Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #46547

Connecting the Dots - Exploring the Benefits of Cross-Disciplinary Learning in First-Year Programming Courses

Prof. Salma Emara, University of Toronto

Salma Emara is an Assistant Professor, Teaching Stream in the Department of Electrical and Computer Engineering at the University of Toronto. She received her B.Sc. in Electronics and Communications Engineering from the American University in Cairo in 2018, and her Ph.D. in Computer Engineering from the University of Toronto under the supervision of Professor Baochun Li in 2022. Her Ph.D. research focuses on improving reinforcement learning algorithms to solve problems in computer networking algorithms. Currently, her research focuses on developing pedagogical practices to enhance debugging skills for beginner programmers and utilizing natural language processing in engineering education. She believes that engineers learn by doing, which makes her committed to engaging students through in-class activities and problem-solving assignments and projects. She strives to create inclusive learning environments for all students from different backgrounds.

Dr. Fatemeh Jazinizadeh, University of Toronto
Dr. Hamid S Timorabadi P.Eng., University of Toronto

Hamid Timorabadi received his B.Sc, M.A.Sc, and Ph.D. degrees in Electrical Engineering from the University of Toronto. He has worked as a project, design, and test engineer as well as a consultant to industry. His research interests include the application of digital signal processing in power systems.

Connecting the Dots - Exploring the Benefits of Cross-Disciplinary Learning in a First-Year Programming Course

Abstract

Engineering students often perceive their first-year course as disconnected, leading to disengagement with subjects they view as irrelevant to their fields. This issue is amplified when course assignments focus narrowly on specific concepts without showcasing interdisciplinary or real-world applications. However, as students progress in their studies and careers, they often recognize the interconnectedness of these foundational topics. In this work, we investigate the impact of integrating interdisciplinary concepts into programming labs and assignments, drawing on principles from Dynamics and Electrical Fundamentals. By embedding these core engineering concepts into a first-year programming course, we aim to foster a broader perspective, enhance problem-solving skills, and spark greater engagement among students. Our approach involved designing assignments that incorporated cross-disciplinary content and others without such integration, enabling comparative analysis. Anonymous surveys captured students' perceptions, learning experiences, and engagement levels. A mixed-methods research design combined quantitative survey data with qualitative feedback. Our results reveal that 78% of students felt cross-disciplinary assignments enhanced their understanding of real-world programming applications, while 92% recognized connections between programming and concepts from Dynamics and Electrical Fundamentals. Moreover, 60% found these assignments more engaging, with one student noting they provided "a taste of what upper-year projects might look like." We conclude that incorporating cross-disciplinary content enhances the student learning experience through observing fields intertwining and their knowledge applied to real-world applications.

Introduction

Engineering students often perceive different subjects that they study are unrelated, particularly in their first year of studies. This perception can lead to a lack of engagement with courses they deem irrelevant to their chosen fields. The issue is exacerbated when course assignments focus narrowly on specific concepts, without demonstrating real-world relevance or interdisciplinary connections. As a result, students may struggle to appreciate the broader applicability of these foundational topics taught in various courses [1]. Perhaps later as students progress in their studies or when they transition into professional roles, they often realize how related the concepts they learned truly are.

Research shows that students learn better when they can see clear, real-world connections among the topics they study [2]. However, creating strong links between foundational courses requires thoughtful curriculum design that highlights shared principles and interdisciplinary applications. For example, Schulz *et al.* [6] introduced interdisciplinary learning in design courses to meet sustainable development goals in design projects [7]. In some cases, it requires creating new courses as Baker et al. did to promote an interconnected view of concepts [8]. Also, linking topics from different fields together is not restricted to a particular level as prior efforts show its effectiveness in all levels [3-5].

The problem is in many introductory programming courses the focus often remains on teaching syntax, algorithmic thinking, and basic programming constructs. Although crucial for technical competence, this narrow approach can inadvertently de-emphasize how programming skills integrate with other engineering disciplines, such as Dynamics and Electrical Fundamentals. As a result, students may struggle to see immediate connections between their programming assignments and practical applications in engineering contexts—thereby contributing to disengagement and a lack of motivation.

In this study, we explore the impact of integrating interdisciplinary concepts from other courses students are currently taking into the design of programming labs and homework assignments. Specifically, we incorporated principles from Dynamics and Electrical Fundamentals into a first-year programming course. Our approach reflects a belief that students engage in a Problem-Based Learning approach that promotes transferable skills, fosters a more holistic understanding enhancing students' problem-solving skills, ultimately producing well-rounded engineers capable of addressing complex challenges [9].

The goals of this work are threefold: (1) to develop a broader perspective on the relevance of programming to engineering tasks, (2) to provide opportunities for students to apply theoretical knowledge to practical scenarios, reinforcing their ability to transfer concepts across different contexts, and (3) to engage students with cross-disciplinary content, sparking curiosity and deeper engagement with the material.

Our research questions are:

1. Does incorporating cross-disciplinary content in programming labs improve students' perceptions of real-world applications of programming?
2. Are students' knowledge getting reinforced when exposed to interdisciplinary programming assignments?
3. To what extent does embedding Dynamics and Electrical Fundamentals concepts affect student engagement and motivation in programming courses?

This approach may serve as a framework for integrating interdisciplinary learning in other foundational courses, demonstrating how targeted cross-disciplinary assignments can improve both student engagement and conceptual understanding. This paper's findings have implications for curriculum designers, instructors, and administrators who seek to cultivate a more cohesive and applied engineering education experience.

Methodology

Design of the Assignments

We designed a series of programming assignment questions for a first-year programming course, i.e. APS 105 Computer Fundamentals, that draws on concepts from MIE 100 Dynamics and ECE 110 Electrical Fundamentals. We call these questions related-to-other-courses questions. Other assignment questions were created without interdisciplinary connections, which we call "conventional" assignments or questions. There are three programming assignment types: a programming assignment with (i) only conventional questions, (ii) only related-to-other-courses questions, or (iii) both kinds of questions. The presence of a variety of questions in programming assignments allow for a comparative analysis as inspired by Prince *et al.* [10].

A team of instructors teaching programming, dynamics and electric fundamentals in first year engineering met to discuss the concepts discussed in dynamics and electric fundamentals that we can create a program for. We did not want to introduce new concepts that have not been taught to students as this defeats the purpose of engaging students with other familiar topics. We also ensured that we did not create questions on topics not yet discussed in the dynamics or electric fundamental courses. Topics we focused on were kinematics and motion in space in dynamics and electric forces and circuit analysis in electric fundamentals.

Question 1: In another question, students were asked to calculate the escape velocity, which is the minimum initial velocity to escape a planet's gravitational pull, of a particular planet. The programming concepts required are using math library functions and arithmetic operations. Students were told about this program for a small probe that is exploring a mystery planet as part of a space exploration initiative to link it to a real-world application.

Question 2: A question designed to link programming with electrostatic forces asked students to calculate the force along with the units between two particles of known charges and distance between them. This required students to apply the Coulomb's law equations after doing the appropriate unit conversions, and print the force with appropriate units. The programming concepts required are using math library functions, nested-if statements to determine the appropriate unit conversions at the time of calculating and printing the force. Students were told this program is helping a scientist in a lab determine the force quickly to link it to a real-world application.

Question 3: The question that related kinematics to programming was asking students to find the angle at which a robot has to throw a ball to score a goal in basketball. This required knowledge of projectile motion and loops in programming. Students had to program a code that loops over angles and plug it in the projectile motion equations until it finds the angle that scores a goal. The question was framed as a story where students are told this is part of a robot basketball shooting competition to link it to a real-world application. The projectile motion equations were given to students to help them review the concepts.

Question 4: In another programming assignment, students were asked to write a program for a circuit simulator. The circuit simulator stores the values of resistors in a circuit in series to insert/remove resistors, calculates the current across resistors and the voltage across one resistor. This requires knowledge of Ohm's law and circuit analysis from electric fundamentals in addition to linked lists in programming to add data structures of resistors into the list/circuit.

In all of the related-to-other-courses questions, first, we were relating coding tasks to real-world engineering scenarios that students were already exposed to in other courses. Second, by leveraging programming skills in other disciplines, we aimed to have students transfer their foundational knowledge of other courses to programming and vice versa. Third, linking multiple areas of study fosters curiosity and sense of purpose increasing engagement with material studied in the curriculum.

These questions were mixed with other conventional questions such as finding the arrival time of a trip given the start time and the length of the trip, setting rules for lucky numbers and writing a program that finds if a number is lucky, playing the word puzzle using 2D arrays, and estimating the value of pi using Monte Carlo estimation technique. These questions were added to have allowed students to reflect on their experience by comparatively analysing the difference.

Measuring the Effectiveness

Anonymous surveys were administered to assess students' perceptions of the integrated approach, its impact on their learning, and overall satisfaction. Ethical approval for this study was obtained from the University of Toronto under protocol number RIS Protocol Number 46956. A mixed-methods research design was employed, combining quantitative survey data with qualitative feedback from open-ended questions.

The goal of the survey was to ask the students on their experience to answer our research questions. For our first research question, "Does incorporating cross-disciplinary content in programming labs improve students' perceptions of real-world applications of programming?", we asked students to what extent related-to-other-courses questions helped them link programming concepts and dynamics and electric fundamentals concepts together and did related-to-other-courses questions allow the student to see real-world applications.

For our second research question, “Are students’ knowledge getting reinforced when exposed to interdisciplinary programming assignments?”, we asked students to what extent “related-to-other-courses” programming questions reinforce knowledge of other engineering concepts.

For our third research question, “To what extent does embedding Dynamics and Electrical Fundamentals concepts affect student engagement and motivation in programming courses?”, we asked students to what extent including “related-to-other-courses” content made programming lab assignments more interesting and engaging compared to the “conventional” assignments.

To help us determine if there is bias, we asked students to what extent was it easy to understand related-to-other-courses courses and conventional questions.

Our open-ended questions involved asking if there were any challenges students faced doing related-to-other-courses questions and if there are any other suggestions they have.

The survey was distributed to 550 students taking a first-year programming course at the end of the semester shortly before final exams, and after the last lab of the course was released. For survey questions that have asked for a rating, the score is 1 for strongly disagree, 2 for disagree, 3 for neutral, 4 for agree and 5 for strongly agree.

Results and Discussion

Out of 550 students, 50 students participated in the survey. Looking at how students perceive the related-to-other-courses questions and conventional questions, students thought that both questions had the same level of difficulty. The responses show that students rated the easiness of conventional and related-to-other-courses questions 4.36 out of 5 on average. We can safely assume that the level of difficulty did not affect students' perception in determining their experience.

In Fig. 1, we show the students answers to if students were able to see the connection between the courses in related-to-other-courses questions. 92% students were able to see the connection between the courses, which allows us to ask these students if this connection had an impact on their experience.

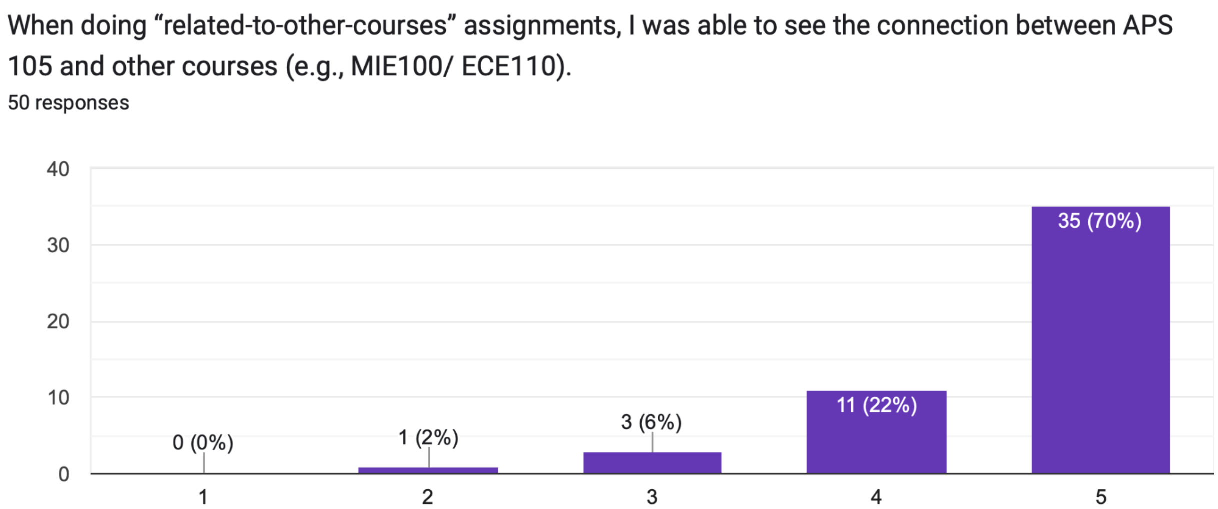


Fig. 1. Were students able to see a connection between the programming course (APS 105) and the other courses (MIE 100 and ECE 110)?

In Fig. 2, we observe students were capable of seeing real-world applications of programming through related-to-other-courses questions, which helps us answer the first research question with 78% of students strongly agreeing or agreeing.

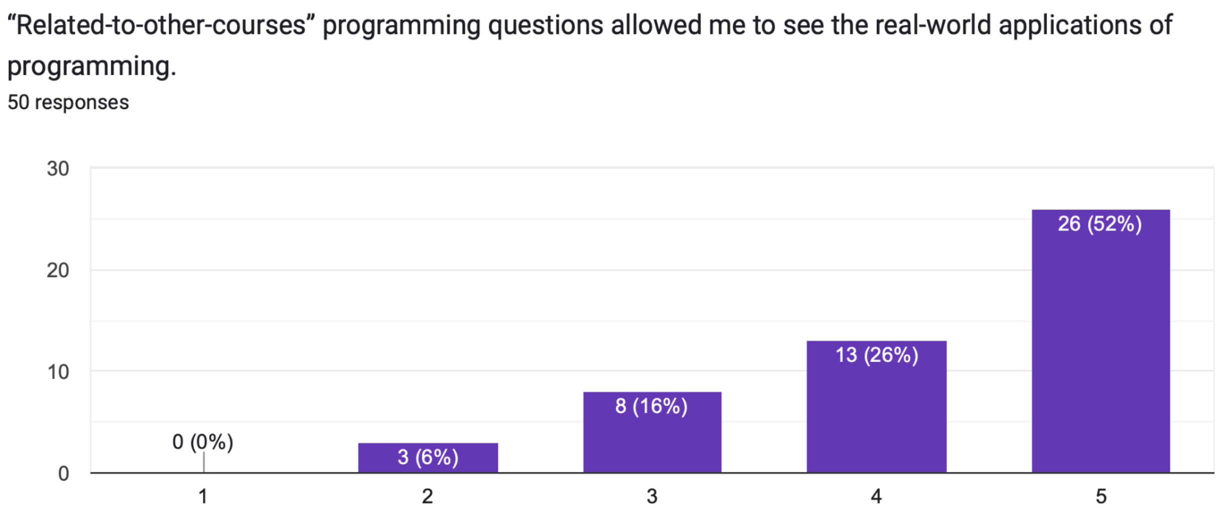


Fig. 2. Were students able to see real-world applications of programming in related-to-other-courses questions?

In Fig. 3, we observe students having different opinions on where related-to-other-courses questions reinforce their understanding of other engineering courses. This helps us answer the second research question that 74% of students are agreeing or neutral, which shows that there is at least no harm done to their understanding of other engineering courses. From the challenges that students shared, a student mentioned that related-to-other-courses questions brought the challenges they have with other courses into a programming course. This made them afraid that they may not score high on these programming questions for reasons unrelated to their programming skills. This may explain the reason for 26% disagreeing with the statement.

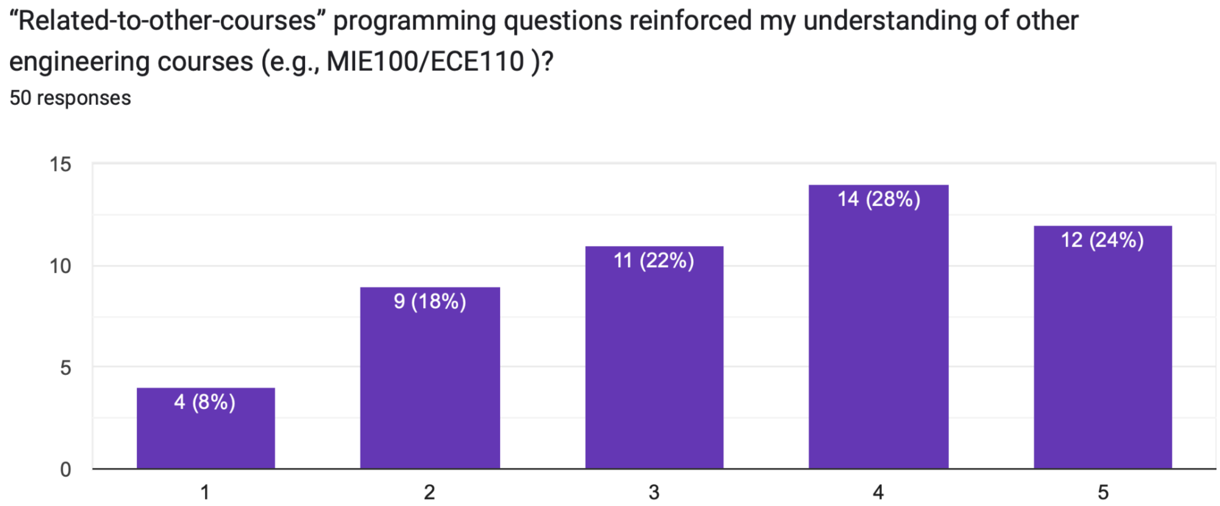
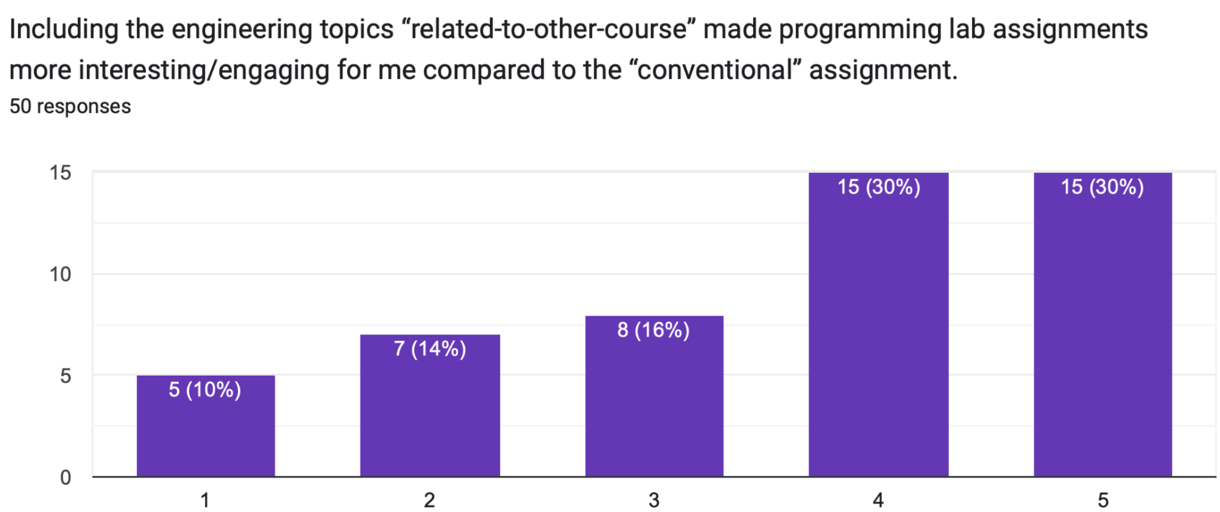


Fig. 3. Were students’ knowledge of other courses getting enforced.

In Fig. 4, we observe 60% of students find related-to-other-courses more engaging and interesting than conventional questions and 16% are neutral. This helps us answer our third research question.



In the open-ended questions where students were asked if there is something they would like us to continue or start doing, many students mentioned that they appreciate their inclusion and were interested in having problems from Calculus included as well. Multiple students commented that the related-to-other-courses questions were not focusing on difficult or challenging concepts. This was put in mind as we were developing the questions ensuring we give the students the equations needed to solve the problem. For example, a student mentioned "I believe that the "related to other courses" are more interesting than conventional ones. I like the fact that they are related, but **do not require difficult skills that would shift the attention away from coding.**" However, there were other students who thought the engineering concepts were simplistic: "Perhaps consider choosing more difficult problems because the students are mostly familiar with the basics, so creating a program that only calculates resistors in series was a little easy. Regardless, lab9 did teach me a lot about linked lists in the end, so not really much of a complaint if I understood the topic."

Some of the comments mentioned that the related-to-other-courses made them feel like an engineer, for example, a student mentioned: "I really enjoyed the "related-to-other-courses" parts for the labs. I think having more of them would be a great idea. **It really makes you feel like an engineer**, when you are able to use knowledge from different courses to code software." Also, the same student appreciates the relation to real-world applications in the problem as they said: " When coding things for MIE 100 and ECE 110, **it is very easy to pretend that you're doing something cool like coding for a spaceship software or a cool robot**. As well, it gives us a taste of what future projects might look like (I've seen upper year hardware/software design courses). So in the future, I think other students would appreciate having more "related-to-other-courses" parts or even full labs."

However, some students in the open-ended questions about challenges mentioned that they dropped the dynamics and electric fundamentals courses or struggled with these two other courses, which made them feel they are at a disadvantage to other students due to reasons unrelated to programming. In the future, we may mention in lab assignments that since the problem is explained in the lab handout, we don't believe if someone is not taking the other courses, they won't feel at a disadvantage.

Conclusion

Our study highlights the significant benefits of integrating interdisciplinary concepts into programming courses for engineering students. By connecting foundational programming skills with real-world applications drawn from Dynamics and Electrical Fundamentals, we observed increased student engagement, improved perceptions of the relevance of programming, and enhanced problem-solving skills. The majority of students appreciated the interdisciplinary approach, with many noting that it deepened their understanding of real-world programming applications and provided a glimpse into the complexities of upper-year projects.

Qualitative feedback further emphasized the value of these assignments, as students expressed how they felt more like engineers applying knowledge across disciplines. However, the results also revealed important considerations for future implementations. While many students found the assignments engaging and appropriately challenging, some suggested incorporating more complex problems to better align with their growing expertise. Conversely, others who struggled with Dynamics or Electrical Fundamentals felt at a disadvantage, raising the need for clearer communication that the assignments are designed to be accessible regardless of prior coursework.

In response to this feedback, future iterations of the labs will aim to balance complexity with accessibility, ensuring all students can engage meaningfully with the material. Additionally, expanding the interdisciplinary scope to include concepts from Calculus and other core engineering courses could further enhance the learning experience and broaden the appeal of these assignments. Overall, our findings showed the value of interdisciplinary integration in early engineering education and provide a framework for designing programming assignments that prepare students for the multifaceted challenges of their future careers. In addition, this paper's findings have implications for curriculum designers, instructors, and administrators who seek to cultivate a more cohesive and applied engineering education experience. For future work, we aim to extend this approach to another second-year programming course and include more advanced problems as suggested by the students.

References

[1] T. A. Litzinger, L. R. Lattuca, R. G. Hadgraft, and W. C. Newstetter, "Engineering education and the development of expertise," *Journal of Engineering Education*, vol. 100, no. 1, pp. 123–150, 2011. doi: 10.1002/j.2168-9830.2011.tb00006.x.

[2] A. H. Espera and N. P. Pitterson, "Exploring Students' Learning of Electric Circuits in Real-world Context," presented at the 2021 ASEE Virtual Annual Conference Content Access, Virtual Conference, Jul. 2021. doi: 10.18260/1-2--37148.

[3] Y. Zhang, X. Zhang, T. Li, and M. Ye, "Design of Embedded Interdisciplinary Educational Program: A Case Study-based on an AI Certification Program," presented at the 2024 ASEE Annual Conference & Exposition, Portland, Oregon, Jun. 2024. doi: 10.18260/1-2--47127.

[4] R. Lavi, C. Cong, Y. Lai, J. A. Lavallee, G. L. Long, N. Melenbrink, and A. B. Mitra, "The Evolution of an Interdisciplinary Case-Based Learning First-Year Course," presented at the 2023 ASEE Annual Conference & Exposition, Baltimore, Maryland, Jun. 2023. doi: 10.18260/1-2--44451.

[5] P. Xu, W. Zhang, and W. Zhang, "Influence of Interdisciplinary Teaching System on Interdisciplinary Competence of Engineering Graduate Students: Analysis of Mediating Effects of Interdisciplinary Identity," presented at the 2023 ASEE Annual Conference & Exposition, Baltimore, Maryland, Jun. 2023. doi: 10.18260/1-2--43692.

[6] M. Borrego, J. E. Froyd, and T. S. Hall, "Diffusion of engineering education innovations: A survey of awareness and adoption rates in U.S. engineering departments," *Journal of Engineering Education*, vol. 99, no. 3, pp. 185–207, 2010. doi: 10.1002/j.2168-9830.2010.tb01056.x.

[7] A. Schulz, C. Shriver, B. Seleb, C. Greiner, D. Hu, R. Moore, M. Zhang, N. Jadali, and A. Patka, "A Foundational Design Experience in Conservation Technology: A Multi-Disciplinary Approach to Meeting Sustainable Development Goals," presented at the 2022 ASEE Annual Conference & Exposition, Minneapolis, MN, Aug. 2022. doi: 10.18260/1-2--40887.

[8] J. M. Baker, K. Chandrashekhara, R. H. Hall, and S. E. Watkins, "Interdisciplinary Learning through a Connected Classroom," *International Journal of Engineering Education*, vol. 20, no. 6, pp. 1167–1176, 2004.

[9] J. Strobel and A. van Barneveld, "When is PBL more effective? A meta-synthesis of meta-analyses comparing PBL to conventional classrooms," *Interdisciplinary Journal of Problem-Based Learning*, vol. 3, no. 1, pp. 44–58, 2009. doi: 10.7771/1541-5015.1046.

[10] M. J. Prince and R. M. Felder, "Inductive teaching and learning methods: Definitions, comparisons, and research bases," *Journal of Engineering Education*, vol. 95, no. 2, pp. 123–138, 2006. doi: 10.1002/j.2168-9830.2006.tb00884.x.