative Commons License CC BY**.



This licence information is available online:  
<https://creativecommons.org/licenses/by/4.0/>

# For What Tasks and Purpose Do Computer Science Students Use Code Generators -- Preliminary Results of an Online-Study

Sonja Niemann<sup>1</sup> und Ute Schmid<sup>1,2</sup> 

bidt – Bayerisches Forschungsinstitut  
für Digitale Transformation, Munich, Germany  
University of Bamberg, Germany

sonja.niemann@bidt.digital  
ute.schmid@uni-bamberg.de

## Abstract

Generative AI is applied in different educational contexts such as essay writing or translation of texts. A specific application of generative models is creation of program code. Educators in schools and universities face the challenge how to assure that students acquire relevant competencies while making use of generative AI tools. In this paper, we present preliminary results of an online study of the use of code generators by Computer Science (CS) students. Beginner students get to learn basic programming concepts and skills, advanced students get to learn how to solve complex programming tasks and create programs which are correct, efficient, modifiable, and well documented. With the presented online study we want to explore which tasks students solve with code generators and with what purpose. 285 students participated in the survey, new beginners as well as advanced students, with the goal to compare students who learned programming basics without code generators with those who always had access to them. Preliminary results show that students are very eager to learn coding skills and want to understand underlying concepts. Students who struggle with coding skills tend to ask code generators more often for explanations and use code generators for the purpose of understanding.

Students who chose not to use code generators are more skeptical of such systems and the mistakes they make. Our empirical results provide insights in how we can support students in acquiring coding skills while using code generators.

**Keywords** Code Generators, Programming Education, Programming Competencies

## Introduction

Concepts and skills in writing programs are the core of computer science education and also a prerequisite for educational programs in artificial intelligence (AI). The current fast development of LLMs as code generators on the one hand provides a powerful tool to support students in many programming related tasks such as code generation from natural language specifications, explanation of code, test case generation, and code repair. On the other hand extensive use of code generators to solve programming assignments has the danger that students do not acquire relevant competencies and skills. That is, a core question for computer science and AI education is: Are students over-relying on code generators and missing critical programming skills? [1].

While first surveys try to gain insight in generative AI use at universities and their struggle with legal questions about copyrights and exams [6][2], there is little to no research about the usage habits of code generators among CS students. Therefore a small survey among CS students was conducted, asking about their code generator use when coding and their trust in such systems. Several aspects of code generator use are covered to understand the user behavior of this particular group. The item groups can be roughly described as followed: Frequency of use and systems used, specific tasks, purpose of use, trust in code generators, technical knowledge and personal goals. Questions about the tasks students give to code generators, allows us insight into the areas we can provide students with additional support.

First results suggest that first year students let code generators generate whole code blocks more often than master students. Master students use

code generators to correct their code more often and both groups use generative AI to explain things they don't understand. This paper will discuss preliminary results of the survey with a focus on the tasks students give to code generators and the purpose they are trying to achieve. The results of our survey will guide the next step of developing interfaces for code generators that are designed to cover the needs of CS students. Instead of condemning the use of code generators for CS students, the opportunity should be seized to harness its potential. To achieve this several options have come to our attention, for example providing explanations via pre-set prompts [4] or repairing buggy code and giving high precision feedback [7][5]. In an exploratory approach an interface combining the findings of the survey and sustainable LLM use will be developed.

## Questionnaire and Sample Group

The Survey follows an exploratory approach since there is little to no comparable work. Other surveys concentrated on students in general, rather than focusing on a specific field of study. Instead, they provided an important overview [6, 2]. Previous work gives us the chance to use similar items and compare our specific sample group of CS Students to the more general sample groups.

Table 1: Item Groups; number of items

Group Title	Number of Items
1. Frequency and Systems	2-4
2. Tasks and Purpose	3
3. Perceived Advantage	1
4. Trust	19
5. Technical Knowledge	6
6. Personal Goals and Skills	4
7. Demographic Data	5

The items of the questionnaire are split into seven item groups that cover different topics. You can see the item groups with the number of items in

Table 1, the groups important for the results presented in this work will be explained closer. The first item group starts with participants selecting if they have heard of and used code generators like ChatGPT for example. Two of the possible answers suggest that they have not heard of or do not use code generators for coding task, those participants will skip a few item groups after the first one. Item block two again is for participants who use code generators for coding, it is about the specific tasks they use code generators for as well as the purpose they hope to achieve. The first question of the item block asks about the setting they use code generators in, at work, at university, for private project, a combination is possible. The second item covers the purpose of the use, giving them three options to choose from, for example 'I use code generators to submit my programming tasks in time.'. The last question of the item block asks participants to rate how often they have specific tasks done by code generators. The answer format is a five point Likert scale, participants indicate the frequency code generators are used for each tasks individually. From item block four on all items are presented to every participant, no matter if they use code generators. As a first approach to the concept of trust in AI and trustworthy AI we used the 'Trust in Automation' questionnaire by Körber [3]. Item block six contains questions about their personal goals and skills, again giving them seven statements they have to agree or disagree to on a Likert scale, for example 'I really want to understand the content of my degree programme'. They also can rate to which extent they find other platforms like Stackoverflow or Youtube helpful. As a last item in this group they can choose to give additional information in a text field. The complete survey can be found in the appendix.

The survey was distributed via several channels, with the goal to reach bachelor and master students in computer science or closely related fields. Several professors, a student council and the 'Junge Gesellschaft für Informatik' were contacted and asked to distribute the survey to the students via email. Students were free to participate. Over four weeks data from 342 participants were collected, 289 fulfilled the criteria of being a CS student or from a closely related fields as well as completing the survey. The group will be split to compare students who have started university before popular LLMs like ChatGPT have been announced and students who have

been exposed to LLMs since day one of their degree. Everyone with more than 4 semester in a bachelor degree will be considered a higher level CS student. The two groups formed will be called 'beginners' and 'advanced' and have roughly the same amount of participants with  $n = 144$  beginners and  $n = 145$  advanced. The beginner group has an average age of 22 with an average of 3 semesters studied. The advanced group has an average age of 26 and an average of 7 semester studied. In a next step students who do not use code generators were separated from both groups, in total 38 students formed the group of non users and leaves the original two groups with  $n = 127$  advanced students and  $n = 124$  beginners.

## Results

The survey covers several variables that are expect to influence each other, coding skills, trust, knowledge about code generator to name a few. Aspects we want to take a closer look at are students mindset towards coding and perceived struggles and goals they personally have. Some of the answers students have written as additional information allow us a more detailed, but subjective look at their experiences.

One rather negative possibility is that students do not see the need in understanding and learning of coding skills anymore and therefore use code generators to finish their assignment in time and get good grades.

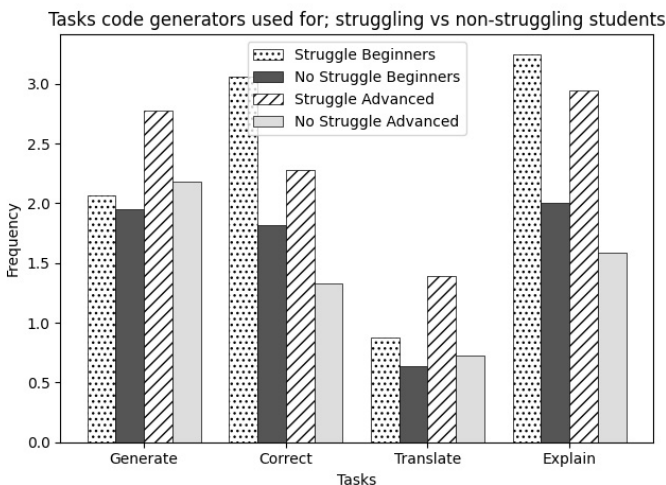


Figure 1 Tasks different student groups use code generators for; Group sizes: SB = 16, NSB = 77, SA = 18, NSA = 84

Out of all 251 students using code generators only ten declared that they do not believe acquiring coding skills is important for them. The most important aspect is understanding the content of their study program, with a mean of 4.47 on a 5 point Likert scale (1 meaning not important at all, 5 meaning very important) over all students using code generators. This is closely followed by a mean of 4,29 for the item 'Learning programming skills is important to me'. Further participants seem to enjoy programming and most of them have a feeling of self-efficacy when asked if they are capable of solving coding tasks on their own. There is a smaller group with  $n = 34$  that is less confident in their programming abilities compared to their fellow students,  $n = 57$  would neither agree nor disagree to the statement and  $n = 160$  felt confident that they were as good as their class mates. We found a difference in the use of code generators in the two extreme groups, the 57 indecisive students are excluded in the following results. As displayed in Figure 1, 195 students are divided into four groups depending on their self-reported coding skills: advanced students ( $n = 102$ ) and beginners ( $n = 93$ ) and each of them are subdivided again into students who report struggles with coding tasks (advanced  $n = 18$ , beginners  $n = 16$ ) and those who are confident in their abilities (advanced  $n = 84$ , beginners  $n = 77$ ). The X-axis show the different tasks students used the code generators for and the y-axis displays the average reported frequency the groups would use code generators for the specific task. Using code generators to explain code is more important to students who struggle with coding tasks, no matter if advanced or beginner. Generating code is more often used by the advanced group that struggle with coding. While beginners who struggle rely on code generators to correct their code. In a second step the purpose these groups want to find in code generators are analyzed. Figure 2 displays the purpose students hope to find in code generators on the x-axis, with the groups and y-axis staying the same as in the first graphic. Understanding tasks is most important to struggling students coinciding with the findings in Figure 1, where explaining was the most used tasks. They also want to learn to code but more often rely on code generators to finish assignments in time than students who are confident in their programming skills.

Participants were free to give additional information about their experience with code generators, or why they chose not to use them.

First some insights into the answers from students who use code generators: Common ground for the fast majority is that code generators can solve easy task or smaller chunks for a bigger task, but are perceived as less helpful with complex tasks or very specific questions. One comment sums up a lot of their experiences 'Copilot is useful to make automated repetitive or standard code blocks, e.g. creating for-loops to alter arrays. For specific or complex tasks it regularly fails.' Some even went further and described code generators as 'sparing partner' or 'pair programmer' that could give ideas how something can be done or one could discuss new approaches with. One last example how students developed their very own approaches to code generators is a person who reported his code documentation got a lot better and more detailed, because he is making sure to give needed details to his LLM of choice.

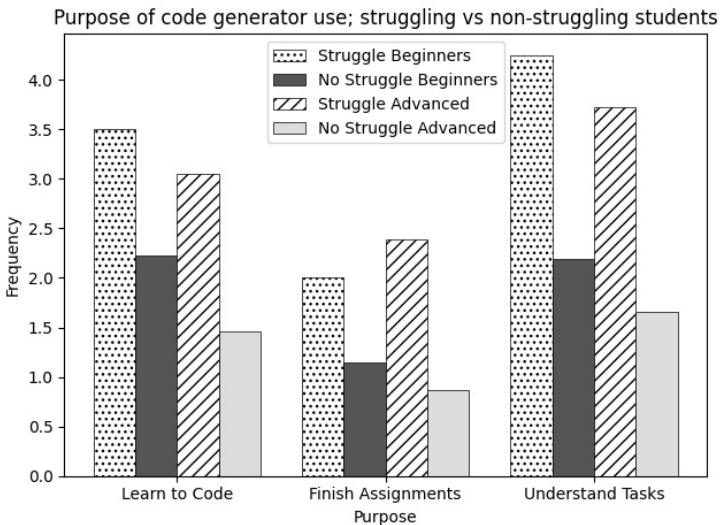


Figure 2 Purpose of code generator use between different student groups;  
Group sizes: SB = 16, NSB = 77, SA = 18, NSA = 84

On the other hand we have CS students who decided not to use code generators, they agree on certain aspects as well. One they do not trust AI to be correct and are afraid to not find the mistakes AI makes. Two they want

to understand the concepts and code they create and believe they wouldn't if they used code generators. And three some are concerned about their academic integrity 'I don't trust them to generate the best solution and I won't learn anything from just copying their solution anyways. I also would rather not get in trouble for academic misconduct.' Some expressed concerns about their privacy and data use in models like ChatGPT and said they don't trust the people behind the systems.

## Discussion

It has to be pointed out that this survey is not representative, a known problem is a selective bias in such surveys. Students who choose to participate in such surveys are likely to be motivated and interested in their study, students who truly struggle are less likely to participate. However, the results can still be used to listen to students' needs and guide future research projects. On the positive side it was shown that the students who participated are eager to learn and understand programming concepts, despite the current discussion that they might rely on generative AI too much. The user patterns, different types of students show, can help us to develop support structures to ensure their needs are met. For example, students who are struggling with their programming skills seek to understand the concepts. In a next phase Interfaces can be designed to explore what type of support gets students the needed information, for example through pre-set prompts [4]. Students would not have to figure out what to ask and how to write a good prompt and could focus on understanding. This group also tends to rely on code generators to hand in assignments in time, therefore are in danger to fall behind with understanding the concepts. Another group that could profit from input are students that up until now do not use code generators. Research should look into trustworthiness of LLMs to enable students to use newest technology without fear. The trust questionnaire in this survey might bring some more insights into this topic.

## References

- [1] Becker, B.A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., Santos, E.A.: Programming is hard - or at least it used to be. In: Proceedings of the 54<sup>th</sup> ACM Technical Symposium on Computer Science Education V. 1. pp. 500–506. ACM, New York, NY, USA (2023).
- [2] Hüsich, M.: Künstliche Intelligenz im Studium: Nicht überall auf Kurs (2024), <https://www.che.de/2024/kuenstliche-intelligenz-im-studium-noch-nicht-in-allen-fachern-auf-kurs/>
- [3] Körber, M.: Theoretical considerations and development of a questionnaire to measure trust in automation. In: Bagnara, S., Tartaglia, R., Albolino, S., Alexander, T., Fujita, Y. (eds.) Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018). pp. 13–30. Springer, Cham (2019)
- [4] Nam, D., Macvean, A., Hellendoorn, V., Vasilescu, B., Myers, B.: Using an LLM to help with code understanding. In: Roychoudhury, A., Paiva, A., Abreu, R., Storey, M. (eds.) Proceedings of the 46th IEEE/ACM International Conference on Software Engineering. pp. 1–13. ACM Digital Library (2024).
- [5] Phung, T., Cambronero, J., Gulwani, S., Kohn, T., Majumdar, R., Singla, A., Soares, G.: Generating high-precision feedback for programming syntax errors using large language models.
- [6] Schlude, A., Mendel, U., Stürz, R.A., Fischer, M.: Verbreitung und Akzeptanz generativer KI an Schulen und Hochschulen (2024), bidt, Munich.

## Appendix

### Summary of the Items of the Questionnaire

#### Item Block 1: Frequency of use and system selection

1.1 Have you heard of code generators for creating and/or checking programme code and have you ever used systems such as ChatGPT or Copilot for this purpose?

*Answer options:*

- *Yes, I have heard of it and use it regularly for these purposes.*
- *I have heard of it and use it sometimes for these purposes.*
- *Yes, I have heard of it and used it once or twice for these purposes.*
- *Yes, I have heard of it, but never used it for these purposes.*
- *No, I had never heard of it before this survey.*

1.2 Indicate to what extent you agree with the following statement about yourself: I know and understand the principles of generative AI.

*Answer options: 5 point Likert scale 'Strongly disagree' to 'Strongly agree'*

1.3 I use the following code generators for programming/correcting code or related tasks:

*Answer options:*

- *OpenAI GPT-3.5*
- *OpenAI GPT-4*
- *GitHub Copilot*
- *Amazon Codewhisperer*
- *Mistral AI*
- *Others*

1.4 What other code generators do you use?

*Answer in open text field*

## **Item Block 2: Tasks and Purposes**

2.1 When do you use code generators for programming tasks

*Answer options (multiple choice):*

- *At University*
- *private projects*
- *at work*
- *others*

2.2 With what intention do you use code generators when using them for programming tasks?

*Answer Options (5 point Likert scale for each statement from 'Never' to 'Always'):*

- *I use code generators to learn to programme better.*
- *I use code generators to submit my programming tasks in time.*
- *I use code generators to understand programming tasks properly.*

2.3 How often do you have code generators perform the following specific tasks?

*Answer Options (5 point Likert scale for each statement from 'Never' to 'Always'):*

- *Generate program code*
- *check program code*
- *translate program code into other language*
- *explain program code*

### **Item Block 3: Advantages**

*3.1 Code generators have helped me...*

*Answer Options (5 point Likert scale for each statement from 'Strongly disagree' to 'Strongly agree'):*

- *...to improve my performance.*
- *...to improve my programming skills.*
- *...to improve my knowledge about programming.*
- *...to improve my knowledge about programming concepts.*
- *...to save time when coding*
- *...to have more fun with programming.*

### **Item Block 4: Trust**

19 questions from the Trust in Automation Questionnaire, see Körber (2019)

### **Item Block 5: Technical Knowledge**

Read through the statements about code generators and indicate whether they are correct or incorrect.

*Answer Options (for each statement 'Correct', 'Incorrect' and 'I don't know'):*

- *'Code generators such as ChatGPT and the similar systems are large language models with transformer architecture.'*
- *'Code generators search the Internet for the right answers.'*
- *'The main function of the "self-attention" mechanism is the modelling of the time sequence of data points.'*
- *'The transformer architecture can be divided into encoder and decoder phases.'*
- *'Code generators can solve every task equally well.'*
- *'Code generators can provide incorrect answers.'*

## Item Block 6: Personal Motivation and Preferred Platforms

6.1 How would you rate the usefulness of the following platforms for solving programming tasks?

*Answer Options*(5 point Likert scale from 'Never useful' to 'Always useful' and 'I don't use it'):

- *Stackoverflow*
- *Reddit*
- *StackExchange*
- *GeeeksforGeeeks*
- *Youtube*

6.2 Read the following statements and indicate to what extent they apply to you personally.

*Answer Options*(5 point Likert scale from 'Strongly disagree' to 'Strongly agree'):

- *'I really want to understand the content of my degree program'*
- *'I want good grades', 'I enjoy programming'*
- *'Learning to programme is more difficult for me than for my fellow students.'*
- *'I believe that I can solve the programming tasks in my course independently.'*
- *'I regularly attend tutorials/practice sessions.'*
- *'Learning to programme is important to me.'*

6.3 What works particularly well when you use code generators for programming, and what doesn't? Is there anything else you would like to share? OR Why don't you use code generators? Is there anything else you would like to share?

*Answer Options: Open text field*

## Item Block 7: Demographics

7.1 Please enter your age in years.

7.2 Which gender do you identify with? (male/female/divers/no answer)

7.3 Which degree program are you enrolled in? Please indicate whether you are enrolled in a Bachelor's (BSc) or Master's (MSc) program.

7.4 What semester are you studying in?

7.5 Which university are you enrolled at?