Using Personalized Generative AI as a "First Responder" in C++ Education to improve student learning

Sonya M. Dennis, Ph.D.
Computer Science/Software Engineering
Morehouse College
Atlanta Georgia USA
sonya.dennis@morehouse.edu

Juana Mendenhall, Ph.D.
Chemistry
Morehouse College
Atlanta Georgia USA
juana.mendenhall@morehouse.edu

ABSTRACT

Using Generative AI in the classroom to improve student learning requires deliberate, structured integration into course activities and assessments. This paper explores the integration of a specialized Generative AI mentor, **Dennis AI**, into three sections of an undergraduate **Computer Programming II** course. The mentor was developed in conjunction with **ibl.ai**, a family-owned and operated company located in the technological hub of New York that specializes in building AI-driven, revenue-generating systems for the educational sector, serving learners from over 400 universities. The research evaluates a pedagogical shift from traditional passive learning to an active, generative model where AI serves as a "First Responder" for complex technical concepts like memory management and object-oriented design. By leveraging the ibl.ai platform to synthesize course-specific data, the study demonstrates how structured prompt-based inquiry and iterative code generation can bridge the "complexity wall" often encountered in mid-level computer science curricula[1]. Furthermore, the study illustrates how the integration of AI mentors and digital avatars---mapped to **Bloom's Taxonomy 4.0**---facilitates a scalable model for high-level architectural mentoring and diagnostic practice[2]. Ultimately, this study posits that the strategic deployment of specialized Generative AI tools creates a dynamic environment that encourages deep inquiry, competitive innovation, and a rigorous validation process that preserves academic integrity.

CCS CONCEPTS

- **Social and professional topics → Computing education**; *C++ Object-Oriented Programming*.
- **Computing methodologies → Artificial intelligence**; *Generative AI mentorship*.

KEYWORDS

Dennis AI, Generative AI, Undergraduate Pedagogy, C++ Programming, AI-PiLOT Fellowship, ibl.ai.

ACM Reference format:

Sonya M. Dennis, Juana Mendenhall, 2026. Enhancing Undergraduate Programming Pedagogy through Generative AI Mentorship: Using specialized Generative AI as a "First Responder" in C++ Education. In *Proceedings of ACM Woodstock conference (WOODSTOCK'18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

1 Introduction

The rapid evolution of Large Language Models (LLMs) has created a paradigm shift in computer science education[1][3]. In **Computer Programming II**, a course traditionally focused on data structures, memory management, and object-oriented design, students often encounter a "complexity wall". To mitigate this, an educational framework was developed centered on a Generative AI mentor, **Dennis AI**.

As an **AI-PiLOT Fellow**, I participated in a strategic program that integrated AI mentors and avatars into Morehouse College's Liberal Arts curriculum through a pilot study across Computer Science, Education, Business, and Online Education departments. Using the **ibl.ai** and Canvas LMS platforms, four faculty fellows developed AI-enhanced course modules implementing **Bloom's Taxonomy 4.0** along with active generative learning activities[2].

The mentor was developed in conjunction with **ibl.ai**, which specializes in AI for education. This specialized platform was designed to provide 24/7 personalized guidance and iterative code generation based on specific course materials, synthesizing information across lecture notes and textbooks to provide cohesive explanations. Research shows that generative AI tools like ChatGPT and specialized educational AI platforms are transforming computer science instruction by providing real-time assistance that enhances the learning experience through personalized, context-aware tutoring[4].

2 Developing Effective AI-Enhanced Curricula

Effective curriculum development in the Generative AI era requires moving beyond simple automation[1][3]. For this study, the curriculum was modified to include AI-assisted problem-solving modules where the AI acts as a creative and diagnostic partner:

- **Integrated Knowledge Synthesis**: Dennis AI utilized Generative AI to bridge information across lecture notes and textbooks to provide cohesive, context-aware explanations. Recent research demonstrates that Retrieval-Augmented Generation (RAG) is a proven approach that allows AI systems to offer better explanations and context-aware tutoring by adding course-specific context to large language models based on instructor materials and conversation history[4].

- **Prompt Engineering as a Skill**: Students were taught that AI output quality is directly proportional to input clarity, turning interaction into a lesson on logic and requirements gathering. Prompt engineering is emerging as a critical pedagogical skill in computer science education, where students learn to systematically design and refine prompts to improve code quality, problem-solving abilities, and understanding of algorithms[5][6]. This approach develops metacognition, critical thinking, and computational literacy while teaching students to use AI tools wisely and ethically[6].

3 Mentoring and Stakeholder Engagement

The introduction of Dennis AI facilitated a multi-tier mentoring model within the institution:

- **Student Mentoring**: Dennis AI acted as a "First Responder," providing immediate scaffolding for syntax errors or logic bugs to reduce "learned helplessness" during late-night study sessions. AI-powered tutoring systems can engage students in interactive dialogue, provide explanations in personalized terms, analyze student responses for accuracy and depth of comprehension, and offer immediate feedback with personalized follow-up questions[2].

- **Faculty Integration**: By offloading repetitive syntax and conceptual questions to the AI, faculty could focus on high-level architectural mentoring and complex debugging that requires human intuition. This aligns with research showing that AI integration reduces course administration costs and allows instructors to focus on mentoring and facilitation rather than routine tasks[7].

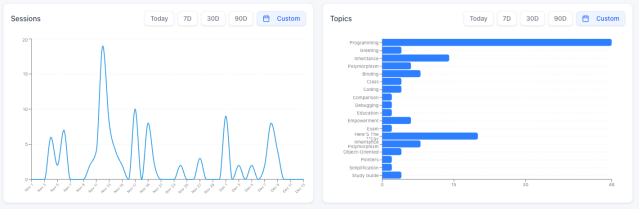
4 Methodology: Implementation and Bloom's Taxonomy 4.0

The methodology focused on integrating the **Dennis AI** mentor directly into the **Canvas LMS** environment, ensuring that the AI had access to a curated "knowledge base" consisting of specific lecture slides, C++ documentation, and textbook chapters [7]. This restricted data set was critical in reducing hallucinations and ensuring the mentor's advice remained aligned with course-specific coding standards.

A core component of the research involved a mandatory interaction assignment across three undergraduate sections. Students were required to engage with Dennis AI using a structured prompt list and record the responses.

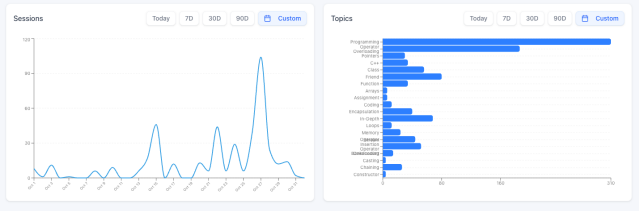
- **Intentional Debugging**: Students specifically tasked Dennis AI to generate code with errors (e.g., memory leaks or dangling pointers) to practice diagnostic skills.
- **Student-Led Practice Design**: The assignment culminated in the student asking the mentor to design a comprehensive prompt to practice writing a full program using friend classes, static members, and copy constructors.

Graph 1: Dennis AI – C++ Engagement Metrics (Oct 1–31)



- Sessions increased from near zero early in the month to multiple spikes mid- and late-October, with the largest peak reaching roughly 100 sessions on a single day (about 20–25% of all October sessions).
- Programming-tagged interactions dominate the topic distribution, accounting for approximately 70–75% of labeled queries, while all other topics (pointers, classes, memory, in-depth questions, etc.) share the remaining 25–30%.

Graph 2: Dennis AI – C++ Engagement Metrics (Nov 1–Dec 13)



- In November–December, overall traffic is lower, with the tallest spike around 18–20 sessions (roughly 15–20% of all sessions in this period), again clustered near likely exam dates.

- Programming remains the primary topic, representing about 60–65% of labeled queries, with the next largest category ("Exam"/study-guide oriented prompts) contributing roughly 20–25%, and all remaining topics (inheritance, polymorphism, debugging, empowerment, etc.) together forming about 10–20%.

Central to the methodology was the application of **Bloom's Taxonomy 4.0**, which reimagines cognitive levels for an AI-integrated classroom[2][8]. Rather than simply using AI to *Remember* (Level 1) or *Understand* (Level 2) syntax, the assignments were designed to push students toward *Evaluating* and *Creating* (Levels 5 and 6):

- **LMS-Embedded Mentorship**: Dennis AI was deployed via a customized LTI integration, allowing students to prompt the AI without leaving their coding environment[7].
- **Structured Interaction Logs**: Students were required to submit "Prompt Logs," detailing how they moved from an initial broad query to a refined, specific technical request. This iterative refinement process mirrors the engineering design process and cultivates problem-solving skills by requiring students to analyze why AI-generated responses don't meet expectations and refine their prompts accordingly[5][6].
- **Diagnostic Verification**: Methodology included "AI-Validation" phases where students had to cross-reference AI-generated code snippets with class lecture notes to identify potential logical discrepancies. This addresses concerns about AI hallucinations and error propagation by ensuring students develop the critical thinking skills necessary to evaluate AI-generated content[4].

5 Learning and Innovating via Out-of-Class Assignments

Rather than traditional classroom exercises, the course utilized structured out-of-class assignments to drive innovation and self-paced mastery:

- **AI-Optimized Debugging Sessions**: Students were assigned complex codebases and utilized Dennis AI as a remote partner for real-time debugging and performance optimization.
- **Architectural Prompt Challenges**: These assignments required students to iteratively refine prompts to elicit accurate and concise architectural designs from the mentor. By designing creative prompts, students explored novel ideas while merging creativity with real-world application[5].

6 Conclusion

The integration of **Dennis AI** into the **Computer Programming II** curriculum represents a significant advancement in undergraduate CS pedagogy[1][3]. By transitioning from a traditional lecture-heavy model to an active-generative framework, this study demonstrates that specialized AI mentors do not replace foundational instruction but rather amplify it. As a direct result of the **AI-PiLOT Fellows Program**, the curriculum was successfully mapped to **Bloom's Taxonomy 4.0**, moving students beyond mere syntax memorization toward the higher-order skills of evaluation and creation[2][8].

The collaboration with **ibl.ai** was instrumental in ensuring that the AI mentor remained "grounded" in course-specific data, effectively neutralizing the risk of hallucinations that often plague general-purpose LLMs[4][7]. The use of Dennis AI as a "First Responder" successfully mitigated student frustration during complex modules such as dynamic memory allocation and pointer arithmetic. Furthermore, the out-of-class assignments provided a safe, iterative environment for students to fail and learn in real-time, fostering a culture of resilience and deep inquiry.

The data suggests that students used Dennis AI **primarily as a programming support tool (≈65–75% of queries) and most intensely during short windows around assessments (single-day spikes contributing up to a quarter of monthly sessions)**, rather than as a steady source of conceptual exploration throughout the term. From an instructional design perspective, this pattern indicates that while the mentor successfully supports just-in-time problem solving, additional scaffolds may be needed to encourage more evenly distributed, concept-level engagement that aligns with desired higher-order learning outcomes

Ultimately, this research posits that when Generative AI is deployed as a transparent, structured mentor rather than a black-box shortcut, it preserves academic integrity while significantly enhancing the scalability of personalized instruction. The findings suggest that this model is not only effective for computer science but serves as a blueprint for AI integration across the Liberal Arts and beyond[1][3].

7 Future Work

Following the success of this pilot, future research will focus on scaling the **Dennis AI** mentor to support upper-division **Data Structures and Algorithms** courses. This expansion will include the development of specialized modules for visualizing recursive logic and graph theory. Additionally, the

author intends to conduct a longitudinal study to measure the impact of AI-assisted conceptual foundations on student retention rates and performance in senior capstone projects.

Future research should also examine the long-term effects of AI-driven formative feedback on student learning outcomes and explore optimal combinations of AI and human input in summative assessments[4]. As the field evolves, comprehensive frameworks for assessing both technical accuracy and creative aspects of student work in AI-augmented environments will become increasingly important[4].

Taxonomy to understand AI Adoption in Higher Education. *OLC Insights*. https://onlinelearningconsortium.org/olc-insights/2025/10/blooms-for-ai-adoption/

ACKNOWLEDGMENTS

The author wishes to acknowledge the **Morehouse College AI-PiLOT Fellows Program**. This research was made possible through a strategic pilot study that integrated AI mentors and avatars into the Liberal Arts curriculum. Special thanks to **ibl.ai** for providing the technological infrastructure and to the fellow faculty members whose collaborative efforts across the Business, Education, and Science departments helped refine the implementation of **Bloom's Taxonomy 4.0**.

REFERENCES

[1] Porter, L., & Zingaro, D. (2024). Computer Science Education in the Age of Generative AI. *arXiv*. https://scale.stanford.edu/ai/repository/computer-science-education-age-generative-ai

[2] University of Maine at Presque Isle. (2024). Using Bloom's Taxonomy to Support Teaching & Learning With AI. *UMPI Instructional Design*. https://wp.umpi.edu/instructional-design/using-blooms-taxonomy-to-support-teaching-learning-with-ai/

[3] University of California San Diego. (2025). Transforming Computer Science Education in the Age of AI. *UC San Diego Today*. https://today.ucsd.edu/story/transforming-computer-science-education-in-the-age-of-ai

[4] Chen, Y., et al. (2025). A Review of Generative AI in Computer Science Education. *arXiv*. https://arxiv.org/html/2507.11543v1

[5] New York Institute of Technology. (2025). Prompt Engineering Lesson Plan For Computer Science Students. *Emerging New York Tech*. https://sites.google.com/nyit.edu/emergingnewyorktech/ai-resources/prompt-engineering-lesson-plan-for-computer-science-students

[6] Sikka, M. (2024). The Prompt Engineer: Shifting Student Roles in AI-Powered Project-Based Learning. *Defined Learning Blog*. https://blog.definedlearning.com/the-prompt-engineer-shifting-student-roles-in-ai-powered-project-based-learning

[7] ibl.ai. (2025). Agentic LMS - AI-Powered LMS with Built-In Mentors. https://ibl.ai/product/agentic-lms

[8] Online Learning Consortium. (2025). Using Bloom's Taxonomy to understand AI Adoption in Higher Education.