Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #48087

Reducing the DFW Rate for Engineering Majors in Introductory Computer Science Through Contextualized Learning and Peer-Supported Engagement

Muhammad Naveed Aman, University of Nebraska - Lincoln
Moomal Bukhari, University of Nebraska - Lincoln

Moomal Bukhari received the B.Sc. degree in Electrical Engineering and M. Sc. degree in Electrical Engineering from the National University of Computer and Emerging Sciences, Islamabad, Pakistan in 2013 and 2017, respectively. She worked as a Lecturer at National University of Computer and Emerging Sciences. She is currently doing her PhD in Computer Engineering from University of Nebraska-Lincoln. Her research interests include cyber-physical systems, and cybersecurity.

Eric Clarence Einspahr, University of Nebraska - Lincoln

Eric Einspahr is the Director of Academic Navigators at the University of Nebraska - Lincoln. He has 16 years of experience in higher education student affairs. The areas of student affairs Eric has worked in include academic advising, STEM student retention in the College of Engineering, and data driven/proactive outreach and interventions for undergraduate students. During this time Eric has developed in interest in using data to identify students in need of early interventions and to examine the student academic experience to ensure student success. Eric's work has a special focus on applying the Ecology of Validation to student experiences in higher education and working to close institutional equity gaps. Eric holds a Bachelors Degree in English and a Masters Degree in Higher Education Administration with an Emphasis in Student Affairs - both from the the University of Nebraska - Lincoln.

Dr. Jena Shafai Asgarpoor, University of Nebraska - Lincoln

Dr. Jena Asgarpoor is a Professor of Practice at the University of Nebraska – Lincoln and the Director for the Master of Engineering Management Program in the College of Engineering. She received her Ph. D. and M.S. in Industrial Engineering, specializing in Engineering Management, from Texas A&M University in College Station where she had previously earned a B.A. in Political Science (Summa Cum Laude). Prior to UNL, she was a professor at Bellevue University for 26 years, where in 1994 as part of her teaching portfolio she developed and taught the first fully online asynchronous web-based course offered by that institution. Her interests lie in engineering management, quality management, pedagogy, and assessment of teaching and learning, particularly in the online space. She is active in the American Society for Engineering Management (ASEM) and the American Society for Engineering Education (ASEE). In recent years she has President of the Council of Engineering Management Academic Leaders (CEMAL) and Program Chair and Chair of the Engineering Management Division (EMD) of ASEE. Dr. Asgarpoor is currently serving as President of the American Society for Engineering Management (ASEM).

Dr. Bhuvaneswari Gopal, University of Nebraska - Lincoln

Dr. Bhuvaneswari (Bhuvana) Gopal is an Assistant Professor of Practice in the School of Computing at the University of Nebraska-Lincoln where she teaches Software Engineering, Software Security in Practice, the industry internship course, and leads the Learning Assistant Program that serves several computing courses at the School of Computing. Dr. Gopal has extensive experience in the software industry, where she spent 14 years in various roles, including Software Architect and Lead software engineer positions before switching to full time academia. She is also a Microsoft certified professional, with an MCPD certification. Her research work is focused on evidence-based, active learning pedagogies to improve software engineering education. She has published papers specifically on Peer Instruction and Process Oriented Guided Inquiry-based Learning-like pedagogical approaches in the undergraduate software engineering classroom, both in person and virtual. She holds a Bachelors degree in Physics from the University of Madras, Chennai, India. She holds two Masters degrees - one in Computer Science from the University of Nebraska-Lincoln, and one in Medical Physics from Anna University, Chennai, India. She holds a Ph.D. in Computer Science from the University of Nebraska-Lincoln.

Reducing the DFW Rate for Engineering Majors in Introductory Computer Science Through Contextualized Learning and Peer-Supported Engagement

Abstract

In this paper, we examine the efficacy of two major approaches implemented to redesign the Computer Science I course for non-computing engineering majors, with the primary aim of reducing the DFW (Drop, Fail, and Withdrawal) rate. As computing skills have become indispensable in 21st-century engineering, the lack of an engineering-focused curriculum in introductory Computer Science courses often results in suboptimal learning outcomes and high DFW rates. Our course redesign tackled these issues by integrating two key strategies: contextualized learning and the inclusion of undergraduate learning assistants (LAs) to foster a peer-supported learning environment.

The first approach, contextualized learning, embeds computing concepts within real-world engineering problems. By presenting engineering-focused challenges, students were able to bridge the gap between theoretical knowledge and practical application. This approach not only fostered deeper comprehension but also significantly increased engagement, helping students to connect computing principles to their future careers. As a result, we observed improvements in both retention and academic performance.

The second approach involved the integration of undergraduate learning assistants in weekly labs. These LAs, who were often seen by students as peers, provided invaluable support in helping students navigate more complex problem-solving tasks. The peer connection fostered by the presence of LAs created a more open and approachable learning environment. Students felt more comfortable seeking help and asking questions, which encouraged active participation in labs and discussions and reduced hesitation in engaging with challenging material and promoted collaborative problem-solving, further enhancing the learning experience.

We applied these two strategies in tandem by presenting students with engineering problems, which were solved through in-class discussions and instructor guidance. During weekly labs, LAs were available to assist students with more advanced problems, reinforcing key concepts introduced in class. In addition to labs and discussions, the course included a semester-long project in which students identified a problem relevant to their major, developed a computational solution, and produced a professional report that explained the engineering design phases. This project not only reinforced technical skills but also developed students' ability to communicate and document their work in a professional setting. To further support student learning, auto-graded homework assignments were utilized to provide real-time feedback, ensuring continuous reinforcement of the material.

We analyzed the grades of over 200 students across two semesters to assess the impact of the redesign. In addition to academic performance data, we conducted a post-course survey to evaluate students' learning expectations, outcomes, and perceptions of the course. The survey responses were categorized based on student majors: engineering, non-engineering, and undecided. One particularly noteworthy finding was the positive impact the redesign had on students in the undecided category, who were typically freshmen unsure of their major and more likely to withdraw from the class. This group, historically at higher risk of dropping the course, showed significant improvement in retention and engagement following the redesign.

Overall, the course redesign yielded substantial benefits for all student categories, resulting in a marked reduction in the DFW rate—from over 30% in traditional iterations of the course to just 9%. Additionally, feedback from instructors of subsequent courses that depended on the foundational knowledge provided by Computer Science I was overwhelmingly positive, affirming the effectiveness of our redesign approach. These results suggest that the combination of contextualized learning and peer-supported engagement via undergraduate TAs can serve as a powerful strategy for improving student outcomes in introductory computing courses, particularly for non-computing engineering majors.

This work demonstrates the success of employing targeted, pedagogically sound strategies to create a more engaging, supportive, and relevant learning environment, ultimately reducing the DFW rate and better preparing students for future coursework and professional challenges.

Keywords: Faculty paper, Contextualized Learning, Learning Assistants, Introduction to Computer Science, non-Computing majors, DFW rate, Peer-led learning.

1 Introduction

It is now essential for engineering students to acquire strong programming skills early in their academic careers due to the quick integration of computing skills into engineering specialties. However, the special requirements and viewpoints of non-computing engineering majors are sometimes overlooked in conventional introductory computer science courses. Disengagement, poor learning outcomes, and a high rate of drop, fail, and withdrawal (DFW) might result from this imbalance. A major reason is a lack of perceived significance. Because they might not perceive the direct application of learning programming or algorithmic problem-solving to their field, engineering students may become less intrinsically motivated. According to research, students are less likely to remain involved in a course if they don't see how it applies to their everyday lives, which increases the likelihood of dropout and failure [2]. Innovative instructional strategies that increase programming's accessibility, relevance, and support for engineering students are needed to address these issues.

The difficulty of grasping fundamental programming principles, which frequently call for abstract reasoning and iterative problem-solving, is another contributing factor. Frustration and a decline in confidence may arise from these cognitive demands as well as a lack of prior computing experience [7]. This problem is exacerbated when students believe that their teachers or peers are not supporting them, which can make them feel alone, especially if they are women or underrepresented minorities in STEM programs. The incompatibility of conventional evaluation techniques with the learning preferences of non-computer science majors, such engineering students, is another element that leads to increased DFW rates [9]. These students may find it challenging to understand basic programming ideas as a result of this mismatch, which could result in further performance and retention issues. Furthermore, classes that don't use contextual examples or active learning techniques might not be able to create a welcoming and stimulating environment. Prior research has demonstrated that peer-led support networks and cooperative learning settings can enhance student performance and lower failure rates in beginning STEM courses [13].

Understanding the unique obstacles that non-computing majors confront in computing courses is critical for developing effective interventions. Strategies that have demonstrated potential in raising student motivation and lowering DFW rates include incorporating active learning, connecting content to engineering applications, and creating a sense of community through mentorship and peer learning [5]. This study expands on these findings to investigate the efficacy of recent pedagogical modifications in an introductory computing course and determine whether they contributed to the large decline in DFW rates in recent semesters.

In order to lower the DFW rate among non-computing engineering majors, this study examines a thorough redesign of the Computer Science I course. Two strategies served as the basis for this redesign: contextualized learning and peer-supported engagement. Contextualized learning incorporated programming concepts into real-world engineering scenarios, using domain-specific examples from mechanical engineering and biological systems engineering to show the practical relevance of computing skills. This approach sought to improve understanding and engagement by bridging the gap between abstract programming principles and their applications in engineering. By including undergraduate learning assistants (LAs) in weekly lab sessions, peer-supported engagement was accomplished. By serving as relatable and approachable mentors, these LAs fostered a collaborative, engaging, and open-minded learning environment. Combining these techniques allowed the course to handle the different difficulties experienced by students with different backgrounds while also offering a more individualized and encouraging learning environment.

This study looks at how these tactics affect student engagement, retention, and performance. The study offers proof of the efficacy of these pedagogical improvements by examining survey responses, course grades, DFW rates, and feedback from later teachers over a three-year period. According to this research, focused, contextually appropriate, and peer-supported teaching strategies can greatly enhance learning results and better equip students for upcoming academic and career obstacles.

The remainder of the paper is organized as follows. Section II describes the related literature. Section III represents the methodology. Section IV describes our work in detail. In Section V we present the results of our study, and Section VI concludes this paper.

2 Related Literature

In this study the course redesign was done by integrating two key strategies: contextualized learning and the inclusion of undergraduate learning assistants (LAs) to foster a peer-supported learning environment. In this section we discuss prior literature pertaining to both these topics.

Contextual learning was used in a prior study that emphasized practical applications pertinent to the area of mechanical engineering by including conservation education into an internship preparation course [16]. This strategy sought to enhance student learning outcomes and promote conservation-based behaviors, emphasizing the influence of discipline-specific, useful content on behavior and student engagement. In another study contextual learning was used by the Children Designing & Engineering (CD&E) Project, which incorporated design-and-make activities into K–5 lesson plans that linked science, math, and technology to real-world scenarios modeled after New Jersey companies [9]. The research demonstrated the value of experiential, context-rich learning opportunities for young pupils by combining aspects of design and technology education with problem-based learning to promote early exposure to skills and attitudes relevant to the workplace. Contextual teaching and learning (CTL) principles were used to improve student competency in an Indonesian study on applied physics education [4]. The study found that while modeling and reflection might be made simpler because of the nature of the subject, important CTL elements including constructivism, inquiry, questioning, learning communities, and genuine assessment were crucial for Applied Physics. In another research, the "Teaching Circle for Large Engineering Courses" combines contextual learning by adjusting teaching tactics to large-class issues and encouraging reflection on real-world classroom experiences, so promoting the implementation of student-centered practices [1]. The Engineering Grand Challenge Framework (EGCF) improves contextual learning by embedding engineering principles in societal challenges, enhancing student motivation and engagement, particularly in upper-level courses, and improving outcomes for real-world applications and lifelong learning [8]. According to a study on contextual learning in engineering education, problem-solving skills require an awareness of social, cultural, and environmental contexts [12]. The study proved that contextual competency is an essential and quantifiable element of successful engineering education by creating a trustworthy four-item scale (Cronbach's alpha = 0.91). In another research the University of Michigan ASEE Student Chapter emphasizes the value of pedagogical practices by supporting professional development, improving classroom conditions, and encouraging collaboration among students, teachers, and administrators [15]. Their efforts have helped engineering education adopt a culture of constant development.

A peer learning initiative spanning three undergraduate courses at a Canadian university was the subject of a study on Peer-Assisted Learning (PAL) in recreational education [6]. The study emphasized the advantages of peer learning and determined the variables influencing its efficacy using information from focus groups, reflection tasks, and surveys. The results highlight how peer-to-peer instruction can improve learning outcomes and student engagement. In another study on LAs in STEM education examined classroom interactions between a LA and a student in a Calculus I course to better understand how LAs improve student results [3]. Through an analysis of their conversation's social and informative components, the study demonstrated how LAs create connections of support and skillfully frame information. The results offer guidance to strengthen the effectiveness of LA programs and their influence on student learning. A study of formative assessment in undergraduate life sciences courses found that Undergraduate Teaching and Learning Assistants (UTLAs) provide useful input on course logistics, instructional materials, and student comprehension [10]. In order to promote student learning, the results show how UTLA feedback facilitates formative assessment and strengthens teacher-peer tutor teamwork. The authors in the study [11] emphasizes the importance of peer-engagement learning by describing how peer mentors in a makerspace-based engineering course provided emotional, technical, and collaborative assistance to help students feel more connected and work together. According to the results, peer mentors strengthened students' technical proficiency and self-esteem while also strengthening their ties to the engineering community. Another study highlights the value of peer-engagement learning by adopting collaborative, peer-led workshops to build a sense of belonging among first- and second-year engineering students [14]. According to the results, these workshops, which were conducted by qualified facilitators utilizing student-centered, active learning techniques, enhanced students' sense of mentorship, connection, and respect in the larger college community. Women and underrepresented minorities in STEM fields benefited most from these workshops.

3 Methododology

3.1 Contextual Learning

Contextualized learning is a pedagogical approach that integrates subject matter with practical, domain-specific applications, making the learning process more relevant and engaging for students. In our redesigned Computer Science I course for non-computing engineering majors, this approach was pivotal in improving both comprehension and engagement. By embedding programming concepts within the framework of engineering problems, students were able to see the direct application of computational skills to their field of study. For example, tasks such as writing code to model stress distribution in mechanical components or simulate thermal conductivity introduced students to fundamental programming constructs while simultaneously deepening their understanding of mechanical engineering principles.

The gap between abstract computing concepts and practical engineering applications was closed by this domain-specific contextualization. It inspired students to view programming as a crucial tool for problem-solving in their professional careers rather than just as a technical need. These examples' applicability encouraged intrinsic motivation, which enhanced the significance and satisfaction of learning.

3.2 Peer-Supported Engagement Through Learning Assistants

The incorporation of undergraduate learning assistants (LAs) to promote a peer-supported learning environment was the second pillar of the course redesign. As former students of the course, LAs were crucial in establishing a connection between the students and the teaching staff. Their participation in weekly lab sessions gave students a relatable and approachable support structure, lowering obstacles that frequently keep students from asking for assistance. LAs helped students solve engineering-based computational issues, debug code, and comprehend difficult programming principles. Students were more at ease participating in conversations, asking questions, and seeking clarification since they were able to communicate in a peer-like manner. LAs fostered a collaborative environment that promoted peer-to-peer interactions and active involvement, both of which are essential for deep learning. Additionally, students gained confidence in handling difficult material thanks to the informal mentoring given by LAs, which also promoted a feeling of community and belonging in the classroom. Peer support had a particularly significant effect during labs, when students worked on challenging problem-solving exercises that reaffirmed the ideas covered in lectures. In addition to helping students gain technical proficiency, LAs fostered important soft skills like teamwork, communication, and resilience by creating a cooperative and encouraging learning environment. Peer-supported engagement can improve learning outcomes in introductory computer courses for non-computing majors, as seen by the overall decrease in the DFW rate in this study.

4 Work

For this study, faculty members from the School of Computing (SoC), referred to as the serving department, and faculty from the departments whose students were enrolled in the course, referred to as the served departments, were consulted. To initiate the redesign process, it was essential to begin by reviewing the current course content and understanding the expectations of the served departments regarding what the course should teach their students.

Faculty members from the SoC who were teaching CSCE 155N were asked to provide a detailed list of topics covered in the course, along with the depth of coverage for each topic. The depth of coverage was categorized into three levels:

- Mastery: Students are expected to have a comprehensive understanding and demonstrate independent application of the concept. Topics included were Programming concepts, including branches, loops, functions, and matrix operations.
- Familiarity: Students encounter the topic multiple times and begin to practice it. Topics related to problem identification, problem formulation, and the mathematical modeling of engineering systems—such as linear systems, graphs and networks, ordinary differential equations, and optimization problems were included in this category.
- Exposure: Students are introduced to the topic and develop a basic understanding. Topics in this category included data types, optimization algorithms, and discrete-time simulations.

The served departments evaluated the topics and assigned depth-of-coverage levels based on their needs. This collaborative approach ensured alignment between the course content and the practical needs of students from diverse academic backgrounds, facilitating a more effective and targeted redesign.

The three primary served departments for this course include Mechanical and Materials Engineering (MME), Biological Systems Engineering (BSE), and Chemical Engineering (Chem). Each department provided specific expectations regarding the depth of topic coverage required for their students, which was instrumental in aligning the course design with their needs and desired outcomes.

For BSE, topics categorized at the mastery level included matrix operations, data graphing, and scripting to store commands. Topics such as variable naming conventions, Boolean logic, debugging, and data manipulation were identified at the familiarity level, while developing new functions, identifying and formulating problems, and using Simulink were classified at the exposure level.

Similarly, MME emphasized mastery-level understanding of fundamental programming concepts, such as computing basics, variables, branches, loops, functions, engineering design, problem identification, and problem formulation. Topics such as linear systems, graphs and networks, and differential equations were placed at the familiarity level, while more advanced topics, including optimization, data fitting, and introductory Simulink in MATLAB, were categorized at the exposure level.

Understanding these specific requirements allowed the course to be redesigned to meet the expectations of these departments effectively. The resulting curriculum ensured students acquired the necessary foundational skills while being introduced to more advanced topics that aligned with the interdisciplinary demands of their respective fields.

The teaching faculty of CSCE 155N revamped the course outline in Spring 2022 to align with the requirements of the three served departments. Two major types of changes were implemented: curriculum-related and pedagogy-related adjustments.

The curriculum changes focused on integrating essential engineering concepts, including motivation, problem-solving, linear systems, graphs and networks, ordinary differential equations (ODEs), optimization, and Simulink. Prior to Spring 2022, the course lacked motivational content, particularly regarding the relevance of computing in engineering. To address this, the first week of the semester was dedicated to emphasizing the importance of computing in 21st-century engineering. In terms of problem-solving, the pre-Spring 2022 course content focused solely on programming constructs. This was expanded to include the engineering design process, emphasizing problem identification, formulation, and solving—critical skills for engineering students. Key topics introduced for contextual learning included:

- Linear Systems: Over 75% of mathematical problems in scientific or industrial applications involve solving a linear system. Lectures were designed to teach students how to model engineering problems as linear systems and solve them using MATLAB. Previously, the course only covered basic techniques for solving sets of linear equations.

- Graphs and Networks: As essential tools for modeling biological systems, route planning, and transportation networks, graphs and networks were introduced to address their application in solving diverse engineering problems.

- Ordinary Differential Equations (ODEs): Recognized as fundamental in Chemical Engineering, ODEs are crucial for understanding process dynamics. They also play a significant role in modeling mechanical systems and changes in biological models.

- Optimization: Many complex engineering problems are formulated as optimization problems, making this topic indispensable for the course. Simulink: Simulink, MATLAB's graphical programming environment, was introduced to model, simulate, and analyze dynamic systems. As many engineering problems involve discrete event systems, students were provided with an introduction to Simulink.

The pedagogical changes focused on transforming the way programming was taught and practiced. The adjustments included:

- Programming as a Tool: Shifting the focus from programming as an abstract concept to programming as a tool for solving real-world engineering problems.
- In-Class Tutorials: Incorporating hands-on tutorials during class to enhance student engagement and understanding.
- Homework and Projects: Designing assignments and projects that were directly relevant to students' engineering disciplines, ensuring they could see the practical applications of the skills they were learning.
- Engineering Problems: Embedding engineering-specific problems into the course content to provide context and relevance for the topics covered.

In Fall 2022, several curriculum and pedagogy-related changes were made to CSCE 155N, building on findings from the Spring 2022 semester. These changes were based on student feedback and aimed to enhance learning outcomes while maintaining consistency and alignment with course objectives.

Curriculum-related changes included:

- Emphasis on Programming Constructs: In Spring 2022, students highlighted the need for more time spent on programming constructs. To address this, approximately four weeks of lectures in Fall 2022 were dedicated to programming constructs. Consequently, engineering design examples, previously covered in the first month, were postponed until after programming constructs were taught.
- Synchronization of Labs with Lectures: Prior to Fall 2022, labs were conducted using zyBooks, which lacked synchronization with lecture content. In Fall 2022, labs were redesigned to align closely with lecture topics and reinforce contextual learning. Students were required to complete labs in person during designated times, with LAs providing real-time feedback. Feedback from LAs regarding the difficulty and time requirements of lab tasks was collected and used to improve lab design for subsequent semesters.

Pedagogy-related changes in Fall 2022 included:

- In-Class Programming Practice: Based on Spring 2022 feedback, in-class coding exercises were introduced alongside tutorials. To support this, two Learning Assistants (LAs) were present in each lecture to assist students, and the overall number of LAs was increased to eight due to the larger class size.
- Use of Auto-Grading Platforms: Homework assignments were moved to the MathWorks grader platform, allowing for consistent grading and providing students with real-time feedback. This change addressed prior inconsistencies in grading and improved the turnaround time for feedback.
- Introduction of In-Class Quizzes: Quizzes were added to encourage participation and engagement, a feature that was previously absent.

- Active Monitoring and Individual Support: The instructor actively monitored student progress and identified those lagging behind. These students were invited for one-on-one meetings, where tailored plans were devised to help them succeed. Opportunities to make up missed assessments were expanded, allowing students to miss up to two homeworks, quizzes, labs, or tutorials without penalty. This change reduced stress and improved student morale.
- Standardization of Course Projects: Previously, students selected their project topics, leading to grading inconsistencies. In Fall 2022, students were given a predefined list of five project topics across various domains relevant to their majors. This approach improved grading consistency and clarified expectations.

The SoC faculty who taught this course in Spring 2022 and Fall 2022 was replaced by another faculty member for Spring 2023 and Fall 2023. The course was then reassigned to the original faculty member for Spring 2024 and Fall 2024. Based on student feedback gathered from Spring 2023 through Spring 2024, further improvements were implemented before the Spring 2024 semester. These enhancements included:

- Allocating additional time to programming constructs and introducing more in-class programming exercises to reinforce these key concepts. Lecture slides were revised to incorporate examples that were directly relevant to the students' majors, making the material more engaging and relatable. Additionally, new practice exercises were developed to help students master programming constructs more effectively. These changes led to a significant improvement in student comprehension of programming constructs, evident from a marked increase in student performance on both homework assignments and lab tasks.
- Introduced clicker questions, which significantly enhanced student engagement and enabled real-time formative assessment. These questions also allowed the instructor to adapt the teaching pace to match the students' learning speed, ensuring a more effective and responsive learning experience.

5 Results

Figure 1 illustrates the DFW rates for this class over 14 semesters, starting from Spring 2018. During the period from Spring 2018 to Fall 2019, the DFW rate consistently exceeded 30%. However, a noticeable reduction in DFW rates began in Spring 2020. This reduction, observed through Fall 2021, was achieved by significantly narrowing the course content. Instructors focused on foundational programming topics, such as variables and branching, while omitting more advanced concepts like loops and functions. While this approach successfully lowered the DFW rates, it compromised the course's rigor, rendering it inadequate as a prerequisite for subsequent courses, such as the object-oriented programming class. Consequently, the course was dropped as a prerequisite for these advanced classes.

To address this issue, the course underwent a redesign starting in Spring 2022. The objective of the redesign was to maintain the reduced DFW rates while ensuring that students met all essential learning objectives and were adequately prepared for advanced coursework. The redesigned course was implemented and monitored over three years, from Spring 2022 to Fall 2024, with feedback collected from students, instructors teaching subsequent courses, and DFW rate trends.

Figure 1 highlights the impact of the new strategy, which incorporated contextual learning and peer support, on the DFW rates. From Spring 2022 to Fall 2024, the DFW rates remained consistent, ranging from 8% to 13%, despite being taught by different instructors. During this period, the mean DFW rate was 10.76%, with a standard deviation of 1.47%. In contrast, prior to Spring 2022, the mean DFW rate was 28.03%, with a standard deviation of 7.43%. The significant reduction in the mean DFW rate reflects an overall improvement in student performance, while the lower standard deviation demonstrates greater consistency across semesters, indicating that outcomes were less influenced by variations in teaching styles. Together, these metrics validate the effectiveness of the structural and pedagogical revisions, contributing to more stable and positive student outcomes.

The notable difference in mean values and the corresponding reduction in variability underscore the success of the course redesign. The integration of standardized contextual learning, thorough documentation of instructional strategies, and support from learning assistants ensured that these improvements were sustained regardless of the instructor. These findings affirm that the course modifications effectively enhanced and maintained student achievement over time.

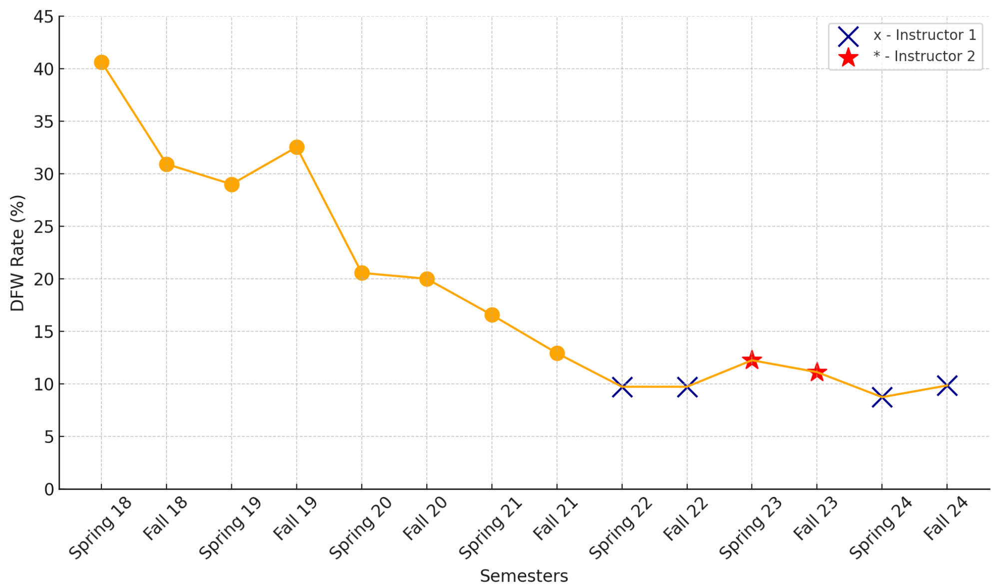

Figure 1: DFW rates for CSCE 155N.

The results also highlight the importance of partnership between the SoC faculty and the faculty of the served units. This ensures alignment of the contents covered with the desired objectives and helps in addressing any gaps in the curriculum. These discussions have helped shape a strategy for increasing student performance and improving learning outcomes in basic courses.

In Fall 2024 the students were asked to take a survey to assess the effectiveness of the changes made to the course CSCE 155N. The questions are given below.

- Q1: How well do you think the course activities and examples related to real-world problems in your field of study? Note: Think about whether the problems, examples, or scenarios used in this course matched the kinds of challenges you might encounter in your career or other classes related to your major.
- Q2: Did the real-world examples and applications in this course help you better understand the concepts?Note: Consider whether the examples made it easier for you to see how the concepts can be applied outside of the classroom. Did they make the material clearer or more meaningful.
- Q3: How helpful and engaging did you find the support provided by the Undergraduate Teaching Assistants (UTAs) during the course?Note: Reflect on your interactions with the LAs and compare them to any experiences you've had with GTAs in other courses. Was it easier to communicate with LAs because they are peers? Did their guidance feel more relatable or relevant to your learning experience?
- Q4: How would you rate your understanding of the course material at the beginning of the class compared to now?Note: Think about how confident or knowledgeable you felt about the topics when the course started versus how you feel now. Did the course help you improve your understanding, and if so, by how much? Reflect on specific skills or concepts you feel you have gained mastery over.
- Q5: How effective was the course in helping you apply what you learned to practical or real-world scenarios?Note: Think about whether you could connect what you learned in this course to solving problems or handling situations outside the classroom. Did the course prepare you to apply your knowledge in practical ways?

The students were provided with five options to select from for each of the above questions. The options and their corresponding score are given below.

- Very well: 5
- Somewhat well: 4
- Neutral: 3
- Not very well: 2
- Not at all: 1

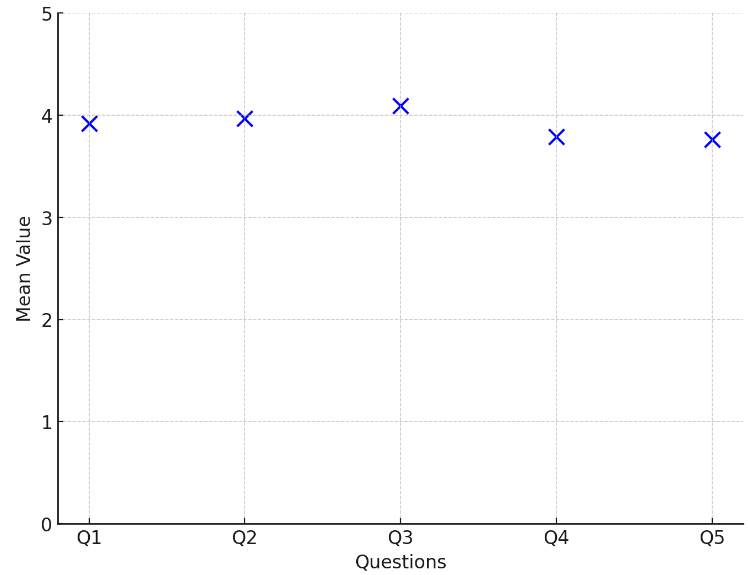

Figure 2: Mean values of survey responses for Q1 to Q5.

The results of the survey are given in Figure 2. The plot shows the mean score of the answers selected by the students. With the majority of replies falling between "Somewhat Well" and "Very Well" on the feedback scale, the mean score for the questions ranges from 3.76 to 4.09, suggesting that students' perceptions are typically favorable. With little difference in satisfaction levels across several course components, this range indicates that most students thought the course material, instructional strategies, and overall learning experience were efficient and pertinent.

6 Future Recommendations

While the proposed strategy to improve learning outcomes while maintaining a low DFW rate has been effective, some faced challenges in specific areas, including navigating nested loops to create multi-dimensional matrices with special conditions and formulating engineering problems in ways that can be solved using computing. To address these challenges, several targeted interventions are recommended:

- An additional lab session will be introduced to help students become more comfortable with MATLAB Grader. This session will feature hands-on activities designed to familiarize students with the platform's features and functionality. It will also include guided tutorials that provide step-by-step walkthroughs of common tasks, helping to build confidence and reinforce key concepts. Additionally, troubleshooting tips will offer focused guidance on resolving common errors and maximizing the platform's utility, ensuring students can effectively navigate and utilize MATLAB Grader for their assignments.
- Additional lecture and lab time will be allocated to cover nested loops and their application in creating and manipulating two-dimensional matrices. The instruction will begin with step-by-step examples, starting from basic concepts and gradually progressing to more

complex scenarios. To enhance understanding and engagement, real-world applications will be incorporated, encouraging students to solve engineering problems involving matrices. This approach aims to make the content both relatable and practical, bridging theoretical concepts with practical engineering use cases.

- Lab activities will be revised to incorporate AI tools that assist students in a variety of tasks. These include debugging code to efficiently identify and resolve errors, generating automated test cases to validate their solutions, and exploring alternative approaches through AI-powered suggestions, which encourage independent learning. By integrating these tools, the labs aim to enhance technical skill development while promoting the ethical and effective use of AI in engineering problem-solving.
- To improve consistency and efficiency in project evaluation, auto-grading mechanisms will be integrated for coding tasks with defined outputs or algorithm correctness, utilizing platforms like MATLAB Grader or custom scripts. Manual grading will be reserved for open-ended components, allowing instructors to provide detailed feedback on creativity, innovation, and problem-solving approaches. This balanced approach ensures both objective evaluation of technical tasks and meaningful assessment of higher-order thinking skills.

7 Conclusion

This project's main goal was to investigate and record the adjustments made by the School of Computing (SOC) that resulted in a notable decrease in DFW rates, which are grades of D, F, or course dropouts. Making ensuring that these effective tactics could be maintained and improved upon for upcoming semesters was the aim. This strategy's evaluation of faculty assignments for 100-level courses was a crucial component. The incorporation of contextual learning strategies was a key component of the course enhancement. The goal of contextual learning is to give students opportunities to solve problems and experience real-world situations that relate to their experiences and future careers. For this, contextualized examples that were pertinent to the units on BSE and MME were included. Students were better able to understand difficult material and identify how it related to their future occupations when programming ideas were presented through real-world applications that were pertinent to these fields. In addition to increasing engagement, this tactic strengthened important learning objectives.

In addition to contextual learning, LAs played an important role in increasing student outcomes. Throughout lectures and labs, LAs were on hand to help students with coding tasks, answer questions, and provide one-on-one assistance as needed. Their presence promoted an engaging and cooperative learning environment by bridging the gap between instructor-led explanations and individual student comprehension. In order to make sure that the lab activities' degree of difficulty and completion time matched the students' learning speed, the LAs also provided insightful input during their planning and execution. Additionally, the course material was examined to see if it sufficiently prepares students for later courses. Whether the course material satisfied the learning requirements of students in relevant programs was the main focus of this evaluation. The results show the significant decline in DFW rates. The DFW rates remained consistent between 8% to 13% with a mean of 10.76% and standard deviation of 1.47%, which are significantly less than the DFW rates in the previous semesters.

References

[1] Olivia S Anderson and Cynthia Finelli. A faculty learning community to improve teaching practices in large engineering courses: Lasting impacts. In *2014 ASEE Annual Conference & Exposition*, pages 24–46, 2014.

[2] Russell Benford and Julie Gess-Newsome. Factors affecting student academic success in gateway courses at northern arizona university. *Online Submission*, 2006.

[3] Rebecca Butler and Orly Buchbinder. Learning assistant-student interaction in calculus: A critical discourse analysis. *North American Chapter Int. Group Psychol. Math. Educ.*, 2023.

[4] Fadhilah Fadhilah, Z Mawardi Effendi, and Ridwan Ridwan. Analysis of contextual teaching and learning (ctl) in the course of applied physics at the mining engineering department. In *International Journal of Science and Applied Science: Conference Series*, volume 1, pages 25–32, 2017.

[5] Scott Freeman, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. Active learning increases student performance in science, engineering, and mathematics. *Proc. Nat. Acad. Sci.*, 111(23):8410–8415, 2014.

[6] Barbara-Ann Hamilton-Hinch, Christie Stilwell, Cassandra Manuel, Susan Hutchinson, Kimberley Woodford, and Allison Ellis. Peer assisted learning: What can students teach us and each other? *SCHOLE: J. Leisure Stud. Recreation Educ.*, 38(3):161–174, 2023.

[7] Emma Hogan, Ruoxuan Li, and Adalbert Gerald Soosai Raj. Cs0 vs. cs1: Understanding fears and confidence amongst non-majors in introductory cs courses. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 25–31, 2023.

[8] Lisa Huettel, Michael R Gustafson, Joseph C Nadeau, David Schaad, Michael M Barger, and Lisa Linnenbrink-Garcia. A grand challenge-based framework for contextual learning in engineering: Impact on student outcomes and motivation. In *2015 ASEE Annual Conference & Exposition*, pages 26–48, 2015.

[9] Patricia Hutchinson. Children designing & engineering: Contextual learning units in primary design and technology. *J. Ind. Teacher Educ.*, 39(3):122–145, 2002.

[10] Hannah Jardine and Gavin Frome. Undergraduate teaching and learning assistants provide feedback as part of a formative assessment system. *J. College Sci. Teaching*, pages 1–9, 2024.

[11] Louis S Nadelson, Pamela L Dickrell, and Katherine DeJesus. Mentoring for making: Peer mentors working with learners in a making-focused engineering course. In *2024 ASEE Annual Conference & Exposition*, 2024.

[12] Hyun Kyoung Ro, Lisa R Lattuca, Dan Merson, and Patrick T Terenzini. Measuring engineering students' contextual competence. In *2012 ASEE Annual Conference & Exposition*, pages 25–920, 2012.

[13] Ibukunoluwa E Salami, Segun S Oladipo, and Logan A Perry. Exploring high dfw rates in an engineering statics course: Insights from faculty and teaching assistants. In *2024 ASEE Midwest Section Conference*, 2024.

[14] Ryan Sauve, Celia Evans, and Lisa Schneider-Bentley. Work in progress: Peer led collaborative courses develop a sense of belonging and community for all undergraduate engineering students. In *2023 ASEE Annual Conference & Exposition*, 2023.

[15] Sara Soderstrom, Chris Lorenz, and Michael Keinath. Influence of the asee student chapter on college wide pedagogical issues. In *2001 Annual Conference*, pages 6–582, 2001.

[16] Yeri Sutopo et al. The formation of conservation-based behaviour of mechanical engineering students through contextual learning approach. *Int. J. Environ. and Sci. Educ.*, 12(4):617–627, 2017.