

How encouraging computer science students to use generative AI for college coursework affects their programming skills and learning outcomes.

Brandon Evans

Valentina Waltman

Wootark Kim

Michael Sorensen

CST 462S: Race, Gender and Class in the Digital World

Bude Su

Brian Robertson

Brianna Sanchez Ortega

3/14/2026

Introduction/Research Problem

The rise of Artificial Intelligence tools like ChatGPT, Claude Code and Github's Copilot has significantly changed how computer science (CS) students approach coursework. These tools can generate code, debug programs and complete full assignments with minimal input. Universities that strive to prepare graduates for the modern world are increasingly allowing students to use generative AI for coursework. To better understand the implications of this shift in behavior we must understand how encouraging computer science students to use generative AI for college coursework affects their programming skills and learning outcomes.

Foundationally, education in CS depended on problem-solving, developing critical thinking skills, debugging, and learning specific syntax to enable learners to be successful. Generative AI has proven to be a useful tool to reduce a lot of these frictions, and it has served as a bridge to better understand concepts. On the other hand, skeptics postulate that these tools serve as shortcuts which undermine critical learning opportunities and devalue independent learning experiences. To get a glimpse of the truth, this research article will explore from the perspectives of four computer science students.

Literature Review

In the article, "Using Artificial Intelligence to Your Advantage in College: How ChatGPT Can Help", Jude Joseph, a college professor, discusses the benefits of utilizing artificial intelligence (AI) tools like ChatGPT in higher education. He focuses on how these tools can improve students' learning experiences by supporting different academic activities, including research, critical thinking, and time management. The article emphasizes several areas where AI tools can be beneficial for college students; ChatGPT intensifies college education by providing quick access to information that works as a supplement to their learning process and develops their critical thinking. Joseph states, "Students learn to ask thoughtful questions and evaluate the quality of answers by interacting with the system. This method promotes in-depth

learning and the development of analytical skills” (Joseph). It is debatable if this theory is factual, and if so, could it truly be called “analytical skills” if the opinions are received before the students had the chance to formulate one? How are the development of critical thinking and problem-solving skills to be measured if one receives a generated answer before the analytical side is able to be processed first? Nonetheless, the author focuses on the optimism of AI being a “virtual tutor by providing immediate comments, tips, and direction” (Joseph); however, this depends on responsible use. The article admits potential controversies associated with AI usage in education, especially plagiarism, dependence on technology, privacy accuracy, and ethical issues. Joseph does acknowledge later that “An excessive reliance on AI tools such as ChatGPT can result in a loss of critical thinking and problem-solving abilities.” Although the article promotes the use of AI as a beneficial tool for education, the concerns it raises about potential misuse and the risk of dependency could be seen as contradictory. In summary, the article argues that while AI can enhance education, without appropriate rigor to promote positive outcomes, the risk of losing comprehension and understanding is real. To test this possibility, the participants were separated into two groups, one group was free to use copilot and the other was instructed to write code manually.

The benefits of generative AI were obvious. The copilot group was able to complete more tests more quickly than the group that was writing code by hand. However, when they were tasked to answer open-ended questions about the code that was written, the group that wrote the code manually showed significantly better comprehension. This conclusion is evidence that there is a risk to letting gen-AI write code when the goal is comprehension and understanding.

In addition to comprehension debt, technical debt may also accumulate with AI usage and dependence. “Tracing Technical Debt by AI-Based Correlation Agent - A Case Study of Nokia Corporation” by Satu Grönroos, Tuula Hietanen, Anni Salmi, Katja Turtiainen shows that AI dependence increased code inconsistencies, unnecessary dependencies, or hidden security

flaws within a software. As such “debt” accumulates, interest is paid in the form of lengthy bug fixes. So, what should software developers do then? Forgo AI? No, instead, perhaps it is the mentality of the user, not the AI, that can offer the solution. The article implies that the next generation of software developers should master the “Human-in-the-Loop” principle, where they understand their role to be that of a supervisor and director of AI assistants. Mastering this role requires a shift in the mentality of “having to write (or push) code” to that of a critical thinking “validator.” Such a paradigm shift can allow computer science students to focus more on critical thinking and architectural understanding skills rather than on minute, nuanced programming language knowledge.

On February 3, 2026, Anthropic released a paper, “How AI Impacts Skill Formation” and their findings from a study on how AI assistance impacts the formation of coding skills. In their article housing the link to this research paper, Anthropic emphasized how the study was devised from the observation of Anthropic internal data, suggesting certain tasks are sped up by over eighty percent with AI. Naturally, researchers began to wonder if this increased productivity comes with trade-offs.

To find out, Anthropic conducted a randomized controlled trial of fifty-two software engineers who fulfilled certain requirements, such as having used Python at least once a week for more than a year. They also made sure these participants were somewhat familiar with AI coding assistance and unfamiliar with Trio, a Python library on which their tasks were based. The overall design between the control group and treatment group were similar, with the only difference being the treatment group having AI assistance in the Trio task. There is also a warm-up coding task, post-task quiz, and post-task survey, where AI is not allowed to be utilized by either group.

The results of this study concluded that participants in the AI group finished about two minutes faster. However, this came with a notable disparity between the two groups in regards to mastery. Despite the quiz being based on concepts that participants used in the Trio task,

average scores were observed with a 17% variance between the two groups. The treatment averaged a measly 50% as opposed to the control, who scored an average of 67% on the same quiz.

The research does emphasize the fact that merely utilizing AI did not guarantee lower scores, rather, it is how someone utilized AI. For instance, treatment group participants who did fare well on the quiz used AI not only for producing code, but to build comprehension as well. According to the research, examples of comprehension building include asking follow-up questions, requesting explanations, or posing conceptual questions while coding independently (all which serve as “appropriate” rigour).

Research Question

How the encouragement of AI-assisted programming may impact CS students and their perceptions about the profession?

Research Design - Interview Questions

Valentina's:

- How do you usually use generative AI tools like ChatGPT, Claude, or GitHub Copilot when working on programming assignments, and do you feel that this use helps you understand the code better or makes you depend on the tool too much?
- Can you describe a time when using generative AI on a coding assignment improved your learning, or a time when it made it harder for you to truly understand the programming concepts?

Brandon's:

- Do you feel the encouragement of AI-assisted programming leading to a positive outcome for computer science students, and thereby the next generation of software engineers?
- Can you describe a time you fixed a bug using AI but realized later you did not actually understand why the fix worked?

Michael's:

- How comfortable would you feel completing a coding challenge like leetcode without AI assistance?
- How confident do you feel in creating a robust scalable system with the help of AI?
- How confident do you feel in designing that same system without AI?

Wootark (Tom)'s:

- When was the last time you were “sober” from AI usage while learning or coding? (not having utilized any LLMs in an entire session)
- Do you pre-create a framework or parameters for an LLM to follow before it answers your questions? (e.g. “Give me a topic direction to pivot without giving me an answer” “in this file of code, can you point to the line that I should take a look at without telling me why this line is problematic?” etc.)

Research Design - Target research participant audience

Computer Science students of any age, across different races, and socioeconomic levels

Interviews can be done through a Google Form

Brandon-specific: I plan to interview a coworker in my software engineering team at work. I can do the interview in-person, but I will also offer the option of filling out our google form. The goal with interviewing her is to get a participant who has already begun practicing in the professional software engineering industry. (Female, Spanish immigrant on a STEM work visa)

Deadline for interviewing: April 6th

Recruitment methods: Linkedin, discord, text message.

Conduct Research**Michael's:****Who at your site/within your target audience answered your questions?**

- I interviewed Participant 1 who recently completed her PhD in planetary physics. She does a lot of data analysis and modeling work.

What characteristics/demographics make this person an ideal interviewee/person to speak on your topic (age, income level, experience working at the site, client of the site, etc.)?

- I believe Participant 1 is an optimal candidate because she was not traditionally trained in Computer Science but LLMs have lowered the barrier and allowed her to expand her skillset to complete her assignments more effectively. She has taken a non-traditional path to the discipline of Computer Science and now has a much better understanding of the work she does.

Valentina's

Who at your site/within your target audience answered your questions?

- Within our target audience I interviewed Participant 2, a senior Computer Science student at Cal State Monterey Bay.

What characteristics/demographics make this person an ideal interviewee/person to speak on your topic (age, income level, experience working at the site, client of the site, etc.)?

- Participant 2 was our TA for the Algorithms class. I believe he was the ideal person to interview because he is an independent student (age 24+ and self supporting status) and non traditional student pursuing a second degree; in his first degree he never used generative AI in an academic setting, but in his current CS degree and in his current job he has been exposed to it. Also, his experience as a TA for CST 370 and 438 have equipped him firsthand with a better understanding of the weaknesses and strengths that students experience when using AI for academic purposes.

Brandon's:

Who at your site/within your target audience answered your questions?

- Within our target audience I interviewed Participant 3, a recent graduate in Computer Science and coworker. We are both software engineers and collaborate daily.

What characteristics/demographics make this person an ideal interviewee/person to speak on your topic (age, income level, experience working at the site, client of the site, etc.)?

- I felt Participant 3 was a great candidate to interview since she is already a practicing software engineer (like myself) but also graduated recently as a computer science student, so she is still very much aware of AI's impact on the discipline. She is also from Spain on a F-1 visa with STEM OPT extension, which allows her to work in the United States for a STEM OPT participating employer for up to two years. This opens up additional perspectives for our research.

Wootark's (Tom):

Who at your site/within your target audience answered your questions?

- I interviewed Participant 4, a senior student at San Diego State University. He plans to enter the career field of cybersecurity.

What characteristics/demographics make this person an ideal interviewee/person to speak on your topic (age, income level, experience working at the site, client of the site, etc.)?

- I believe he can provide excellent insight into our topic as he is currently a computer science adjacent student who has to learn a balance in AI usage due to his field: cybersecurity. LLM can help him teach and debug code, yet as a cybersecurity student, he (ideally) won't be able to input and expose critical company code in LLM such as Claude (at least that is my guess). Therefore I am curious as to how LLM will affect his coursework and his preparation for a career.

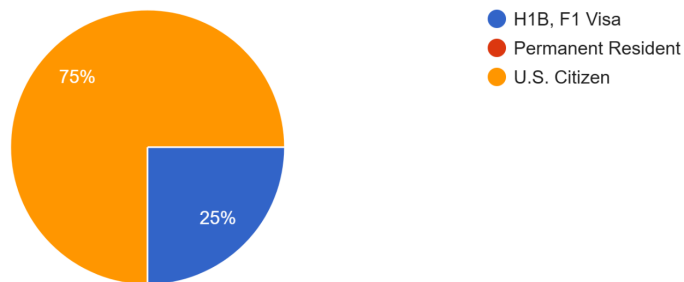
Findings

In order to prevent all the data coming from one isolated cluster, we made sure that the survey participants had a variety of backgrounds in order to create a diverse pool of responses. Demographically, the sample showed some variation. Of the four respondents, three were U.S.

citizens and one identified as holding an H1B/F1 visa. The sample was evenly split by gender, with two women and two men, and evenly split by financial aid status, with two students receiving aid and two not receiving aid. The charts below were used to help us understand the data coming from the immigration status, gender, and students receiving financial aid questions.

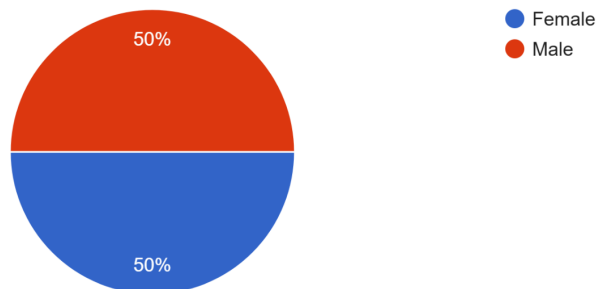
Immigration status

4 responses



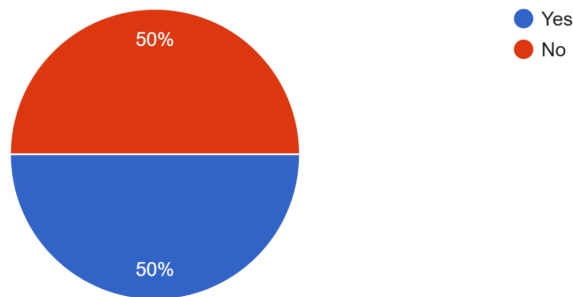
Gender

4 responses



Do you receive financial aid for college?

4 responses



All four participants reported using generative AI to support their programming coursework. Participants' actions show LLMs were primarily used to explain concepts, generate practice questions, and provide guidance rather than relying on AI to simply complete assignments. Across responses, AI was mainly described as a learning support tool rather than a substitute for independent thinking. For example, one participant compared AI to “GPS navigation,” while another described it as “a personal tutor” (Nguyen, 2026; Lozano Gonzalo, 2026).

Despite mostly positive affirmations of AI producing a conducive learning experience, participants' attitude toward AI, however, was that of caution. This can be seen from all participants' propensity to set some form of framework to prevent the AI from divulging the answer before giving the participant a chance to learn. Participants explained that they intentionally ask AI for hints, explanations, or logic rather than full solutions (A. Adams, personal communication, April 4, 2026; J. Kim, personal communication, April 6, 2026).

The responses thus clearly display the presence of tension between learning support and dependency. Although all four participants could describe ways AI improved their learning, they did not all believe that increased AI use would automatically lead to better long-term outcomes for computer science students. Two respondents expressed positive views, while the other two were more conditional in their support, emphasizing that AI can weaken a student's

ability to code independently if it is used too early or too often. This suggests that students see AI as beneficial when it accelerates understanding, but potentially harmful when it replaces the development of core programming fundamentals.

The confidence questions reinforced this pattern. All four respondents reported confidence in creating robust or scalable systems with AI assistance, and three of the four expressed at least moderate confidence doing so without AI assistance. However, confidence dropped noticeably when participants were asked about completing LeetCode-style coding challenges without AI. Only one respondent expressed clear comfort with that kind of independent problem solving, while the other three described themselves as nervous, not very comfortable, or only somewhat confident. This pattern suggests that AI may improve confidence in larger, project-based tasks while reducing confidence in more individual, interview-style coding tasks.

The responses about debugging also added important nuance. Two participants described situations in which AI produced a fix that appeared to work even though they did not fully understand why it worked, while the other two said they continue asking follow-up questions until the reasoning becomes clear. This finding shows that the impact of AI in critical thinking may depend not only on whether students use it, but on whether they actively question and verify the output. This idea is further supported by the fact that all four respondents reported using some kind of prompting framework or parameters to prevent AI from immediately giving away the answer.

Finally, the question about being “sober” from AI use showed that reliance on AI varied widely across the sample. One participant said they had recently completed a learning session without AI, another said it had been three months, another said it had been two years, and another said this mainly happened during exams. This variation suggests that AI dependence is not uniform even within a small group of students.

Conclusions

Using AI as a learning tool creates another crutch in the learning process. This is highlighted by the survey responses that show participants' confidence in their ability with and without AI assistance. With AI assistance they are confident in their abilities but, without AI assistance they are mostly unsure in their ability to write sufficient code. This is similar to reflecting upon one's own ability to multiply two five- digit numbers together without the crutch of a calculator.

However, every survey participant highlighted how they used AI to enhance their understanding of a problem. Each of the participants had their own framework, which used the AI tool to challenge them and their understanding of the broader picture. This is evidence that while syntax level knowledge may be deteriorating, higher level understanding may be accelerating.

Students' responses also suggest that AI-assisted programming is shaping how they view the computer science profession itself. Rather than seeing programming as only memorizing syntax or solving every problem alone, participants increasingly viewed success in the field as involving the ability to work with AI critically, ask strong questions, verify outputs, and still understand the logic behind a solution. At the same time, their lower confidence without AI indicates that they still associate professional competence with independent problem-solving ability. Because of this, the research question was answered to a meaningful but limited extent: encouraging AI-assisted programming appears to support learning efficiency and confidence, but it may also shift students' perception of the profession toward one where AI collaboration is expected, while raising concerns about whether core technical skills may weaken if that collaboration becomes overreliance.

Recommendations

Due to Artificial Intelligence's apparent impact at the syntax level it is recommended that aspiring computer scientists are challenged to begin their journey without the use of AI. To draw

on the mathematics analogy again; students must show the ability to perform basic functions before a calculator is allowed during exams. Unfortunately, the only way to enforce this would be to design special computers that prohibit the use of AI or return to in person, hand written exams. Once students show understanding of core concepts through sufficient repetition, then AI should be encouraged to re-inforce their understanding.

Recommendations for further research

The survey in our study was limited in a few ways that could be improved upon. The first and simplest of which is increasing the number of participants. Following that, a control group is recommended. All of our participants were already leveraging AI tooling in their learning and the opportunity for a with & without AI comparison was limited. For true understanding of this issue a base case is required and as AI becomes more embedded into our society that base case is increasingly more difficult to find as time goes on.

References

- Grönroos, S., Hietanen, T., Salmi, A., & Turtiainen, K. (2025). *Tracing technical debt by AI-based correlation agent: A case study of Nokia Corporation. Theseus*.
<https://www.theseus.fi/handle/10024/903761>
- Joseph, J. (2023, June 27). *Using Artificial Intelligence to Your Advantage in College: How ChatGPT Can Help*. LinkedIn.
<https://www.linkedin.com/pulse/using-artificial-intelligence-your-advantage-college-jude-o-seph/>
- Qiao, Y., Hundhausen, C., Haque, S., & Shihab, M. I. H. (2025). *Comprehension-performance gap in GenAI-assisted brownfield programming: A replication and extension*. arXiv.
<https://doi.org/10.48550/arXiv.2511.02922>
- Shen, J. H., & Tamkin, A. (2026, January 28). [2601.20245] *How AI Impacts Skill Formation*. arXiv. <https://arxiv.org/abs/2601.20245>

Appendix

Survey Response Data

Table A1

List of Survey Questions

Question 1	Email
Question 2	Name
Question 3	Age
Question 4	Nationality
Question 5	Immigration Status
Question 6	Gender
Question 7	Do you receive financial aid for college?
Question 8	How do you usually use generative AI tools like ChatGPT, Claude, or GitHub Copilot when working on programming assignments, and do you feel that this use helps you understand the code better or makes you depend on the tool too much?
Question 9	Can you describe a time when using generative AI on a coding assignment improved your learning, or a time when it made it harder for you to truly understand the programming concepts?
Question 10	Do you feel the encouragement of AI-assisted programming leading to a positive outcome for computer science students, and thereby the next generation of software engineers?
Question 11	Can you describe a time you fixed a bug using AI but realized later you did not actually understand why the fix worked?
Question 12	How comfortable would you feel completing a coding challenge like leetcode without AI assistance?
Question 13	How confident do you feel in creating a robust scalable system with AI assistance?
Question 14	How confident do you feel in creating a robust scalable system without AI assistance?
Question 15	When was the last time you were “sober” from AI usage while learning or coding? (not having utilized any LLMs in an entire session)
Question 16	Do you pre-create a framework or parameters for an LLM to follow before it

	answers your questions? (e.g. “Give me a topic direction to pivot without giving me an answer” “in this file of code, can you point to the line that I should take a look at without telling me why this line is problematic?” etc.)
--	--

Table A2*Answer for Question 1*

Participant 1	redacted
Participant 2	redacted
Participant 3	redacted
Participant 4	redacted

Table A3*Answer for Question 2*

Participant 1	redacted
Participant 2	redacted
Participant 3	redacted
Participant 4	redacted

Table A4*Answer for Question 3*

Participant 1	Participant did not wish to answer
Participant 2	40
Participant 3	21
Participant 4	22

Table A5*Answer for Question 4*

Participant 1	Participant did not wish to answer
Participant 2	Vietnamese
Participant 3	Spanish
Participant 4	U.S.A

Table A6*Answer for Question 5*

Participant 1	U.S. Citizen
Participant 2	U.S. Citizen
Participant 3	H1B, F1 Visa
Participant 4	U.S. Citizen

Table A7*Answer for Question 6*

Participant 1	Female
Participant 2	Male
Participant 3	Female
Participant 4	Male

Table A8*Answer for Question 7*

Participant 1	No
Participant 2	No
Participant 3	Yes
Participant 4	Yes

Table A9*Answer for Question 8*

Participant 1	I use it to complete homework assignments. I don't write coding assignments from scratch but, I use it to better understand concepts and create practice quizzes to help me learn better.
Participant 2	I use generative AI tools to research, plan, and even build most of the programming assignments unless it is prohibited. This helps me understand the code, but it also makes me very dependent on the tool. AI tools are like GPS navigation. I always have them on for guidance, but without them, I can still get to where I want to go, but it would take me longer to get there.
Participant 3	I use it a lot when I get sick or I need to catch up or understand something from class
Participant 4	To understand concepts and a deeper understanding of how the code works. If used correctly as a learning tool then AI usage helps accelerate learning.

Table A10*Answer for Question 9*

Participant 1	In general if I am trying to learn something AI will help me learn it but, it makes it too easy to just finish an assignment quickly and not focus on learning.
Participant 2	I used generative AI on my final project for computer graphics. Much of the math and advanced physics implementations were very difficult. Usually when I use generative AI, I don't feel that it makes it harder for me to understand as I can have the AI explain it to me, however for computer graphics I felt that there was too much I didn't understand. Just to note, AI usage was strongly recommended for the class.
Participant 3	Well, every time because it's like having a personal tutor. You obviously have to be willing to learn and not just get the answers but try and do it yourself and learn from mistakes
Participant 4	Understanding difficult to learn concepts or the logic reasoning as to why the code isn't working. Helped me understand basic fundamentals and instead of asking for the answer it gave me a logical approach as to why my code wasn't running as intended.

Table A11*Answer for Question 10*

Participant 1	I dont know what is expected in the workforce. Some jobs will expect you to use AI a lot while other jobs may want you to write your own code. In general it could be a detriment relative to previous generations who really had to learn. Future generations will have less ability to write code from scratch but, they will be able to learn concepts more quickly.
Participant 2	I actually feel that students should limit their use of AI. Though I believe that AI is necessary for today's software engineers, I believe it is also important to understand computer science principals on a deeper level without assistance. AI should be used after a level of expertise and competence has been reached.
Participant 3	I think so
Participant 4	Yes

Table A12*Answer for Question 11*

Participant 1	I was trying to do a project and I couldn't figure out why something wasn't working so I let AI take over my code. The AI gave me a solution that appeared to work because it was giving me what I wanted but, I discovered later that the AI did not implement what I wanted.
Participant 2	There are too many times for me to identify specifically. Not for school, but at work because of how big code bases can be and how much legacy code and tribal knowledge there is. There are parts of a codebase that almost noone completely understands in those types of projects.
Participant 3	This normally doesn't happen because if I don't understand it I'll just go back and forth with AI until I get it
Participant 4	No, simply keep asking why and how the fix changed the code and the logical reasoning to it. If you still don't understand just keep asking why at which specific part trips you up.

Table A13*Answer for Question 12*

Participant 1	I would be nervous
Participant 2	I feel comfortable because of how much practice I have already done without AI. I may be rusty, but I approached the learning at CSUMB with little assistance and focused on truly understanding the fundamentals. Being able to TA an algorithms class helped solidify my knowledge.

Participant 3	Not super comfortable anymore depending on how hard it is and if i have access to the internet or not
Participant 4	Somewhat confident.

Table A14*Answer for Question 13*

Participant 1	Very comfortable
Participant 2	I am very confident. I do this professionally and I understand that errors due to AI or errors due to human error are inevitable.
Participant 3	Super confident
Participant 4	Somewhat confident.

Table A15*Answer for Question 14*

Participant 1	less comfortable
Participant 2	I am confident in creating a robust scalable system without AI assistance, but I now that the task will take me 5 times longer.
Participant 3	Pretty confident
Participant 4	Somewhat confident

Table A16*Answer for Question 15*

Participant 1	During exams
Participant 2	Very recently. I don't use AI that much in learning unless I have to. At CSUMB I found that I did not need to rely much on AI at all. The coding assignments were very easy for an experienced developer.
Participant 3	Three months
Participant 4	Two years ago.

Table A17*Answer for Question 16*

Participant 1	I use LLMs to generate quizzes intentionally hiding the answers. So yes, I think.
Participant 2	For large tasks, I create a framework and parameters. For example, when using Claude Code or Cursor, you can create projects where you can provide instructions, templates, and extra context for a task. For Claude, this exists in the Claude.md file, and for other AI coding assistance tools, it's in some kind of markdown file that is parsed before the agent begins working.
Participant 3	Always
Participant 4	Yes, always set parameters to not have it directly spit out the answers only exception would be syntaxes. Instead of directly stating or fixing the issue have it explain/hint towards the issue and the reasoning as to how when the code is run why it messes up. Usually when having issues I ask why the code isn't running as intended and try to pinpoint the issue myself in the prompt by asking "could it be this or etc."