Transcription-Enhanced AI Code Review for Deeper Learning

Nathan Whittenburg and Bobby Hodgkinson
Smead Aerospace Engineering
University of Colorado Boulder
hodgkinr@colorado.edu

Abstract

This project augments entry level programming education by integrating AI-driven audio transcription and feedback to emphasize code comprehension in a structured review setting. This approach acknowledges the increasing prevalence of AI tools in coding environments, shifting the focus from manual line-by-line code creation to understanding, analyzing, and refining AI-generated outputs. This project seeks to equip students with the skills necessary to thrive in AI-augmented programming environments.

The project consists of two core components. First, AI-supported code comprehension exercises where students verbally describe their own written code while an AI transcription tool captures their explanations and provides feedback on clarity and understanding. This component is akin to traditional 'code reviews' which are common in educational and professional settings. The second component involves students verbally describing unique AI-generated code using the same transcription method. While similar to the first component, this secondary aspect of the project requires students to read code that they did not generate themselves prior to the review. This component is akin to using an AI chatbot to aid in the development of code; a skill that is becoming increasingly prevalent with students and in industry.

The AI-assisted code review component leverages recent advancements in AI to provide meaningful, constructive, and timely feedback on student assignments. The system is implemented in-house, providing the instructional team with full visibility into student interactions and ensuring that feedback aligns with the course's pedagogical goals. This approach addresses the challenge of scaling feedback in large programming courses, where providing high-quality, personalized feedback can be resource intensive. Customized AI chatbots can be valuable additions to traditional course resources[1]. Tools like the CS50 Duck[2] and Code in Place[3] demonstrate the potential of AI to emulate a good teacher by guiding students toward solutions rather than providing solutions directly. By using AI to provide timely feedback, educators can focus on more complex pedagogical issues and provide targeted support to students[4]. This aligns with findings that AI-generated feedback can be rated positively by students, sometimes even more so than human feedback, provided it is constructive and accurate[5].

The AI-supported code review exercises are specifically designed to strengthen students' analytical and debugging skills by engaging with code they did not author themselves. With the rise of "vibe coding," where students prompt AI to write, run, and debug code, it is crucial

to ensure students can effectively read, interpret, and modify generated code[6]. The use of audio transcription tools allows for an effective analysis of student comprehension, capturing their thought processes and identifying areas of confusion. By focusing on code comprehension, the project reinforces computational thinking and structured problem-solving, preparing students for AI-augmented programming environments. Additionally, this portion of the project allows for a personalization aspect which has been shown to be a worthwhile addition to introductory programming courses[7].

Early stage testing of this approach has revealed that while AI-driven feedback offers substantial benefits in terms of cost and response time, it may occasionally diverge from course-specific learning objectives or provide confusing feedback. As such, the AI-augmented code review process is most effective when used in conjunction with human oversight. Rather than replacing instructional staff, AI tools should serve as a complement by supporting the evaluation process through delivering scalable, timely, and personalized feedback that instructors can then contextualize and refine[8]. This balance ensures that students receive guidance that is both pedagogically sound and tailored to their learning journey.

This project offers a modernized augmentation to programming education, shifting the focus from rote code writing in a line-by-line fashion to code comprehension, structured review, and modern-day programmatic engagement. By integrating AI-driven feedback and audio transcription, the project prepares students for AI-augmented programming environments, ensuring they develop the critical evaluation skills necessary to work effectively alongside AI tools. These exercises also improve students' debugging proficiency by encouraging them to think aloud and reflect on logic, fostering a deeper understanding of how and why code works or where provided code could be improved. By interacting confidently with AI-generated code, students gain experience in a workflow that mirrors real-world development practices, ultimately boosting their confidence in using AI as a collaborative programming partner. Additionally, this project allows timely, and personalized feedback in a resource efficient fashion.

References

[1] Yutong Ai, Maya Baveja, Akanksha Girdhar, Melina O'Dell, and Andrew DeOrio. A custom generative ai chatbot as a course resource. In *ASEE Annual Conference*. American Society for Engineering Education, 2024.

[2] Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J. Malan. Teaching cs50 with ai: Leveraging generative artificial intelligence in computer science education. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024)*. ACM, 2024.

[3] Isabel Swafford. AI tool streamlines feedback on coding homework, 2021. URL `https://news.stanford.edu/stories/2021/07/ai-tool-streamlines-feedback-coding-homework`.

[4] Mike Wu, Chris Piech, and Chelsea Finn. Meta-learning student feedback to 16,000 solutions, 2021. URL `https://ai.stanford.edu/blog/prototransformer/`. Stanford AI Lab.

[5] Dong-Kyu Lee. A gpt-based code review system for programming language learning. *arXiv preprint arXiv:2407.04722*, 2024. URL `https://arxiv.org/abs/2407.04722`.

[6] Darryl K. Taft. Vibe coding: Where everyone can 'speak' computer programming. *The New Stack*, February 2025. URL https://thenewstack.io/vibe-coding-where-everyone-can-speak-computer-programming/.

[7] Evanfiya Logacheva, Arto Hellas, James Prather, Sami Sarsa, and Juho Leinonen. Evaluating contextually personalized programming exercises created with generative ai. In *ACM Conference on International Computing Education Research V.1 (ICER '24 Vol. 1)*. ACM, 2024.

[8] Chloe Long and Bobby Hodgkinson. Ai-driven concept assessment framework and application in aerospace engineering sophomore lab. In *Proceedings of the ASEE Rocky Mountain Section Conference*, Boulder, CO, May 2025. To appear.