2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #53395

Evaluating the Role of Research-Inspired Projects in Lower-Division Programming Course

Dr. Sanjana Mudduluru, The University of Oklahoma
Dr. Javeed Kittur, The University of Oklahoma

Javeed Kittur is an Assistant Professor in the Engineering Pathways program at the University of Oklahoma and serves as the Co-Director of Research in the Honors College, bringing a global academic background that includes a Ph.D. in Engineering Education Systems and Design from Arizona State University. Prior to joining OU, he served as an Assistant Professor in Electrical and Electronics Engineering at KLE Technological University, India, and is a certified IUCEE International Engineering Educator and Associate Editor for the Journal of Engineering Education Transformations. Since 2022, he has mentored over 40 undergraduate engineering students, many of whom have published in leading conferences and journals, demonstrating his strong commitment to impactful student mentorship. His excellence in teaching and mentoring has been recognized with multiple prestigious awards, including the 2026 Provost's Distinguished First-Year Student Mentoring Program - Outstanding Mentor Award, 2025 Nancy L. Mergler Faculty Mentor Award, the 2025 Brandon H. Griffith Outstanding Faculty Member Award, and the 2024 SGA Outstanding Faculty Award. Dr. Kittur's research advances engineering education through innovative, data-driven approaches, with a focus on generative AI integration, learning analytics, student retention, broadening participation, and holistic faculty development across cognitive, affective, and psychomotor domains.

Dr. Sean N Mackay, The University of Oklahoma

WIP: Evaluating the Role of Research-Inspired Projects in Lower-Division Programming Courses

Abstract

Computing Education's reliance on highly structured, syntactically focused assignments in introductory programming (CS1/CS2) often limits students' exposure to ill-defined, complex, and real-world problems, which can hinder the transfer of computational concepts and negatively impact student motivation. This Work in Progress paper presents a systematic investigation into the impact of integrating research-inspired, open-ended programming projects into a high-enrollment, lower-division computer science course at a large public university. Our methodology is grounded in providing structured support mechanisms—including clearly defined milestones, mandatory peer feedback, mentoring sessions, and architectural design documentation—to guide novice programmers through the inherent ambiguity of open-ended design while reducing common barriers to self-efficacy. Using a mixed-methods approach, the study evaluates quantitative performance data and conceptual test scores alongside qualitative reflective surveys measuring motivation and persistence. Preliminary analysis aims to determine how varying degrees of project independence affect different student populations, particularly those with weaker prior programming experience. The full paper will provide early results detailing project outcomes and shifts in student self-reports, offering evidence-based recommendations for faculty seeking to embed authentic, research-inspired work into introductory curricula to cultivate robust computational thinking skills in early-career engineers.

1. Introduction

Introductory programming curricula (CS1/CS2) frequently prioritize syntax-driven, well-defined logic problems over the iterative, non-linear nature of real-world engineering challenges. While this foundational approach equips students with the essential "grammar" of code, it often inadvertently limits exposure to the ill-defined problems common in professional practice [1]. A significant consequence of this narrow focus is a deficit in transferability—the ability to apply core computational concepts to novel or ambiguous contexts. This lack of transfer can negatively impact student motivation and inhibit the development of creative problem definition skills [2]. To address this gap, this "Work in Progress" (WIP) study investigates the integration of research-inspired, open-ended programming projects into a high-enrollment, lower-division course, aiming to bridge the divide between theoretical academic exercises and the demands of professional engineering environments.

2. Background and Framework

The transition from prescriptive assignments to open-ended projects is rooted in the principles of Project-Based Learning (PBL). Literature suggests that managing complexity through PBL can improve collaboration, critical thinking, and algorithmic reasoning [3]. However, for novice programmers, a lack of structure can lead to cognitive overload and frustration. Our framework is grounded in Scaffolding Theory, specifically targeting the reduction of "search and decision hurdles" that often paralyze beginners when faced with open-ended tasks [4]. By providing a highly structured pathway through an inherently unstructured research problem—using milestones and mentorship—we aim to maintain student self-efficacy while simultaneously increasing the technical and conceptual complexity of the curriculum.

3. Methodology

This study is conducted at a large central public university with a cohort of 92 students over an eight-week instructional period. The primary intervention replaces traditional, highly structured programming assignments with research-inspired, open-ended projects that challenge students to engage with computational concepts in a professional engineering context. This pedagogical approach requires students to navigate three developmental phases: first, defining a specific problem scope by transforming a vague research prompt into a concrete programming objective; second, producing a comprehensive architectural design document to prioritize structural planning over immediate code implementation; and finally, executing the project through iterative development milestones, benchmarking, and final releases. This structured workflow is intended to move students beyond simple syntax proficiency toward holistic algorithmic reasoning and project management.

To support novice programmers through the inherent ambiguity of open-ended design, the curriculum incorporates a multi-tiered scaffolding framework consisting of peer feedback, graduate mentorship, and strict progress milestones. Students participate in weekly mandatory peer feedback sessions to review design documents, ensuring technical uniformity and maintaining high standards through collaborative critique. Simultaneously, mentoring sessions led by Graduate Student Guides provide a space for brainstorming and technical troubleshooting; this arrangement not only fosters mentorship skills in the graduate assistants but also exposes undergraduate students to authentic research methodologies. The impact of this framework is evaluated using a mixed-methods approach. Quantitative assessment includes student performance scores across project deliverables and pre/post-tests measuring conceptual understanding and self-efficacy. These data are integrated with qualitative insights from mid- and end-of-project reflective surveys.

4. Future Work

As this study continues, our future work will focus on two primary areas:

- We will complete the coding of the remaining student responses to validate whether the themes of "collaborative troubleshooting" and "ambiguity management" persist across the entire cohort.
- Finally, we will compare the final exam scores and conceptual test results of this cohort against historical data from previous semesters that utilized traditional, highly structured assignment models. This will provide empirical evidence regarding the impact of research-inspired projects on the long-term transferability of computational skills.

References

[1] C. Izu, A. Cruz, and C. Mirolo, "Learning transfer in novice programmers: A preliminary study," in *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, 2021.

[2] S. Sharmin, "Creativity in CS1: A literature review," *ACM Transactions on Computing Education (TOCE)*, vol. 22, no. 2, pp. 1-26, 2021.

[3] J. Peng et al., "Computer-based scaffolding for sustainable project-based learning: Impact on high-and low-achieving students," *Sustainability*, vol. 14, no. 19, p. 12907, 2022.

[4] I. Newman, M. Daniels, and X. Faulkner, "Open ended group projects a 'tool' for more effective teaching," in *Proceedings of the fifth Australasian conference on Computing education-Volume 20*, 2003.