Two Sides of the Same Coin: Differing Approaches to Generative AI in Two Computer Science Classrooms

Sean Mackay
Department of Engineering Education
University at Buffalo
Buffalo, NY, USA
0000-0002-0358-7594

Kurt Eiselt
Department of Computer Science
University of California Davis
Davis, CA, USA
0009-0007-7206-6526

Adrienne Decker
Department of Engineering Education
University at Buffalo
Buffalo, NY, USA
0000-0002-0822-4813

***Abstract*—This innovative practice paper explores two differing approaches to the use of artificial intelligence tools in computer science classrooms. Artificial Intelligence (AI), while not a new technology, has seen a rise in popularity over the past two years, with companies such as OpenAI, Google, and Microsoft making readily available generative AI tools for anyone to use. This surge in AI popularity has led to a rise in its use in educational settings, in some cases allowed and in others discouraged or disallowed. A debate has risen among academics, particularly in higher education, about what AI's place in education is, with some educators actively encouraging its use while others view the use of AI as a form of academic dishonesty. Given AI is only advancing and is becoming more prevalent, it will only continue to become a more dominant force throughout education and the world at large. This paper presents two educators' distinct viewpoints and experiences on how AI should be handled in computer science courses (absolutely forbidden vs. decriminalized). The goal of this paper is to present different perspectives as well as concrete experiences we have had with AI in our own classrooms to encourage others to consider their own positions on its use and its implications for their own learning environments. While the debate on the place of AI in education is a long way from being settled, educators need to think about making choices, clearly articulating policies, and evaluating the positives and negatives of positions about AI in their classrooms.**

Keywords—academic integrity, LLM's, course policies, AI in education

I. Introduction

The release of ChatGPT has stirred up the centuries-old debate of when is technology a blessing and when is it a curse. In the educational domain, large language models (LLMs) like ChatGPT require us all to decide whether the technology provides our students with an opportunity to learn more and faster than ever before, or whether the technology represents the last nail in academic integrity's coffin. This conflict is perhaps more poignant for computer science educators as we have in some cases directly, and in more cases indirectly through our students, contributed to the development of this new technology. What do we tell our students about when and how, if ever, to employ LLMs in their education? This paper explores two very different answers to that question as implemented in multiple computer science courses by two instructors at two separate institutions.

II. Background

Academic Integrity has been an issue of much contention within the world of education, particularly within computer science. There has existed for several years a range of different tools designed to detect plagiarism within code, including programs such as Moss [1]. Historically, common violations of academic integrity came from students copying code from the internet or from their peers. Now, with the emergence of LLMs, we see a much larger set of possibilities for what academic dishonesty could be. On the other hand, there is a growing number of faculty who view LLMs as a potential tool to aide in the learning process as opposed to something that should be banned. This shift could even push educators away from the focus on syntax and instead time in classes can be spent developing higher level skills such as problem solving and critical thinking [2].

A. Reducing and Catching Academic Integrity Violations

There exists a difference between computing assignments and prose-based assignments commonly found in other fields that impacts how academic integrity could be handled, specifically in terms of how much students can rely on external resources outside of the course. In a writing course, for example, students can use grammar checkers as well as the option to review and cite resources to the extent they deem appropriate. In computer science, the question of how much students can rely on external resources and peers varies greatly depending on the course and context. Because of this, there exists a wide range of standards for what constitutes an academic integrity violation [3,4]. What has been continually argued, however, is the importance of providing students with clear guidelines for what constitutes academic integrity violations [5]. Concerns around academic integrity within computer science grew following the proliferation of the internet, with many authors posing it as a platform by which students would easily be able to obtain answers to programming assignments and share answers with their peers [6].

Several publications have proposed different methods of both discouraging academic integrity violations as well as catching them. Sheard et al. categorized existing strategies for handling academic integrity obtained from 30 academics across 25 different universities [7]. They found that solutions for attempting to reduce academic integrity violations fell into the following five groups: educating students about academic integrity, discouraging cheating through making it risky via detection or designing assignments where cheating is difficult, reducing the benefits of cheating, making cheating difficult, and empowering students [7].

In addition to reducing instances of academic integrity violations, detection of these violations has also been an area of research within computing. Tools have been developed for written assignments such as TurnItIn, which allows for the comparison of written prose [8]. For programming assignments this can be more difficult, particularly within lower-level courses such as CS1 or CS2 where programming assignments are relatively simple and there is little in the way of variation from one possible solution to another. Programs such as Moss do an excellent job of scanning more advanced pieces of code beyond a first-semester programming course against other student-submitted solutions to detect similarities in the semantic structure of their programs [1].

Other attempts at discouraging academic integrity violations have been proposed across all education fields. One of the more popular methods proposed has been honor codes [9, 10]. These pledges have been shown to have a mild improvement upon students' attitude towards academic integrity. Other research has focused on attempting to understand students' motivations for violating academic integrity. Motivations can range from students feeling as though they would do more poorly than they would like to in the course the academic integrity violation occurs in, to students not understanding that their actions run against the academic integrity policies present within their institution [5, 11, 12].

B. *Recent AI Developments*

The landscape of these academic integrity violations has rapidly shifted in the past two years. In late 2022, OpenAI published the first broadly publicly available large language model (LLM), ChatGPT. Within months, other companies followed suit including Microsoft and Google. These advancements in the potential power of LLMs stemmed from advances in computer technology from hardware manufacturers such as Nvidia and gave students for the first time the ability to use these technologies for little to no cost. Students' adoption of the radical technology was swift, with discussions of the implication of ChatGPT and its competitors becoming commonplace among academics in 2023 publications [13, 14, 15]. Conversations around how ChatGPT can be effectively handled have been conflicting, with some academics arguing against allowing its use in classrooms and others arguing that it should be acknowledged and integrated.

Lau and Guo discuss the tension that has occurred between those that are supportive of ChatGPT and other LLMs and those who actively dislike their use in classroom environments [15]. They argue that in the short term, there are immediate concerns about academic integrity that have led to some educators adopting strategies such as weighting exams more heavily and banning the use of LLMs outright. Others have approached it with the acknowledgement that these LLMs are not going away and have instead chosen to expose students to what these LLMs can provide as well as their inherent limitations. Lau and Guo argue that long term, educators have a choice to either embrace or shun the use of LLMs within their classrooms.

Lau and Guo stop short of claiming that they know the "correct" answer to the use of LLMs. They raise several valid points around the implications of both solutions, from the reality that students will have continued access to these tools, to the implications that failing to provide students with adequate experiences with how to use LLMs will impact their ability to utilize these tools in professional environments. There exists a concerning issue with the consistency of the policies educators adopt with regards to how students should or should not use LLMs. This contributes to the already difficult navigation of what is considered a violation of academic integrity because there are already some differences in policies in place for some courses, particularly computer science ones with regard to when an outside source is permitted and when it is not.

Other work has considered the problems as well as implications that LLMs have directly brought to educational environments [13]. Becker et al. claim that it is both important to acknowledge that LLMs have had a strikingly fast surge in general availability and popularity, and their presence has caught educators largely off guard [13]. They call for the importance of reviewing existing educational practices in light of these new developments in artificial intelligence [13]. We are also beginning to see additional calls for the adoption of LLMs as supports for learning in K-12 classrooms [16], in traditional CS1 (first-year, introductory programming) classrooms [17], and in upper-level software engineering classrooms [18, 19].

Hellas et al. has considered the accuracy of code responses provided by a range of different LLMs to typical first-year coding problems [14]. They note that while the LLMs are quite powerful at producing results for programming prompts they also produce a large volume of errors including oftentimes not getting code in a form that the automated grader can acknowledge. Simply allowing students to use these tools to create code either legally or illegally might not support their learning needs.

These issues, we argue, stem from inherent biases felt by educators and academics about what the purpose of education is, what purpose assessment serves, and how we should measure students' performance in our classrooms. We also argue that regardless of what strategies are employed, our existing educational frameworks are not prepared to handle the vast and long-standing implications of LLMs and their broad availability. Regardless of future advancements, these existing technologies are not going away; nor is their popularity waning. GitHub's Copilot has been continually gaining functionality and has seen widespread use and is now free for verified student accounts on GitHub [20]. Microsoft has continually been adding options for more LLM functionality into their operating system, with the latest fall update adding Copilot integrations throughout their operating system in Microsoft Office, the web browser, and in Visual Studio Code [21, 22]. It is unlikely for these trends to reverse course and thus educators will be forced to take up a strategy of handling these LLMs in their classroom on a permanent basis. We argue that it is important for educators to first consider what the true purpose of education serves prior to considering how they should handle the use of LLMs.

In this paper, we will discuss two different approaches of handling the use of LLMs, like ChatGPT, in multiple classroom environments. We will explain the two approaches, as well as what we have perceived has been effective from these differing strategies and what the next steps are for the next offerings of these courses.

III. The Experiences

This section presents information about the two different approaches to the possible use of LLMs by students. There are a total of three courses being discussed in this section taught by two different instructors at two different schools within the United States.

University at Buffalo is a state university in the northeast United States, classified by the Carnegie Classification[1] as "Doctoral Universities: High Research Activity." The student body of about 30,000 students is "primarily residential," and the university's main campus is in a suburban setting. The university's undergraduate profile is characterized as "Selective, Higher Transfer-In." The Computer Science major is located in a College of Engineering in a department of Computer Science & Engineering.

UC Davis is a state university in the western United States, classified by the Carnegie Classification as "Doctoral Universities: High Research Activity." The student body of about 30,000 students is "primarily residential," and the university is a public land-grant university in a suburban setting. The university's undergraduate profile is characterized as "More Selective, Higher Transfer-In." The Computer Science major is located in the College of Engineering in a department of Computer Science.

Both instructors taught in the summer term (not during the regular academic year, i.e., shortened course schedule) for their universities. Instructor1 (first author) has three years of teaching experience at one institution and Instructor2 (second author) has over 35 years of teaching experience across four institutions in the US and Canada. Instructor1 taught a CS1 course, while Instructor2 taught Introduction to Data Structures and a Programming Languages course. Instructor2 carried their policies into a regular academic term and those experiences are discussed in IV.

A. Course Structures

Instructor1. The CS1 course in the summer is taught over 10 weeks and is an online-only course. The course has no formal prerequisites and requires no prior programming experience. Students should have taken a pre-calculus prior to enrolling in the course (either in university or prior). The course has a mix of first-year students who intend to major and others who are non-majors. The course was taught online via Zoom with class times at midday Monday, Wednesday, and Friday. There were 60 students enrolled in the course. The course covered the basics of the Python programming language, data types, conditionals, loops, lists, dictionaries, file operations, and working with web APIs in Python. The course had one teaching assistant that helped with running the two lab sections for the course on Monday and Wednesday afternoons, which were also entirely online, as well as offering office hours.

The CS1 course had two primary types of assignments, labs, and homework. The labs occurred once a week and had a corresponding lab time where students could work online synchronously and get help from the TA. The students had all week to complete the lab from Monday morning till Friday night, though some weeks contained extensions to this assignment due to holidays or conflicts. The labs were designed to be more involved than the homework and required delivering a more substantial piece of code. The homework assignments were geared towards providing students opportunities to reinforce their knowledge of theoretical concepts, their ability to read others' code, and write smaller snippets of code outside of an IDE. The course also had one final exam held on the last day of class. The exam was fully remote and open-note. No monitoring of students during the exam took place. Students had two hours to complete the exam any time within a 24-hour window. Student attendance for online synchronous lab sections and the Friday lectures were also weighted as part of the grade.

Instructor2. The Introduction to Data Structures course in the summer is taught over six weeks and is an in-person only course. The prerequisite for this course is an introductory programming course in Python. The course is explicitly designed for non-majors, although students have the option of changing to the computer science major later. The course met three days a week (Tuesday, Wednesday, and Thursday) for 100 minutes each day. There is a discussion section scheduled for one additional hour each week. Attendance at the discussion section is encouraged, but not required. There were approximately 60 students enrolled in the course. The course covered algorithm analysis, abstract data types, classes, objects, recursion, lists, trees, dictionaries, graphs, searching and sorting. The course employed one teaching assistant who led discussion sections, held office hours, and graded homework and quizzes.

The Data Structures course had five assignments, one on algorithm analysis and four with small programming objectives. The class also had four quizzes, a midterm, and a final exam that contained multiple choice, short answer, small coding questions, and larger code comprehension questions. The quizzes and exams were given in class. Quizzes were closed book, while exams were closed book and limited open notes.

Instructor2 (Second course). The Programming Languages course was taught by Instructor2 in both a summer and fall (regular) term, both in-person only. The course requires students to have completed courses in introductory programming, object-oriented (OO) programming, data structures and algorithms, discrete math, and introduction to computer architecture which includes instruction set architecture (ISA), and assembly language programming. This course is an upper division required course for Computer Science majors. In summer, the course met three days a week (Tuesday, Wednesday, and Thursday) for 100 minutes each day for six weeks. In the regular academic term, the course met for two days a week (Tuesday and Thursday) for 80 minutes each for ten weeks. There is a discussion section scheduled for one additional hour each week. Attendance at the discussion section is encouraged, but not required. There were approximately 100 students enrolled in the summer section and 195 students enrolled in the fall term. The

[1] https://carnegieclassifications.acenet.edu/

course covers syntax, semantics, OO languages, functional programming, logic programming, and concurrent programming. The course employed teaching assistants (two in summer, three in fall) who led discussion sections, held office hours, and graded exams.

The Programming Languages course has five assignments, one on syntax and four others, one each on each of the four covered programming paradigms. The course also had two midterms and a final that contained multiple choice, short answer, small coding questions, and larger code comprehension questions. The exams were given during class and were closed-book exams with limited open notes.

B. LLM Policies

Each instructor handled the use of LLMs rather differently within their courses. Instructor1 took a stance of not allowing their use, while Instructor2 chose to "decriminalize" their use, only requiring students to cite what they used in completing the assignments.

Instructor1 had a clear policy that disallowed the use of LLMs as being akin to other forms of plagiarism or academic dishonesty. The specific policy wording contained within the syllabus is as follows: "The use of any generative artificial intelligence application in this course is considered an automatic violation of academic integrity and the student will receive a grade of F and a recommendation for expulsion from the program." The intended goal of this policy was to act as a deterrent from students trying to use LLMs within the course. The wording was initially borrowed from another syllabus in the department from years prior about different forms of academic dishonesty, and is the policy being held by most of the rest of the department currently. The policy was further explained several times during lectures including the first day. Instructor1 reiterated multiple times that they viewed the use of ChatGPT or other LLMs as being the same as copying an answer from another student and thus it was an academic integrity violation.

Instructor1 stated that they arrived on handling LLMs this way as it was the way most other faculty in their department were handling the issue. At the beginning of the summer, Instructor1 stated they did not realize how pervasive this problem was going to truly be, and that they had initially figured that a sternly worded policy as well as multiple discussions about the matter would go a long way to solving the problem. Instructor1 did at the time and still does view the use of LLMs as being detrimental to these students' learning of the concepts covered in the course. They stated concerns over ChatGPT being fully capable of providing functioning solutions to every problem they could realistically give to the students. The instructor stated they had also hoped that after the number of students who had been caught for the use of LLMs in the spring semester, students would be more receptive to seeing its use as not allowed within the department.

Instructor2 took a rather different stance on the use of LLMs. They essentially decriminalized what instructors tend to think of as cheating on homework assignments, including the use of LLMs and any other sources so long as students cited their sources and stated why they used them and how they helped. Some specific wording of the policy was: "You may use ChatGPT and other similar systems (e.g., Microsoft's Bard, Google's Copilot) without penalty when you are working on your homework assignments, so long as you give credit where credit is due (i.e., plagiarism is never acceptable) and provide me with helpful information about why you are using these systems and whether or not they helped. In other words, for any numbered problem on a homework assignment, if you used ChatGPT (or others) to arrive at a solution, you must include with the solution your answers to these questions:

- What source did you use? (failure to provide this information would be plagiarism)
- Why did you use that source? (explain what you were unable to do without the help)
- Did using that source benefit you? (if the answer is yes, explain how so...what did you learn?)
- Did using that source interfere with your understanding in any way? (if the answer is yes, explain how so...how did the source interfere with your understanding?)

Of course, the discussion above raises more questions: Can you also use Course Hero or Chegg? What about getting help from your classmates, or any other people? This may sound a bit crazy, but in the spirit of emphasizing learning and discovery, yes, those sources are allowable too BUT, once again, you have to answer the questions listed above, for each problem. Note however that while you may get help from another person, you are still responsible for completing your own work. We are not allowing groups of students (or others) to work on assignments as teams. This policy only applies to homework assignments, not in-class quizzes, or exams."

Instructor2 stated that they arrived on this policy based on their ongoing opinions regarding the viability of "policing" cheating on homework assignments. Instructor2 indicated that homework assignments have evolved into take-home exams in many courses and that assigning so much grade weight to work over which we have no control is simply unfair to the students who don't cheat, and results in an inaccurate perception of what has been learned by the students who do cheat. Making LLMs and other sources available for everyone while at the same time reducing the final grade contribution of homework assignments helps solve those problems. Homework assignments are now more about learning and discovery and less about earning the most points one can, as a result.

C. Course Results and Perceptions

The instructors had differing opinions of how their course went with regards to academic integrity and their policies. Instructor1 ended up catching 12 or 20% of their students for the use of LLMs or other academic integrity violations that could not be separated from LLMs being involved. A further 20 students were suspected of having used LLMs, but no action was taken due to a lack of confidence in what had actually occurred, meaning a further 33% of the class were suspected of having violated the policy. By contrast Instructor2 has failed zero students for academic integrity in any of their three courses.

Instructor1 has stated that while they do not feel that the way they approached the policy was particularly effective, they do not particularly feel as though they have an alternative.

Instructor1 stated they deeply dislike the idea of allowing the use of LLMs in their class as they firmly do not see the difference between it and copying from a friend. Instructor1 emphasized that they view this as students missing the learning objectives and the purpose of the course. Instructor1 has stated, however, that they are receptive to hearing other means of how LLMs could be handled in the course and has stated they feel that the current method of simply disallowing the use of LLMs has been ineffective.

Instructor1 received mostly positive feedback from the policy. Most of the students who provided feedback on the course were in favor of the academic integrity policy and voiced their own concerns about their peers using ChatGPT. Instructor1 noted that they have since heard from another student in the department who is frustrated enough at their peers using ChatGPT that they are contemplating changing their major. The only student to have a negative opinion of the policy and voiced it to the instructor was a student who complained after having been caught violating the policy.

Instructor2 has not noticed changes in grades in their Data Structures course but did notice a change in their summer Programming Languages course, with grades trending higher than final grades in prior offerings of the course Instructor2 had in-class feedback sessions for the summer offerings of both the Data Structures and Programming Languages class.

During verbal in-class questioning during class, the relatively few students who were willing to talk were very positive about the policy and seemed to be using LLMs as a sort of always available teaching assistant. However, many students were silent during the in-class sessions. In the academic term Programming Languages class, written surveys were collected and information from those surveys are in III-E.

D. *Future Changes to Courses*

Both instructors have stated that they have plans for modifications both to course content as well as their policies going forward.

Instructor1 felt as though they had not succeeded in their existing policy and change was needed for it to be more effective at discouraging academic integrity violations. They stated they were uncertain about what they want to change going forward. They state that it is clear to them that what they did was not effective, although they are confident in their ability to catch cases where LLMs were in use. However, they do not feel as though the policy has worked as a proper deterrent in the way they had hoped it would. While Instructor1 states they do not want LLMs used in their class, they feel as though they need to go about handling their use in a different way. They state "I think for lower-level courses, LLMs only serve as a tool akin to Chegg in that it can simply solve their homework for them, rather than provide them an opportunity to learn the concepts I am trying to teach them. Given there are certain ethical expectations on my end in terms of ensuring the students meet the learning requirements outlined in the course description, I feel that the use of LLMs still needs to be at the least heavily discouraged or framed in a way that is still productive while ensuring they are still learning the course content fully. While this might seem like a lazy answer, students' motivations in courses particularly in their first year are so aligned to that of just earning the grade that they do not see the value in putting in the work to use LLMs in a meaningful or productive way. I envision this as a fundamental issue present within our modern education system and while I would deeply love to change it and am disturbed by its implications, I am not confident that I have any practical means of changing their view of grades in a 10-week course. Despite this, I do try to show students the power of what they are learning and try to ensure they are engaging with the material in a productive manner. Despite this, the temptation of just letting an LLM do their work effectively for them is something I am still deeply concerned about. While I am excited to see other solutions to this problem, I am not yet convinced we have a practical method of handling LLMs in the classroom."

Instructor2 took a more positive attitude with regards to LLMs and the direction they wished to take them in their course. Instructor 2 indicates that they would like to be able to give their students ideas on how best to use LLMs to their advantage. At this point all they have done is say that the students can use them, but the rest is up to the students. Adding help and guidance for the students on more effective use of the LLMs would be a positive addition to the course and policy. Instructor2 also wants to re-think the assessment of the course, leaning toward more quizzes/exams, each with less grade weight. However, balancing the increase in in-class assessment time versus loss of in-class lecture time is a challenge. One solution may be putting some of the lecture time online, but the university has limits in place for how much lecture time can be put online for courses that are in-person. In addition Instructor2 felt that if you trust that your students actually want to learn something, then LLMs help, but some of your students will probably still disappoint you. If you don't trust that your students want to learn something, well maybe you're in the wrong line of work?

IV. Academic Term Feedback

Instructor2 carried their policies into the academic term in Fall 2023 in the Programing Languages class. Instructor1 did not teach in the academic term in 2023-2024, so was not able to continue their policies to date. For Instructor2, no students were caught or prosecuted for academic integrity violations, but grades for the course skewed high again, likely because the homework grades skewed high due to the relaxed academic integrity policy. Instructor2 gave written surveys in Programming Languages in the fall within the normal end of term evaluations.

Of the approximately 195 students enrolled in the class, 58 students completed the survey. The students were asked if they used any of the approved outside sources to help with any of the homework problems assigned in class (e.g. ChatGPT, Stack Overflow, etc.). The responses indicated that 5 students did not use any outside sources on their homework and 53 students did. Table 1 summarizes the responses from those students as to why they did not use any of the approved outside sources. These students seemed to be incredibly self-reliant in their comments about their behaviors, indicating that they were able to find the help they needed from lecture slides, textbooks, online sources, and teaching assistants.

The students were asked if they felt the policy made them use outside sources on homework more, less, or about the same as they would have otherwise. Of the responses, 12 students indicated that they used the outside sources more than they would have otherwise, and 35 said they used them about the same as they would have otherwise. Of those 35, 29 confirmed that they used "ChatGPT, Bard or similar" as one of those outside sources. Thus, regardless of the policy, 60% of the students who indicated about the same level of use were likely going to use ChatGPT or similar anyway.

The survey asked students who used outside sources (53 of the 58 respondents) three additional questions:

1. Did you like the approach this class took in allowing sources of outside help? Why or why not?
2. Overall, do you think using these outside sources helped you in learning the course content? How so?
3. Are there ways in which using the outside sources interfered or otherwise negatively impacted your learning the course content? How so?

For Q1 (did you like the approach), 49 of the responses indicated yes, 1 indicated no, 1 indicated ambivalence, and 2 did not answer, which overwhelmingly indicates that the students were comfortable with the policies. For Q2 (did it help you learn the course content), 50 indicated that the sources did help them learn and 3 indicated that the sources did not help or didn't help much.

The comments from the students for each of these questions illuminated how they connected with the outside sources and their learning. Table II provides utterances from students with regard to Q1 (did you like the approach) and Table III provides utterances from the students with regard to Q2 (did it help you learn). While no formal coding was done on these responses, the overwhelming message within them seems to indicate that students used the sources to support their learning, that people will use them irrespective of the policy, and that they felt it was more authentic to what doing work in the real world would be like.

There were not many comments for Q3 (negative impacts), but those that answered the questions mainly discussed the fact that ChatGPT and other LLM sources are not always correct, which is supported by Hellas et al [14]. It is clear from the responses that the students were at least some of the time able to recognize this and/or knew it to be true so that their interactions with the LLMs or other sources were not just viewed as interactions with an all-knowing and perfect knowledge oracle. Table IV provides comments from this question where students expressed other ideas about the negative impacts of these outside sources on their learning.

V. Discussion

These two instructors took very different approaches to the problem of the use of LLMs in their classrooms. If you remove the one course taught during the regular academic term, the courses are remarkably similar in surface characteristics. They are summer courses that are on a compressed timescale, meet multiple times a week for a somewhat long duration and have very similar enrollments. The CS1 course and Data Structures course even have somewhat similar student characteristics in that the both courses has both major and non-major students. The large difference is the approach to LLMs and their use by students in the course.

In both courses, students engaged with LLMs. In the CS1 course, they were caught (often) and failed the course for it. In the other courses, provided they indicated they used LLMs, they received no punitive measures. Instructor1 spent a lot of time dealing with the finding and prosecution of academic integrity violations. Instructor2 did not spend nearly the same amount of time. In the surveys that students filled out for Instructor2, it is noteworthy that many of the students who felt that the policy did not impact their use of outside sources indicated that they used LLMs. So, it is likely irrespective of the policy, that they would have used the tools. In fact, the percentage of students indicating that they would use LLMs to Instructor2 nearly matched the numbers of students found or suspected of using them in Instructor1's course. Both instructors realize that LLMs are not going away and that their courses must adapt and change to this new reality. It is only in their approach to that change that they differ.

There are unanswered questions in both instructor's minds about how to best integrate LLMs into learning. Instructor1 seems convinced that LLMs will always be used as a "free-pass" for students to not do their work, while Instructor2 seems to believe that allowing for a tool like an LLM to be a helper, students may focus more on learning than on getting the answers correct. It is likely the case that they are both right depending on the particular student. In fact, the responses from students to the end-of-course survey in Fall 2023 shows this to be the case. By in large, the students saw these tools as a benefit, but recognized that there will always be some students who will use any means necessary to gain an advantage. It seems incumbent on the education system around the students to help better support the learning and not the answer-getting. It is likely the case that course structures and grading and incentive structures need to change to better align with the goals of focusing students on the learning as opposed to the quest for correct answers. For example, considering approaches that shift the focus of grading towards assessments of learning as opposed to the acquisition of points [23, 24, 25, 26, 27].

Reading through the student surveys from the Fall 2023 offering from Instructor2 gives some further insight into how students are approaching these issues and some of the ways in which they are using these outside tools. The students recognize that students will be using the tools regardless of the policy, so they appreciated the decriminalization of their use. At least one student noted that due to outside obligations, office hours are difficult to attend and having other resources available for use without penalty was helpful in their learning (Table II). By in large, the students recognized the fact that the tool usage did not replace their learning, but rather augmented it. The LLMs and other tools did not always provide correct answers and even when they did, several students noted that it was still part of their job as a student to recognize it as correct and to understand it. Simply copying it could have negative consequences as referenced by the student comment from Table IV that indicated they felt they didn't learn as deeply and this was reflected on their exams.

TABLE I. COMMENTS FROM STUDENTS WHO DID NOT USE OUTSIDE SOURCES

Please take a moment to tell us briefly why you did not use those approved outside sources
Pride? I also don't really trust it for much. Most of my experience with it is asking it to do smalysh (sic) proofs to test it, and it did poorly.
Most of the problems were simple enough to be complete without it.
I didn't use these sources because I can find what I need in the lecture slides and the website for each language.
Understanding code from the lectures and discussions made it easier to go about the homework without ChatGPT. ChatGPT was a tutor in helping me better understand lecture and discussion material.
It forced me to seek out help from TA's and forced me to work on understanding code on my own which helps me do better in the class.

TABLE II. SELECT COMMENTS FROM SURVEY Q1

Did you like the approach this class took in allowing sources of outside help? Why or why not?
Definitely. Outside sources are always treated as taboo. But I learned a lot from them, and it took off a lot of stress so the material was more enjoyable.
Yes, because we can utilize different tools at our benefit to learn and apply later in our jobs.
Yes because it allowed me to not feel like I was cheating when using resources. It helped me use them to learn the problems and concepts.
I prefer this class allowing using outside help. Because I don't think a normal person will figure everything out all by themselves.
I do. I never used the outside help to generate code but it really helped when learning a new topic or relearning an old topic. Especially with Haskell and Prolog.
Yes, because people will do it either way, and it isn't like you can just put a prompt in and get a correct answer. You still need to know how to get the answer you want. It's just a tool.
Yes, I think it made me worry less about what is conventionally considered cheating and focus more on solving the problem at hand.
Yes. I personally liked it as I'm unable to show up to office hours and tutoring due to my family and scheduling.
I did. I think I felt like I could actually understand the content thanks to the outside sources. I think it promotes easier and more understandable learning.
Yes, I need to test my understanding on lots of examples and edge cases to ensure sufficiently robust comprehension. LLMs allow me to quickly loop through testing my assumptions, adjusting my understanding, and testing it again on as many examples as I need.
Yes, because it is normal to use outside sources like this in the real world.

TABLE III. SELECT COMMENTS FROM SURVEY Q2

Overall, do you think using these outside sources helped you in learning the course content? How so?
Yes, it depends how you use it and what you should expect from it. Use it as a tool for guidance and explanations. The answers it gives may not be correct, but its thought process usually is.
Yes. I would ask ChatGPT to create notesheets of the syntax for each language and it could explain topics in ways I could understand
Yes, I was able to get a much better conceptual understanding.
Yes, it definitely did. Most of the time I don't understand the sample problems from discussion, so usually I go home and ask AI to explain them to me in plain English.
Yes, they explain the materials in other terms that help me understand better.
Yes, as I primarily used outside sources to either explain topics to me or to debug incorrect syntax.
I am of the opinion that it is the responsibility of the educator/institution to provides students with as many resources as possible. It is then up to the students to use them as they wish. For students who really want to learn, LLMs can be an invaluable tool. For those who want to have their homework done for them (for any reason), that should be accepted as okay

TABLE IV. SELECT COMMENTS FROM SURVEY Q3

Are there ways in which using the outside sources interfered or otherwise negatively impacted your learning the course content? How so?
Yes. Ideally, it didn't allow me to learn the concepts as deeply and this reflected on my exams.
Using outside sources too much make me rely on them more instead of thinking by myself.
I think it is easy to become reliant on these tools. When using these tools, it is extremely important to make sure you still understand what you are writing and not blindly copy.
Yes, if I was lazy I could use GPT to solve without learning.
Yes. I think it doesn't create the best problem solving skills for programmers because everything is so easy to come to you.
I think it made me lazy when it came to debugging specifically because if I had a small error, I would put it in ChatGPT to check if anything is wrong, and it was usually a syntax related error

VI. CONCLUSION

The purpose of this paper was not to provide answers to the question of what instructors should do, but rather to give two radically differing approaches to the use of LLMs in computer science classrooms and discussion about the experience each instructor had with implementing these approaches in their course. Both higher education and computer science education are at an inflection point where we need to think critically about how we support student learning, what learning means, what assessment means, and what role these tools will play in education going forward. We need to learn about the tools, their affordances and drawbacks and try out policies and procedures in classrooms to best figure out how these new technologies will help or hurt learning. It is likely not time to assume that all students are simply trying their best to cheat as opposed to trying to learn. Moving away from the punitive model which views students as enemies or adversaries is long overdue.

It is also a time when we need to reflect on what we value about learning, knowledge, and demonstrations of learning. Responding to simple programming prompts with a few lines of code (often the fewest possible) is likely not going to be a skill that students will need going forward and likely isn't a skill that anyone will value. If that is the case, going forward, what are the skills we need to value, teach, support, and assess?

REFERENCES

[1] A. Aiken, 'Plagiarism Detection', theory.stanford.edu. Accessed: Dec. 11, 2023. [Online]. Available: https://theory.stanford.edu/~aiken/moss/

[2] R. D. Caballar, "AI Copilots Are Changing How Coding Is Taught Professors are shifting away from syntax and emphasizing higher-level skills", IEEE Spectrum [Online], May 9 2024. Available: https://spectrum.ieee.org/ai-coding

[3] Simon, B. Cook, J. Sheard, A. Carbone, and C. Johnson, 'Academic integrity: differences between computing assessments and essays', in Proceedings of the 13th Koli Calling International Conference on Computing Education Research, in Koli Calling '13. New York, NY, USA: Association for Computing Machinery, Nov. 2013, pp. 23–32. Doi: 10.1145/2526968.2526971.

[4] Simon, B. Cook, J. Sheard, A. Carbone, and C. Johnson, 'Academic integrity perceptions regarding computing assessments and essays', in Proceedings of the tenth annual conference on International computing education research, in ICER '14. New York, NY, USA: Association for Computing Machinery, Jul. 2014, pp. 107–114. Doi: 10.1145/2632320.2632342.

[5] Simon, J. Sheard, M. Morgan, A. Petersen, A. Settle, and J. Sinclair, 'Informing students about academic integrity in programming', in Proceedings of the 20th Australasian Computing Education Conference, in ACE '18. New York, NY, USA: Association for Computing Machinery, Jan. 2018, pp. 113–122. Doi: 10.1145/3160489.3160502.

[6] L. M. Hinman, 'Academic integrity and the World Wide Web', SIGCAS Comput. Soc., vol. 32, no. 1, pp. 33–42, Mar. 2002, doi: 10.1145/511134.511139.

[7] J. Sheard, Simon, M. Butler, K. Falkner, M. Morgan, and A. Weerasinghe, 'Strategies for Maintaining Academic Integrity in First-Year Computing Courses', in Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, in ITiCSE '17. New York, NY, USA: Association for Computing Machinery, Jun. 2017, pp. 244–249. Doi: 10.1145/3059009.3059064.

[8] A. Hapsari, M. I. Ghali, and M. H. Ammar, 'The Use of Turnitin to Teach Academic Integrity in Essay Writing Coursework', in Proceedings of the 2020 11th International Conference on E-Education, E-Business, E-Management, and E-Learning, in IC4E '20. New York, NY, USA: Association for Computing Machinery, May 2020, pp. 169–173. Doi: 10.1145/3377571.3377595.

[9] D. L. McCabe and L. K. Trevino, 'Academic Dishonesty: Honor Codes and Other Contextual Influences', The Journal of Higher Education, vol. 64, no. 5, pp. 522–538, 1993, doi: 10.2307/2959991.

[10] D. L. McCabe, L. K. Trevino, and K. D. Butterfield, 'Academic Integrity in Honor Code and Non-Honor Code Environments: A Qualitative Investigation', The Journal of Higher Education, vol. 70, no. 2, pp. 211–234, 1999, doi: 10.2307/2649128.

[11] E. Löfström, T. Trotman, M. Furnari, and K. Shephard, 'Who teaches academic integrity and how do they teach it?', High Educ, vol. 69, no. 3, pp. 435–448, Mar. 2015, doi: 10.1007/s10734-014-9784-3.

[12] P. Newton, 'Academic integrity: a quantitative study of confidence and understanding in students at the start of their higher education', Assessment & Evaluation in Higher Education, vol. 41, no. 3, pp. 482–497, Apr. 2016, doi: 10.1080/02602938.2015.1024199.

[13] B. A. Becker, P. Denny, J. Finnie-Ansley, A. Luxton-Reilly, J. Prather, and E. A. Santos, 'Programming Is Hard – Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation', in Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1, in SIGCSE 2023. New York, NY, USA: Association for Computing Machinery, Mar. 2023, pp. 500–506. Doi: 10.1145/3545945.3569759.

[14] A. Hellas, J. Leinonen, S. Sarsa, C. Koutcheme, L. Kujanpää, and J. Sorva, 'Exploring the Responses of Large Language Models to Beginner Programmers' Help Requests', in Proceedings of the 2023 ACM Conference on International Computing Education Research – Volume 1, in ICER '23, vol. 1. New York, NY, USA: Association for Computing Machinery, Sep. 2023, pp. 93–105. Doi: 10.1145/3568813.3600139.

[15] S. Lau and P. Guo, 'From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot', in Proceedings of the 2023 ACM Conference on International Computing Education Research – Volume 1, in ICER '23, vol. 1. New York, NY, USA: Association for Computing Machinery, Sep. 2023, pp. 106–121. Doi: 10.1145/3568813.3600138.

[16] M. Kazemitabaar, X. Hou, A. Henley, B. J. Ericson, D. Weintrop, and T. Grossman. 2024. How Novices Use LLM-based Code Generators to Solve CS1 Coding Tasks in a Self-Paced Learning Environment. In Proceedings of the 23rd Koli Calling International Conference on Computing Education Research (Koli Calling '23). Association for Computing Machinery, New York, NY, USA, Article 3, 1–12. https://doi.org/10.1145/3631802.3631806

[17] M. Liffiton, B. E. Sheese, J. Savelka, and P. Denny. 2024. CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes. In Proceedings of the 23rd Koli Calling International Conference on Computing Education Research (Koli Calling '23). Association for Computing Machinery, New York, NY, USA, Article 8, 1–11. https://doi.org/10.1145/3631802.3631830

[18] B. Arie Tanay, L. Arinze, S. S. Joshi, K. A. Davis, and J. C. Davis. An Exploratory Study on Upper-Level Computing Students' Use of Large Language Models as Tools in a Semester-Long Project. Accepted to ASEE General Conference of the American Society for Engineering Education (ASEE). Available: https://arxiv.org/abs/2403.18679

[19] V. D. Kirova, C. S. Ku, J. R. Laracy, and T. J. Marlowe. 2024. Software Engineering Education Must Adapt and Evolve for an LLM Environment. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA, 666–672. https://doi.org/10.1145/3626252.3630927

[20] T. Dohmke, 'GitHub Copilot is generally available to all developers', The GitHub Blog. Accessed: Dec. 11, 2023. [Online]. Available: https://github.blog/2022-06-21-github-copilot-is-generally-available-to-all-developers/

[21] samanro, 'Microsoft Copilot documentation and resources'. Accessed: Apr. 09, 2024. [Online]. Available: https://learn.microsoft.com/en-us/copilot/

[22] 'GitHub Copilot · Your AI pair programmer', GitHub. Accessed: Apr. 09, 2024. [Online]. Available: https://github.com/features/copilot

[23] A. Decker, S. H. Edwards, B. M. McSkimming, B. Edmison, A. Rorrer, and M. A. Pérez Quiñones. 2024. Transforming Grading Practices in the Computing Education Community. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA, 276–282. https://doi.org/10.1145/3626252.3630953

[24] R. Weber. 2023. Using Alternative Grading in a Non-Major Algorithms Course. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023). Association for

Computing Machinery, New York, NY, USA, 638–644. https://doi.org/10.1145/3545945.3569765

[25] D. Garcia, M. Camarena, K. Lin, and J. Westerland. 2023. Equitable Grading Best Practices. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2023). Association for Computing Machinery, New York, NY, USA, 1200–1201.

[26] C. Ott, B. McCane, and N. Meek. 2021. Mastery Learning in CS1 - An Invitation to Procrastinate?: Reflecting on Six Years of Mastery Learning. In Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE '21). Association for Computing Machinery, New York, NY, USA, 18–24. https://doi.org/10.1145/3430665.3456321

[27] J. Garner, P. Denny, and A. Luxton-Reilly. 2019. Mastery Learning in Computer Science Education. In Proceedings of the Twenty-First Australasian Computing Education Conference (ACE '19). Association for Computing Machinery, New York, NY, USA, 37–46. https://doi.org/10.1145/3286960.3286965