Work in Progress: Integration of AI Tools on an Open-ended Computer Programming Project

Emily Hammond
Engineering Education
University at Buffalo
Buffalo, NY USA
eh32@buffalo.edu

Courtney Faber
Engineering Education
University at Buffalo
Buffalo, NY USA
cfaber@buffalo.edu

***Abstract*— This work-in-progress, innovative practice paper describes the design and implementation of a computer programming project that encouraged students to use artificial intelligence (AI) tools (e.g., ChatGPT). This innovative practice was implemented at a state university in an introduction to programming course. Course assessments included weekly homework assignments, an open-ended project, one quiz, and three midterm exams. Students were explicitly told in class and in the syllabus that no AI was allowed for weekly homework assignments; however, students were encouraged to use AI for the open-ended project. This decision was made because the problems on weekly assignments were more structured and could be easily solved by AI tools. In comparison, the learning objective of the project was focused on code design with open-ended requirements making it harder for AI to provide workable code. Students who reported using AI indicated that they used it to help start the project and/or a piece of the project and to aid in debugging or finding errors. Project grades were similar to grades seen in previous years for similar projects; however, grades on midterm two, which occurred after the project, were surprisingly higher than the grades for midterm one and higher than exam grades from previous years on similar topics. At this point, we do not know if/how the project contributed to these improved exam scores. Given the success of the project in its first implementation, Dr. E plans to use the assignment in future semesters. Additionally, Dr. C plans to collect data to begin exploring how the use of AI tools in this open-ended project supported students' learning and understanding of principles in the course.**

Keywords—undergraduate, learning technology, computational thinking

I. Introduction

In recent years, there have been numerous technological innovations within artificial intelligence (AI) and natural language processing (NLP). Out of the thousands of AI tools, currently available, OpenAI's ChatGPT (Chat Generative Pre-Trained Transformer) has been the most widely discussed [1]. It has set the record as the fastest-growing consumer application in history [2]. ChatGPT is a large language model that uses NLP to generate text-based responses to user prompts. This text includes producing computer code, essays, outlines, and solutions to math problems. Between its first release in 2018 and its third release in 2020, the size of its training parameters increased from 117 million to 175 billion [3]. This increase in training parameters has allowed ChatGPT to identify more complex patterns in language and provide more sophisticated responses [4].

Given the nature of ChatGPT, the relevance of the output is dependent on the prompt the user inputs. The term prompt engineering is being used to describe the process of modifying inputs to get outputs with specific qualities [5]. While ChatGPT is a powerful tool, it has its limitations and there are specific skills associated with using the tool to its full potential.

Within education, instructors have to make decisions about how and if AI tools, like ChatGPT, can be used in their courses. There have been concerns around academic integrity, especially given ChatGPT's ability to pass assessments such as the medical licensing exam [6] and law exams [7]. Within engineering education, Nikolic et al. [4] conducted a study where they tested ChatGPT on various types of assessments (e.g., online quizzes, numerical assignments, code submissions, oral, and visual) in subjects closely connected to engineering. They report the integrity weaknesses, integrity strengths, and opportunities related to ChatGPT across these various assessment types. Specifically relevant to this WIP are their findings related to code submission. They identified that ChatGPT is generally good at writing entry-level code and answering coding questions, but was not able to do more complex activities. They suggest that ChatGPT can be used as a tool to find and fix software bugs, write basic code, and explain the basics of coding [4].

Given the newness of this technology, there are limited examples of how to incorporate ChatGPT into specific types of classes. There is also a need to describe specific challenges and benefits for instructors and students based on actual classroom implementation. We aim to contribute to this knowledge by describing the decisions Dr. E made to incorporate AI tools into a course project, her perceptions of the benefits to students, and the changes she plans to make in future semesters to further support student learning.

II. Course Background

Dr. C.: Describe the course (learning goals, course structure, types of assignments, number of students, level of students).

Dr. E.: The course is an introduction to programming course. It's a 200 level, and it is taken primarily by second-year electrical engineering students. It is assumed that they do not have any programming experience. The electrical engineering students come from a digital logic course, so they have some introduction to basic concepts, but not programming.

The course is structured with the C programming language, for the first 9 to 10 weeks, and then the last three weeks of the class is C++. They have weekly homework assignments. These homework assignments are administered through an online textbook, where students do some reading and complete some interactive activities. They also have programming assignments which are longer homework activities, and they get 3 to 4 of these assignments per week. There are two projects in the course. First, there is a C project where they are expected to write a large program. This semester, I had them create a text-based game. Then for the second project, they had to re-design their game using object-oriented principles.

The general learning outcomes are to introduce students to the basic concepts of programming. They are introduced to variables, math operations, branching, looping, functions, and arrays. In C, some of the more complex topics are pointers, structs, and file input and output. In C++, they are introduced to basic classes, objects, and inheritance. By the end of the class, I hope students will be able to use all these concepts to do functional programming and be introduced to object-oriented programming.

I teach the course in the fall semester, and there are typically 25-35 students in the course. I taught the class for the first time in the fall of 2018 and I have taught it every fall since then.

Dr. E: What was your general AI policy for the class?

A: In my syllabus, I had an AI policy that students were not allowed to use AI, except for the projects. For the weekly assignments, I wanted students to practice and struggle with the weekly concepts. These assignments were also smaller and less complex, so ChatGPT would have been able to give them most of the answers. I allowed students to use AI for the projects because the projects were much more open-ended. I knew that AI would not be able to give them all the code that they wanted and that they would have had to do a significant amount of work which would have been more difficult than writing the code from scratch. I was also more concerned about students designing the general program versus implementing a really small piece of code that does one small thing. So, if they used AI to write small pieces of the project, they would still have to put it all together and understand how it flows together.

Q: Why did you choose this policy?

A: I went that route because I was curious to see what students would do and how they would use it. I also talked to one of the professors who teaches in the spring. He allowed students to use AI in the spring and had them document all their interactions. I thought it was a good idea because of the capabilities of AI and the fact that AI could be used as a tool, but not able to directly complete the whole project. I was just curious to see how they would use it and what it would look like for students to use it.

III. Innovative Practice

Dr. C: Describe the open-ended project that you assigned.

Dr. E: I'm going to focus this question primarily on the first project. I assigned the project in week 6 or 7 of the course. We had covered a lot of the fundamental topics and were still covering some of the more challenging topics in C. I gave students three or four weeks to complete the project. Students were presented with a one-page project brief. Their task was to implement a text-based adventure game. Text-based being that they needed to use basic printing to the command window. They were required to have a map that their player would navigate through. There was a minimum map size so that it provided a sufficient challenge for students. They had to implement movement for their player. Typical gaming is WASD, which most students implemented where you have up, down, left, and right. Then they needed to have things for the player to interact with, such as landscape features or collectible items. As they interacted with things, they also needed to keep track of the player's health and collectible items. I gave them examples of each thing, but they were not limited to my examples. They also needed to implement some sort of win condition in the game to identify a goal for the player. Those were the minimum qualifications. It was very open-ended.

Students could earn an 85% by completing the minimum qualifications. To earn the last 15%, they needed to implement a second map and do something above and beyond the base qualifications. I gave students a rubric. In general, it communicated that they needed to use the C programming language, the concepts we learned in class, have a minimum of 10 functions to make sure that they broke their code down into a sufficient number of small pieces, implement the use of structs, implement the use of arrays, and provide sufficient comments. With their final submission, they also needed to record an 8-to-10-minute video, where they describe their code and play a piece of their game to demonstrate how it works.

If students used AI tools, they were expected to submit documentation of their interactions (e.g., prompts and responses). This requirement was in place so I could see how students used AI and how much they relied on using the AI program interface versus trying to do it themselves.

Dr. C: How did students use AI on the project?

Dr. E: I saw a wide range of AI use and there are some students who legitimately did not use AI. These students said they wanted to program it all from scratch.

In general, students were not very good at documenting how they used AI and we had a discussion about that in class. I did have one student who essentially had AI write the code for their

whole project. This student had extensive documentation. He submitted like 150 pages of documentation, but it was so useful to see. He really struggled with the AI trying to do what he wanted it to do, like 'Nope, I don't like that. Try this instead. Nope, I don't like that'. He even tried to go further, realized it was not working, and had to backtrack. He acknowledged that he probably relied on AI a little too much and that was reflected in his exam score, which was the Monday after the project was due.

A lot of students used AI to get started or to help debug their code, two things that are challenging for introduction to programming students. A majority of students did write some code. More specifically, they seemed to use AI to help implement some of the more challenging topics, such as structs and pointers. They also used it to help figure out what was going on in segments of code. They did a really great job of being creative, implementing something, and getting their code to work. I feel like AI helped them get farther than they would have gotten by themselves.

Students did not do a great job documenting their use of AI. For the second project, students were not writing code, instead, they took the code they wrote and then designed an object-oriented program. They were still allowed to use AI. I weighted the documentation portion more heavily to push them to actually do the documentation. I also could have demonstrated how I wanted the documentation to look and that probably would have helped them do better on the documentation.

IV. Instructor's Reflection on the Practice

Dr. C: Would you say AI was successful for the students who used it? Why or why not?

Dr. E: I think overall, it was successful. First, it was a tool that students could use at any hour of the day, to help them push past some of the challenges of writing a bigger program than they were used to. Second, I do think it helped them get farther than they would have gotten without it. I did, by setting minimum qualifications, try to push students to go farther and see what they could do. They came up with really creative games and every game was different. Third, the exam scores for the second exam were higher on average than the first exam and several students' scores improved. The second exam scores typically get worse, because the topics are much more challenging. I did revise my exam structure this semester, so I do not have exactly a one-to-one correspondence with previous semesters. In general, the second exam is harder, because it has harder material on it. The students did really well. There were a few students that did worse on the second exam but most students performed similar or better on the second exam compared to the first exam. One student had a much lower score and he is the one who used AI extensively. He even mentioned and recognized that he used it to his detriment, which I think is a great learning experience for him.

I do think the open-ended project and allowing students to use AI tools was more successful than anticipated. It helped students when they got stuck and pushed them farther on the project. It also allowed me to be more comfortable requiring students to implement some of the more complicated constructs, because I knew they had AI as a resource.

Dr. C: What are your general impressions on those submissions from the AI semester, as opposed to prior semesters?

Dr. E: As I said, I did change the project so it is not quite a one-to-one correspondence. In previous semesters, I had students write a game, but I gave them two or three options, so implement battleship or implement Connect Four; or something along those lines. In the past, I provided some general code as an example. I gave them an example Tic Tac Toe game to get them started. I feel like the students did not really branch out of their comfort zone. Many students took what I did in Tic Tac Toe, and just modified it a little bit in order to implement the game that they were implementing. There was not a lot of pushing the boundaries or implementing new concepts.

This semester, when I made the project more open-ended, I did not give them example code. They surpassed my expectations. They came up with some really creative games, and they implemented concepts more than I thought they were going to be able to. I am really impressed with pretty much everything that students were able to implement.

Dr. C: How do you think students' performance in the course this semester compares to the prior semesters? What do you think those differences are due to?

Dr. E: As I said about the second exam when I calculated the average for the second exam and got the results back, I was just shocked. We did make some adjustments from exam one to exam two. Both exams required writing code with LockDown Browser so they could not access anything else. However, for exam two, I provided them with some sample code for exam two. So, the problems were a little bit shorter, but had harder topics that students struggle with, so I was just amazed at their ability to do the second exam because I expected more struggles based on past experience.

For final grades, I do not think students on average did any better than students in the past. I would not say it was exceedingly different. I do think students really enjoyed the class more this semester than in previous semesters.

Dr. C: What are you planning to do with regard to AI for future semesters?

Dr. E: I am surprised at how positive of an experience it was. Personally, I am not resistant to technological change, but I am not ready to go, push the boundaries, and try and find something new to use so I was surprised at how generally positive the experience was. I am very much open to using it again, for a big project. I still like the idea of not using it for smaller homework assignments. I think students struggling is extremely important and essential to learning. I do not think that will ever go away, especially in an introductory class.

But for an open-ended project, I do think AI tools help students as long as I do a better job of requiring documentation and demonstrating what to expect. I am very much open to using it again in future semesters.

V. Connections to Research

Summary of the innovative practice – open-ended project and the inclusion of AI as a tool

Most classroom problems are well-structured with a limited number of ways for students to approach the problem. This structure is in contrast to the open-ended problems students will face in their future careers [8]. These open-ended problems require students to make judgments and decisions that they do not have to make on typical classroom problems [9]. An open-ended project was given to students in an introduction to programming course allowing students to use AI throughout the project. Minimum requirements were provided with the expectation that students were to go above and beyond the minimum requirements. Of the students that chose to use AI, one student relied heavily on AI to write their code with most students using AI as a tool to help with initial development and debugging. Overall, students appeared to benefit from the use of AI as anecdotally indicated by exam scores. Students' projects were also more unique and implemented more new concepts compared to the more well-structured projects assigned in previous semesters.

Use of ChatGPT as a tool for students – initial code generation, debugging, tutor

Prior research shows that students struggle in the beginning stages of writing a program where they need to develop a general solution, or algorithm, to solve the problem. Another area is debugging and making sure the program does not contain any errors and performs as expected [10]. Students in this course indicated that these areas were the primary motivators for using AI, especially on a large open-ended project. Additionally, students used it to help guide them with challenging topics, such as implementing structs. However, more work needs to be done in developing best practices for documentation by students to demonstrate how they used AI more explicitly.

It is important to note that prior to AI, many online resources were and still are available for programming help, including Stack Overflow, Github, and a plethora of "do my homework" websites. Programming courses are also a hotbed for academic integrity violations [11], [12]. The widespread use of AI is another emerging resource for students. The use by students in this course indicates that, when used appropriately, it can be helpful for students in open-ended projects.

Use of open-ended projects – students' agency, challenging students – zone of proximal development

Research in engineering education has begun to explore how students enact agency when solving open-ended problems [13]. This work defines student agency as "students being positioned with, perceiving, and acting on, opportunities to shape the knowledge-building work in their classroom community" [14, p. 1058]. Supporting students to develop as doers rather than receivers is a core goal of education, beginning in K-12 [14]. The structure of this project in an introduction to programming class gave students the opportunity to have agency in how they met the project requirements. As previously mentioned, students went above and beyond the project expectations and implemented new concepts. More work needs to be done to understand how this agency supported students' engagement with the project and their learning of course concepts.

VI. Future Work and Conclusions

At this point, we do not know if/how the project contributed to the improved exam two scores. Given the success of the project in its first implementation, Dr. E plans to use the assignment in future semesters. Additionally, Dr. C plans to collect data to begin exploring how the use of AI tools on this open-ended project supported students' learning and understanding of principles in the course. For this initial study, Dr. C plans to collect exam score data for students in the class to compare with previous semester. She also plans to interview students about core concepts from the project that tend to be more challenging for students (e.g., functions, arrays, pointers). These interviews will include short problems designed to assess students' knowledge of these core topics as well as questions to explore how students used AI on the project and what they learned through their experience doing the project and using AI. Open-ended projects combined with AI can be incorporated into a variety of engineering and computer science courses.

This paper presents the innovative practice of incorporating AI with an open-ended computer programming project. We describe how and why one instructor chose to encourage her students to use AI tools on their open-ended project to make this type of instructor decision visible. We also describe Dr. E's reflections on what the students learned as an initial means to support incorporating AI tools into certain classroom assessments and activities.

Acknowledgment

We thank the student assistant for the course, Zhi Yu, who helped develop the project and provided insights from a student's perspective. We thank the students for their effort on the project. We also thank the reviewers for their time and effort. We appreciate the recommendations they provided to strengthen this work.

References

[1] S. Frieder *et al.*, *Mathematical Capabilities of ChatGPT*. 2023. doi: 10.48550/arXiv.2301.13867.

[2] K. Hu, "ChatGPT Sets Record for Fastest-Growing User Base - Analyst Note," Reuters. Accessed: May 08, 2024. [Online]. Available: https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/

[3] OpenAI, "ChatGPT Prompt'How and why is GPT-3 better than GPT-1 OR GPT-2?'[Large languagemodel][Online]." Accessed: Feb. 19, 2023. [Online]. Available: https://chat.openai.com

[4] S. Nikolic *et al.*, "ChatGPT versus engineering education assessment: a multidisciplinary and multi-institutional benchmarking and analysis of this generative artificial intelligence tool to investigate assessment integrity," *Eur. J. Eng. Educ.*, vol. 48, no. 4, pp. 559–614, Jul. 2023, doi: 10.1080/03043797.2023.2213169.

[5] J. White *et al.*, *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. 2023. doi: 10.48550/arXiv.2302.11382.

[6] A. Gilson *et al.*, "How Does ChatGPT Perform on the Medical Licensing Exams? The Implications of Large Language Models for Medical Education and Knowledge Assessment." Dec. 26, 2022. doi: 10.1101/2022.12.23.22283901.

[7] J. H. Choi, K. E. Hickman, A. Monahan, and D. B. Schwarcz, "ChatGPT Goes to Law School," *SSRN Electron. J.*, 2023, doi: 10.2139/ssrn.4335905.

[8] D. H. Jonassen, "Engineers as problem solvers," in *Cambridge Handbook of Engineering Education Research*, A. Johri and Olds, Eds., Cambridge, England: Cambridge University Press, pp. 103–118.

[9] J. Swenson, K. Beranger, and A. W. Johnson, “How Students Take Up Open-ended, Real World Problems,” in *2021 IEEE Frontiers in Education Conference (FIE)*, Lincoln, NE, USA: IEEE, Oct. 2021, pp. 1–5. doi: 10.1109/FIE49875.2021.9637362.

[10] C. S. Cheah, “Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review,” *Contemp. Educ. Technol.*, vol. 12, no. 2, p. ep272, May 2020, doi: 10.30935/cedtech/8247.

[11] 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*, Bologna Italy: ACM, Jun. 2017, pp. 244–249. doi: 10.1145/3059009.3059064.

[12] E. Roberts, “Strategies for promoting academic integrity in CS courses,” in *32nd Annual Frontiers in Education*, Boston, MA, USA: IEEE, 2002, pp. F3G-14-F3G-19. doi: 10.1109/FIE.2002.1158209.

[13] C. Schimpf, E. Komolafe, and J. Swenson, “Exploring how Students Grapple with Agency in Open-Ended Engineering Problems,” in *2023 ASEE Annual Conference & Exposition Proceedings*, Baltimore , Maryland: ASEE Conferences, Jun. 2023, p. 43613. doi: 10.18260/1-2--43613.

[14] E. Miller, E. Manz, R. Russ, D. Stroupe, and L. Berland, “Addressing the epistemic elephant in the room: Epistemic agency and the next generation science standards,” *J. Res. Sci. Teach.*, vol. 55, no. 7, pp. 1053–1075, Sep. 2018, doi: 10.1002/tea.21459.