Traditional Lecture Format vs. Active Teaching Format in an Online Freshman Engineering Course

Dr. Nina Kamath Telang, University of Texas at Austin

Dr. Nina Telang is an associate professor of instruction in the ECE department at UT Austin. She has taught a variety of courses in the ECE department at the freshman, sophomore and junior undergraduate levels that include 4 required (core) courses, and 2 elective (tech-area) courses. Her repertoire of courses is from a range of areas such as circuit theory, digital logic design, solid state devices, computing systems, and embedded systems. Her teaching style fosters an active learning classroom environment where student involvement is highly encouraged. Instructional tools based in technology are heavily used in the classroom to aid the learning process for all students, to strengthen student-faculty interaction, and to improve student engagement. She is passionately involved in supporting the success of at-risk students through the development of the general engineering course and supplemental instruction sessions for introductory ECE courses.

Miss Nisha Abraham, University of Texas at Austin

Nisha Abraham coordinates the Supplemental Instruction program. She received her B.S. in Cell and Molecular biology from The University of Texas at Austin in 2007, her M.S. in Biology from Texas A&M University in 2012 and her M.A. in STEM Education from The University of Texas at Austin in 2019. Additionally, she has over five years of combined industry and science research experience, has worked as a senior bioscience associate at UT's Austin Technology Incubator, and has served as an adjunct faculty member in biology for South University. She was a teaching assistant for several undergraduate biology classes, created TA training modules for the Center for Teaching Excellence, and conducted research on improving student motivation and performance in science education. Additionally, Nisha has over five years of combined industry and science research experience, has worked recently as a senior bioscience associate at UT's Austin Technology Incubator, and has served as an adjunct faculty member in biology for South University.

Mohana Seelan, University of Texas at Austin

Mohana Seelan obtained her B.S. in Electrical and Computer Engineering at the University of Texas at Austin in Spring 2021. During her time at UT, she served as a STEM tutor for the Sanger Learning Center, as an undergraduate teaching assistant for EE306 (Intro to Computing) and EE109K (Enhancing Academic Success), and as a first-year mentor for Ramshorn students. Mohana is interested in computer science and engineering education, specifically in first-year student learning. She is now starting her M.S. at the University of California, San Diego in Computer Science Engineering.

Mr. Ramakrishna Sai Annaluru, University of Texas at Austin

Ramakrishna (Sai) Annaluru is a 3rd Year MS/PhD student in Electrical and Computer Engineering at the University of Texas at Austin, researching at the intersection of machine learning and signal processing. Sai's educational background include 1 semester of graduate Teaching Assistant experience for Signals and Systems and Introduction to Computing, 2 semesters of head instructor experience for a 1 credit hour spatial visualization course, and 2 years of undergraduate tutoring experience in introductory electrical engineering and mathematics classes.

Traditional lecture format vs. Active teaching format in an online freshman engineering course

Abstract

This complete paper analyzes the impact of active classroom learning on student performance in an introductory computing course. This course is a freshman level, foundational course in the Computer Engineering track in our engineering program. Student performance and overall experience in this course can influence their perception of the major, and therefore affect retention rates. A significant number of our freshman students enter our program with some computer programming experience through high school level courses in Java programming. Evidence gathered through a survey conducted at the start of the semester indicated that more than 50% of our incoming students have either completed AP Computer Science A or AP Computer Science Principles or both, about 25% have taken another computer programming course or learned programming through high school club activities, and only about 20% of our students have no experience whatsoever. While this freshman level course does not require any prior knowledge of programming basics, students having some background are at an advantage due to their familiarity with the process of algorithmic thinking, and translation of an algorithm to a computer program. The primary objective of this research project was to determine the impact of active learning on students with different levels of programming experience, and to test whether this teaching style would help level the playing field.

Introduction

Active-learning is by no means a novel style of teaching and has been heavily researched in higher education. In his meta-analysis of active learning research, Prince defines active learning generally as any "instructional method that engages students in the learning process" [1]. He differentiates traditional study practice, such as homework, from active learning by pointing out that active learning occurs during lecture and is juxtaposed to a traditional, passive lecture format. Research done by Di Vesta and Ruhl has shown the positive effects of the active learning strategies like the pause procedure, on short-term and long-term retention [2, 3]. Learning gains associated with active learning in introductory physics courses have been shown by Laws et al. and Hake [4, 5]. In the past two decades, more active learning research has been conducted in science and math courses such as calculus, statistics, physics, and computer science[6-11], and several studies by Felder, who explores the effect of active learning in introductory chemical engineering courses [12].

I. Motivation for Study

The uniqueness of the test course: Introduction to Computer Engineering

While there are few active learning studies that take place in computer science classrooms, very minimal studies exist within engineering courses that integrate both electronic hardware and computer programming components [13-15]. The test course that we chose for this research study was a first year required course that all our incoming engineering students in our department are required to take in their very first semester. A bottom-up approach to computing is followed, with topics such number formats, arithmetic and logic operations, digital logic circuits, and a basic computer model introduced before presenting machine and assembly language programming and debugging. Due to the wide range of topics covered in some depth in this first year course, it is historically considered as difficult. A student's performance in this course can therefore dictate their decision to stay within the major.

Retention within engineering

Studies have shown that out of the 40%-60% of engineering students dropping out of their major [16-18], 85% dropped out within their first two years, and 15% in the last two to four years [16]. The reasons for attrition usually have to do with a lack of calculus and physics readiness, pressure from the transition to college, teaching techniques in first-year courses, and community building within engineering [16,19]. Several studies have shown that average first year GPAs are higher for students who stay in engineering [20, 21]. In fact, most students with GPAs of 2.0 were generally placed on academic probation and eventually left the major [20,22]. Research also shows that first semester and first year GPAs affected long term retention 2-3 years later [20, 23].

Retention within engineering is clearly a problem, with success and performance in first year engineering courses playing a large role in the decision to stay or continue within the major [20]. If active learning can improve one or more factors in a student's decision to stay within engineering, then the potential to impact retention rates is also high. Since GPA within a first-year class is a significant metric for student success and retention [20-22], exploring a method that might directly impact and improve GPA is highly important.

Diversity in student population

Enrollment to our engineering school is diverse, with diversity stemming not only from ethnicity and gender, but also from the level of preparation for our rigorous engineering program. A large fraction of our incoming class has some computer programming experience from high school coursework and/or practical experience from extra-curricular club activities. More notably, there is also a significant fraction of our student body that does not have any experience. Year after year this discrepancy in prior exposure to computer programming poses a challenge for our instructors. Holden et al. showed in a sequence of intro to programming courses that students with prior programming experience completed the first course a full letter grade higher [24]. Wilcox's study also demonstrated that grades were significantly better and retention rates were higher after a first computer science course for students with previous programming experience than those without experience [25].

Given this diversity in the incoming engineering student class, there was more opportunity to explore the effects of active learning and study its impact on student performance using student preparation as a parameter.

Uneven access to prior programming education

Research done by Holden and Hagan shows students with prior programming experience, including at least one programming language, perform better in introductory computer science college courses than their peers [26, 27]. Both studies examined student performance in classes that introduced object-oriented concepts, however, and it is important to note that the computing test course in this study does not. How students gained this experience varied between self-learning online programs, high school computer science classes, and extracurricular clubs.

Though prior programming experience can greatly affect student performance in an introductory class such as ours, access to learning resources where students can gain that experience is uneven. The 2018 State of CS Education, which collected data on 42% of all public K-12 schools and 67% of public high schools in the U.S. showed that schools in rural communities, schools with higher percentages of underrepresented minority students, and schools with high percentages of students receiving free and reduced lunch are less likely to teach computer science to their students [28].

According to Google's Diversity Gaps in Computer Science, underrepresented groups face many social and structural barriers to gaining and pursuing any sort of programming education in high school [29]. Black students are less likely than white students to have classes dedicated to CS at the school they attend (47% vs. 58%). Black (58%) and Hispanic (50%) students are less likely than White (68%) students to use a computer at home. Female high school students are less likely to have learned CS during high school (50% vs 59%), are less likely to have learned CS online (31% vs 45%), and are less likely to have learned CS concepts on their own outside of class (41% vs 54%).

While our study does not analyze demographic data, it is important to note that underrepresented groups have faced many more barriers to receive prior programming experience, a variable which was explored in our study. The ability to increase performance in groups across all prior programming experiences, especially groups with low prior programming experiences is an important step to increasing the graduation rate of underrepresented groups within computing majors.

Online modality of teaching

The coronavirus pandemic that hit the globe in 2020, required all our first year engineering courses in our school to be offered online. Prior to the fall 2020 semester, none of our first year courses had online offerings, and none of the instructors teaching first year engineering courses had any experience with online teaching, or incorporating active learning components into the course curriculum. This added another dimension to our investigation of the impact of active learning in our introductory computer engineering course. It offered more opportunities to develop new active learning course materials, and contribute to the research in this area.

Despite the existing studies in this field of engineering education (referred to in the previous section) which indicated the positive influence of active learning on student performance, the uniqueness of our course content, the diversity of the student body, and the online classroom environment meant we could not assume that active learning would have a positive impact on student learning in our context. This motivated us to design and implement this research study.

II. Limitations of Study

One limitation of this study was the lack of experience of the authors in teaching online, and therefore implementing active learning in an online environment. The challenges of teaching online are real, and these challenges are compounded by the fact that the test course was a first year course, where students were unfamiliar with expectations and standards. Prior to the semester when this study was conducted, the authors were accustomed to the excitement and energy of college classrooms, where we would try to learn the names and faces of all the students during the first week of the semester, and get the students excited about engineering. In an online setting, both instructors and students have to learn new ways of communicating and find alternatives to the social cues and body language that normally guide us in our interactions. The very descriptor, ***active***, seems to suggest that there is some level of movement, energy, and dynamism. As Darby states, "Communication strategies may not be top of your mind when teaching online, but they should be" [30]. Therefore, while our intentions and efforts to create active learning activities were sound, the implementation may not have had the desired effect.

Another limitation was due to a lack of statistical data from which to draw conclusions. Determining correlations between exposure to active learning experiences and student performance using only one semester worth of data is difficult. Additionally, the control and treatment sections of this course had different enrollment numbers, with the treatment section having a 30% lower enrollment compared to the control section.

As will be described in a later section of this paper, the active learning opportunities we offered to our students were: 1) in the classroom during lectures, and 2) in optional Supplemental Instruction (SI) sessions, which were offered outside of class, where teaching assistants led small group problem solving sessions in an online setting. The self-selection bias arising due to the optional nature of the SI sessions is another limitation of this research study that could not be controlled for.

III. Research Questions

To assess the impact of active learning on our first year engineering students, this report addresses the following questions:

1. How does active learning during the lecture affect student performance in different aspects of the course, such as programming assignments, homework assignments, weekly quizzes, and final course GPA?
2. In what way does experience with active learning impact students with different levels of prior programming?
3. How does attendance in the Supplemental Instruction (SI) sessions affect student performance differently in different aspects of the course, such as programming assignments, homework assignments, weekly quizzes, and final course GPA, and in what way does it influence students with different levels of prior experience?

IV. Design and Implementation

Study of the impact of active learning in the classroom

A cooperative learning approach where students work together in groups during class time was the method that was followed in the treatment section of this course, while the traditional lecture format was followed in the control section of this course. The study design featured the following controls to ensure consistency and avoid bias: 1) the same instructor and teaching assistants in both sections, 2) students randomly assigned to the two sections, 3) student access to the same course materials including slides, handouts, videos, homework and programming assignments, and exams.

The active teaching approach provided an inclusive environment to promote effective teamwork skills. Groups consisting of 4-5 students were assigned in the first week of the semester using data collected from a survey that was conducted on the first day of class. Survey questions ranged from questions about gender and race/ethnicity, to educational experience in computer programming and other engineering concepts. The groups were created taking into account the diversity of student prior educational experience.

One of the major concerns that instructors have about active teaching, or any instructional method other than the traditional lecture format, is regarding content coverage. Thus, it is extremely important to carefully design classroom activities to ensure the completion of the entire curriculum for this course. Much of the planning work for this project centered on the design of the class activities. Since the course content spanned a wide spread of topics such as basic computing concepts, logic circuits, computer microarchitecture, and assembly programming, it was imperative that we incorporated different types of activities to suit different content.

Measuring the effectiveness of an instructional technique is problematic, since the technique can have different levels of impact on different course components. All course assessments for the two sections of the course were identical. This facilitated a fair comparison of student performance in all aspects of the course, such as homework assignments, programming assignments, weekly quizzes, final exam, and overall course GPA.

Given the evidence that prior programming experience has impact on performance and retention, we designed our study to account for prior programming experience's effect on performance instead of the applied instructional active learning method. This was done by sectioning the students into categories of varying programming experience during our data analysis phase of this research project.

The following were the types of active learning exercises that were implemented in the treatment section of the course:

1. Breakout room problem-solving sessions, where pre-assigned groups were each assigned a different problem related to the topics discussed in the pre-lecture videos, reading assignments, and during the lecture. Before opening the room, the students were given clear instructions on the activity, including the role of the *recorder* of the group. Each breakout session was timed, and students were given sufficient time to brainstorm solutions. The teaching assistants and instructor wandered from room to room offering help when needed, and providing feedback as necessary. After returning back to the main room, the *recorder* of each group shared the solution to their problem. Students were also instructed to synchronously edit and answer group-specific problems on a classroom Google Doc.
2. Zoom polling served as a very effective way to get a read on the class, while also promoting student engagement. Aggregated poll results were shown to the students after the answer to a problem was revealed.
3. Playing games like Kahoot also helped to foster a more lively and attentive classroom culture.

In contrast, the instructor solved all problems during the lectures in the control section of the course. In both sections, students were encouraged to ask questions both verbally and through the chat feature. The teaching assistants and instructor would answer questions posed in the chat window.

Study of the impact of active learning in (optional) Supplemental Instruction sessions
The historically successful and evidence-based Supplemental Instruction (SI) program created in 1973 at the University of Missouri in Kansas, is an academic support program that is recognized for its track record of improving student performance in historically difficult courses, thus resulting in improving student retention [31]. This program has been implemented at our university in the engineering school, and the authors have studied the impact of this program on student performance in a number of engineering courses [32]. While the results of these studies all point to improved student performance and lower failing rates, transitioning to the online format posed some challenges and questions about the effectiveness of this program, which relies on active learning exercises, in a virtual setting.

Traditionally, the SI program consists of weekly sessions led by undergraduate upper-class students who are recruited and trained in the SI methodology, which consists of active and collaborative learning strategies to engage students. The SI Leaders go through pre-service training, followed by weekly training and development meetings (about one hour a week). These meetings provide the SI Leaders with ongoing practice of facilitation skills, SI strategies, discussion of pedagogy, and continuous feedback. Regular observations are conducted by the SI Coordinator, who is a learning specialist associated with our campus learning center, and Leaders conduct one peer observation per semester. The SI Leaders are responsible for collecting attendance at each session and administering programmatic interventions throughout the semester.

One of the important components of the SI model has been voluntary attendance. While we did not intend to stray too far from this ideal, the fall semester of 2020 presented a unique challenge due to the online delivery of all learning programs. The research collaborators in this study were most concerned about the impact of reduced community building experiences and relatedness on student motivation. For this reason, a cohort-based model was used where students interested in participating in the SI program signed-up for these sessions in the first week of the semester, with the understanding that they were highly encouraged to attend each week for the entire semester. This ensured the voluntary nature of student participation, while also establishing a peer group for students which could serve to enhance their sense of connectedness to their classmates.

V. Methodology
In this study we used a quantitative approach to analyze data collected from student performance in different assessments from the control and treatment sections.
The following quantitative data were collected:

1. Each category in this analysis was determined based on what students reported at the start of the semester. We consider the student to have (i) *Advanced Placement (AP) experience* if they took either AP Computer Science A or AP Computer Science principles in high school, (ii) *Some programming experience* if they either took a non-AP Computer Science course or had extracurricular experience, (iii) The rest of the students were considered to have *no prior programming experience*.

2. Student performance in homework assignments, programming assignments, weekly quizzes, and final course scores.
3. SI Attendance: The number of minutes of SI session attendance throughout the semester.

VI. Findings

Impact of active learning in the classroom

In Table 1 we have listed the number of students in each category for the control and treatment sections separately, followed by the percentages shown in the bar chart in Figure 1. The size of each category in this analysis was determined based on what students reported in a survey conducted at the start of the semester.

	AP Experience	**Some experience**	**No experience**
Control Section	52	28	18
Treatment Section	44	16	12
Total Students	96	44	30

Table 1: Number of students in each prior experience category for each section of the course.

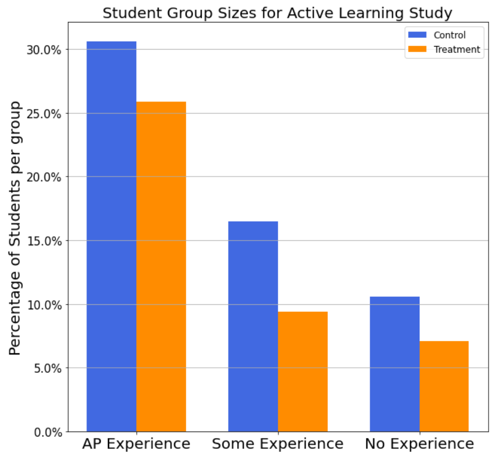

Figure 1: Percentages of each category of students, showing comparable numbers in each section.

We evaluated student performance based on their scores in weekly quizzes, programming labs, homework assignments, and final overall grades. As we can see in Table 2, the active learning format (treatment) had the strongest impact on students with no experience. For these students, we observe better grades in each assignment category. Most notably, there is a 7% improvement in the programming labs scores of students in the treatment section of the course. In fact, these students scored higher than their counterparts in the control section on every lab assignment as seen in Figure 2. The labs were all in assembly language, with labs increasing in difficulty from

Labs 1 through 6. Labs 2, 3, and 4 were based on sorting arrays and collecting statistics, while Labs 5 and 6 were based on linked lists. Lab 7 was an interrupts based lab, where students were given starter code that they needed to edit. This lab was intentionally created in this manner due to the lack of time at the end of the semester. As seen from Figure 2, the difference in lab scores for the more challenging labs was more significant.

			Quizzes		Labs		Homeworks		Final Grades	
		N	Mean	SD	Mean	SD	Mean	SD	Mean	SD
AP Experience	Control	52	88.54	8.38	95.29	5.30	93.67	6.73	92.14	5.25
	Treatment	44	88.76	8.88	93.28	13.29	90.94	9.70	91.17	8.79
Some Experience	Control	28	85.47	7.93	93.36	8.20	91.84	9.25	89.48	5.96
	Treatment	16	84.72	9.19	92.00	9.74	91.03	6.76	89.04	7.07
No Experience	Control	18	83.63	11.42	87.91	23.17	91.47	8.60	86.76	12.96
	Treatment	12	85.65	9.32	94.33	7.00	93.01	6.63	90.07	6.44

Table 2: Student mean scores (out of 100) and standard deviation (SD) arranged by prior programming experience and class. Students with more programming experience performed better in the course than those without. The active learning lectures had the highest impact on students with no prior experience.

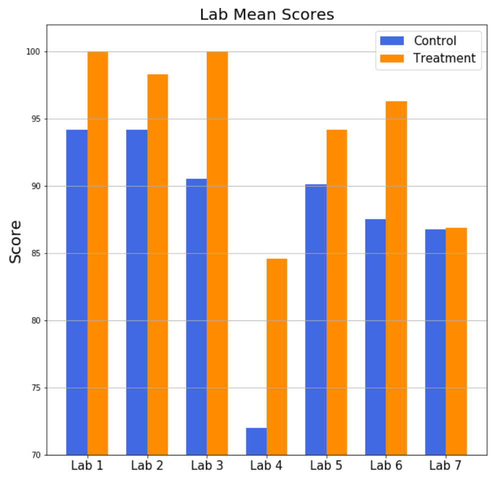

Figure 2: Lab mean scores between treatment and control sections for students with no prior experience, clearly showing positive impact of active learning.

Impact of active learning in (optional) Supplemental Instruction sessions

As previously mentioned, students signed up for SI sessions on a voluntary basis, and stayed with their cohorts for the entire semester. However, there were students who signed up but did not attend every week. Because of the online nature of the course, we were able to observe student attendance down to the minute. There were 97/170 students who showed up to at least one SI session at some point in the semester.

We categorized a student as *SI* if their online attendance was higher than 200 minutes. This corresponded to attendance in at least 4 sessions worth of material throughout the semester, giving us 77 *SI* students and 93 *no SI* students. We observed their performance on various assignments and have summarized our findings in Table 3. Here we can see that *SI* students performed better on all assignment categories relative to *no SI* students. The most significant impact of SI is on those students with no prior programming experience. We observe a statistically significant high positive correlation between SI attendance and final grade outcome using the independent samples t-test [33]. When participants are randomly allocated to one of two groups and mean performance based on an outcome measure is used, an independent samples t-test is appropriate to determine statistical significance.

We performed a two tailed t-test with equal variances on the means of scores in different assignments (quizzes, labs, homeworks) and the final averaged scores for the "No SI" and "SI" groups within each category of previous programming experience. The standard p-value indicating significance was considered <0.05 and if so, the null hypothesis (that any difference in the average scores was by chance) would be rejected. While the "SI" groups outperformed the "No-SI" groups in all three categories, Table 3 shows a statistically significant higher final grade for the "SI group" with no previous programming experience vs the similar "No-SI" group.

			Quizzes		Labs		Homeworks		Final Grades	
		N	Mean	SD	Mean	SD	Mean	SD	Mean	SD
AP Experience	No SI	52	88.24	9.01	93.44	9.99	90.67[2]	9.74	91.03	7.61
	SI	44	89.11	8.09	95.47	9.57	94.49[2]	5.60	92.48	6.50
Some Experience	No SI	29	83.36[1]	9.01	92.20	9.13	90.99	7.27	88.11	7.00
	SI	15	88.74[1]	5.40	94.15	7.95	92.63	10.34	91.67	4.37
No Experience	No SI	13	81.35	11.39	83.22	26.56	86.98[3]	9.18	83.63[4]	14.29
	SI	17	86.80	9.44	96.03	4.26	95.99[3]	3.03	91.49[4]	5.72

Table 3: Student mean scores (out of 100) and standard deviation (SD) arranged by prior programming experience and SI attendance. We observe a correlation between students attending SI and performance in the class in all categories. [1]$p=0.0403$ [2]$p=0.0238$ [3]$p=0.000712$ [4]$p=0.0479$

To illustrate the impact of the extent of SI attendance on final course grades, we show scatterplots in Figures 4, 5, 6, 7. Each plot indicates an improvement in performance with more SI attendance.

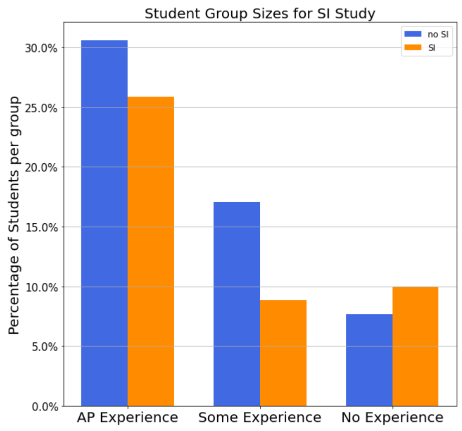

Figure 3: Percentages of each category of students, showing comparable numbers in the SI and no SI groups.

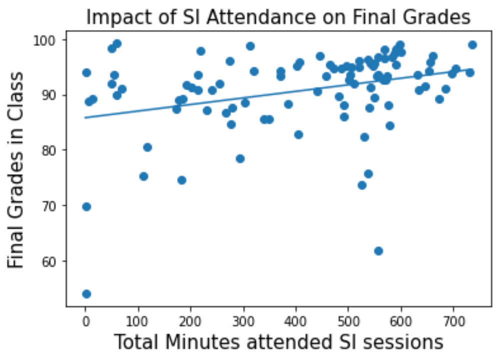

Figure 4: Impact of SI Attendance on end of semester course grades for all students.

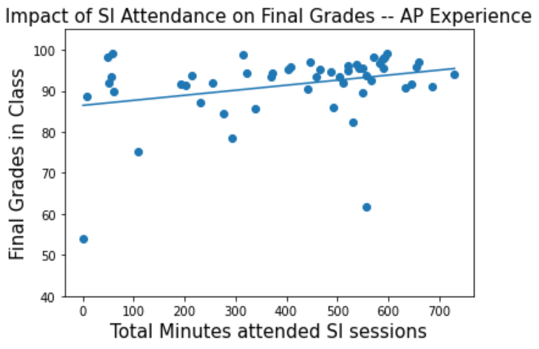

Figure 5: Impact of SI Attendance on end of semester course grades for students with AP programming experience

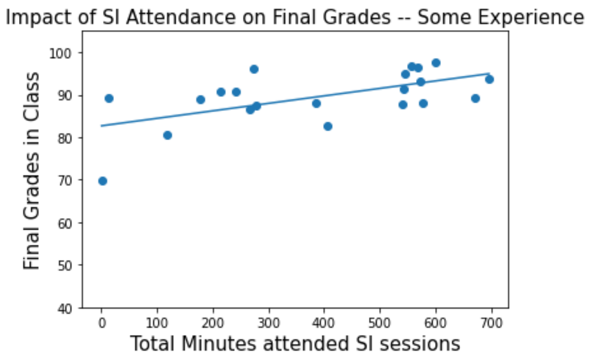

Figure 6: Impact of SI Attendance on end of semester course grades for students with some programming experience.

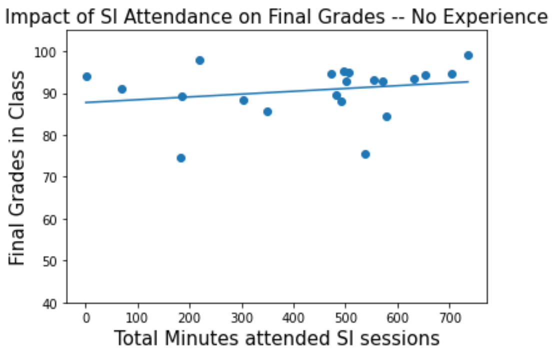

Figure 7: Impact of SI Attendance on end of semester course grades for students with no programming experience.

Our analysis thus far indicates that active learning experiences (whether in the classroom or in optional SI sessions) benefits students with no prior programming experience the most. A combination of participating in the SI program and being part of the treatment group may have resulted in a bigger improvement in scores. To separate the effects of the two methods, we examined the scores of the three categories of students, who chose not to participate in the SI program. These are listed in Table 4. While the final grades of the students with computer programming experience did not differ significantly, there was close to a 9% (a whole letter grade) improvement in final grades for those students with no experience in the treatment section.

			Quizzes		Labs		Homeworks		Final Grades	
		N	Mean	SD	Mean	SD	Mean	SD	Mean	SD
AP Experience	Control	26	86.97	9.61	94.09	6.05	92.32	8.10	90.95	6.12
	Treatment	26	89.52	8.36	92.79	12.89	89.03	11.05	91.11	8.81
Some Experience	Control	18	83.13	8.33	92.90	7.75	91.65	7.71	88.21	6.32
	Treatment	11	83.74	10.45	91.05	11.36	89.90	6.68	87.94	7.92
No Experience	Control	7	79.83	12.27	76.59	35.16	85.62	11.14	80.34	17.99
	Treatment	6	83.11	11.11	90.95	8.71	88.57	6.90	87.46	7.58

Table 4: Student mean scores (out of 100) and standard deviations (SD) arranged by prior programming experience and active learning treatment vs control for those students who chose not to attend SI sessions.

VII. Discussion and Summary

In their book chapter, Riegle-Crumb et al. make the argument that many common explanations of gender and racial disparities in STEM outcomes do not hold water, when national data shows that women perform at similar or better rates in STEM courses in high school than their male counterparts and racial minorities perform poorly in comparison to their white counterparts, until you control for socioeconomic status/wealth [34]. Their findings demonstrate that gender and race, as social constructs, are highly tied to social class and wealth, which can impact standardized testing, how school contexts shape different forms of inequity, the STEM pipeline and eventual college matriculation and persistence. To connect this to our use of prior programming experience to evaluate the impact of active learning techniques, this chapter reports, "the bulk of the research on course-taking disparities strongly implicates within-school sorting processes, such that Black and Hispanic youth are less likely to be enrolled in advanced courses compared to their White peers" (pg. 140). Additionally, Burke et. al, [35] provided important evidence for how implementing active learning as an instructional technique has a positive impact on all student learning gains, but significantly improves underrepresented minorities' (URM's) course achievements in STEM courses. In their research, they cite numerous articles that present strong evidence for the need for integrating active learning during course design to ensure URM's retention and progression through STEM pathways.

This research team found itself, like many other institutions and instructors, at the crossroads of online learning environments, social and educational inequities and historically difficult course content, with all the difficulties and opportunities that these components afford. This unique course taught online for the first time, with a depth and breadth of programming content, can be challenging for all students but can especially halt underrepresented students progress through their engineering coursework and ultimately prevent them from achieving success in engineering. In an already challenging semester -- a pandemic which caused university closure and completely online instruction -- our concerns for students who are already disadvantaged by the K-12 educational system and their eventual outcomes helped drive our research questions and approach to implementing active learning techniques within this freshmen computing course.

The research team implemented two types of active learning interventions; one was within the lecture, with one section of students working on activities such as breakout room problem solving time, low-stakes polls and active games; the other was voluntary attendance in SI sessions that provided out-of-lecture collaborative problem solving practice and active peer learning experiences. In both active learning interventions, the treatment group performed better on assignments compared to the control group. The impact of our two active learning methods was most prominent on students with no computer programming experience. Because of the introductory nature of this course, it is likely that students with prior experience had some exposure to the content and experience with algorithmic thinking. As detailed above, the disparities in educational experiences that create outcome inequities were reduced by a combination of active learning techniques in this course, which we believe will serve these students in being successful in a course where they otherwise might not have been, as well as in the long term.

Along with our quantitative findings, our in-class observations yielded the following advantages and disadvantages of implementing the online format in the context of active learning, that may be helpful for other instructors hoping to utilize such a format:

Advantage of an online format:

1. Students could easily collaborate on Zoom whiteboards for problems.
2. Students had easy access to all group materials.
3. The chat feature during class was a highly-effective academic tool. Students could ask a question to the class and teaching assistants, or other students, could answer the question in real-time. Students who felt uncomfortable messaging the general chat also had the flexibility to privately ask teaching assistants questions. The general chat could be quite social at times, which a student made an engineering joke, or when students shared how they were struggling.
4. Teaching assistants were able to engage with a variety of students, frequently, within the active learning treatment section.

Disadvantages of an online format:

1. Giving instructions and helping students find the breakout room problems took up to 5-10 minutes of class time near the beginning of the semester. After a month of instruction, preparation time decreased.
2. Groups that were inactive were undetectable until a teaching assistant entered a room.
3. Technical difficulties and internet problems took up valuable classroom time.
4. Group-work relied heavily on attendance of specific students, since groups were pre-assigned and remained the same throughout the semester. Some groups only had one student who would consistently attend, so that student would often have to do problems alone, or with a TA.

The following were our observations of student interactions within groups:

1. Some groups were particularly lively, while others were not. The groups that seemed to be more active and engaged in the problems, and in talking to each other, may have tended to be more consistent with attendance and effort.
2. Students often talked about subjects outside the course. Many discussed their experiences in other freshman classes, their quarantine habits, and other social aspects of their freshman experience. This may have enhanced their level of connection to their peers and therefore their level of motivation.

References

[1] Prince, M. (2004), Does Active Learning Work? A Review of the Research. Journal of Engineering Education, 93: 223-231.
[2] Ruhl, K. L., Hughes, C. A., & Schloss, P. J. (1987). Using the Pause Procedure to Enhance Lecture Recall. Teacher Education and Special Education, 10(1), 14–18.
[3] Francis J. Di Vesta, Deborah A. Smith, The pausing principle: Increasing the efficiency of memory for ongoing events. Contemporary Educational Psychology, Volume 4, Issue 3, 1979, Pages 288-296, ISSN 0361-476X.
[4] Laws, P., D. Sokoloff, and R. Thornton, "Promoting Active Learning Using the Results of Physics Education Research," UniServe Science News, Vol. 13, July 1999.
[5] Hake, Richard. (1998). Interactive-Engagement Versus Traditional Methods: A Six-Thousand-Student Survey of Mechanics Test Data for Introductory Physics Courses. American Journal of Physics - AMER J PHYS. 66.
[6] Springer, Leonard & Stanne, Mary & Donovan, Sam. (1999). Effects Of Small-Group Learning On Undergraduates In Science, Mathematics, Engineering, And Technology: A Meta-Analysis. Review of Educational Research. 69. 21-51.
[7] Carlson, K.A., & Winquist, J.R. (2011). Evaluating an Active Learning Approach to Teaching Introductory Statistics: A Classroom Workbook Approach. *Journal of Statistics Education, 19*.
[8] Kvam, Paul. (2000). The Effect of Active Learning Methods on Student Retention in Engineering Statistics. American Statistician 54. 136-140.
[9] F. S. Tsai et al., "From Boxes to bees: Active learning in freshmen calculus," 2013 IEEE Global Engineering Education Conference (EDUCON), Berlin, Germany, 2013, pp. 59-68
[10] Meltzer, D., & Thornton, R. (2012, June 1). Resource Letter ALIP–1: Active-Learning Instruction in Physics. *Am. J. Phys., 80*(6), 478-496. Retrieved March 7, 2021
[11] Scherr, R., & Hammer, D. (2008). Student Behavior and Epistemological Framing: Examples from Collaborative Active-Learning Activities in Physics. *Cognition and Instruction, 27*, 147 - 174.
[12] Felder, R.M., Felder, G.N. and Dietz, E.J. (1998), A Longitudinal Study of Engineering Student Performance and Retention. V. Comparisons with Traditionally‐Taught Students. Journal of Engineering Education, 87: 469-480.
[13] Pirker, Johanna & Riffnaller, Maria & Guetl, Christian. (2014). Motivational Active Learning – Engaging University Students in Computer Science Education.
[14] Beth Simon, Ruth Anderson, Crystal Hoyer, and Jonathan Su. 2004. Preliminary experiences with a tablet PC based system to support active learning in computer science courses. SIGCSE Bull. 36, 3 (September 2004), 213–217.
[15] Jeffrey J. McConnell. 1996. Active learning and its use in computer science. SIGCUE Outlook 24, 1–3 (Jan.-July, 1996), 52–54.
[16] Krause, S., Middleton, J.A., Judson, E., Ernzen, J., Beeley, K., & Chen, Y. (2015). Factors Impacting Retention and Success of Undergraduate Engineering Students.
[17] Santiago, L. Y., & Hensel, R. A. (2012, June), *Engineering Attrition and University Retention* Paper presented at 2012 ASEE Annual Conference & Exposition, San Antonio, Texas.
[18] Santiago, L.Y., & Statler, B.M. (2013). Retention in a First Year Program: Factors Influencing Student Interest in Engineering.

[19] Desai, N.A., & Stefanek, G. (2017). An Introductory Overview of Strategies used to Reduce Attrition in Engineering Programs.
[20] Veenstra, Cindy & Dey, Eric & Herrin, Gary. (2009). A Model for Freshman Engineering Retention. Advances in Engineering Education. 1.
[21] Burtner, J. (2004). Critical-to-quality factors associated with engineering student persistence: the influence of freshman attitudes. *34th Annual Frontiers in Education, 2004. FIE 2004.*, F2E-1.
[22] Zhang, G., Min, Y., Ohland, M.W., & Anderson, T.J. (2006). The Role Of Academic Performance In Engineering Attrition.
[23] Budny, D., LeBold, W. and Bjedov, G. (1998), Assessment of the Impact of Freshman Engineering Courses*. Journal of Engineering Education, 87: 405-411.
[24] Holden, Edward & Weeden, Elissa. (2004). The experience factor in early programming education. 211-218.
[25] Chris Wilcox and Albert Lionelle. (2018). Quantifying the Benefits of Prior Programming Experience in an Introductory Computer Science Course. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 80–85.
[26] Holden, Edward & Weeden, Elissa. (2003). The impact of prior experience in an information technology programming course sequence. 41-46. 10.1145/947121.947131.
[27] Hagan, Dianne & Markham, Selby. (2000). Does it help to have some programming experience before beginning a computing degree program?. ACM Sigcse Bulletin. 32. 25-28. 10.1145/353519.343063.
[28] 2018 State of Computer Science Education. (2018). Retrieved from https://advocacy.code.org/
[29] Google Inc. & Gallup Inc. (2016). Diversity Gaps in Computer Science: Exploring the Underrepresentation of Girls, Blacks and Hispanics. Retrieved from http://goo.gl/PG34aH.
[30] Darby, F. and Lang, J. M. (2019). Small Teaching Online.
[31] Dawson, P., van der Meer, J.; Skalicky, J.; Cowley, K. (2014). On the Effectiveness of Supplemental Instruction: A Systematic Review of Supplemental Instruction and Peer-Assisted Study Sessions Literature Between 2001 and 2010. *Review of Educational Research*. *84* (4): 609–639.
[32] Abraham, N., & Telang, N. K. (2019, June). Effectiveness of the Supplemental Instruction Program in First-Year Engineering Courses-A Longitudinal Report (2015-2018). In *2019 ASEE Annual Conference & Exposition*.
[33] The Independent Samples t-test (Student Test). (2020, August 17). Retrieved April 16, 2021, from https://stats.libretexts.org/@go/page/4023
[34] Riegle-Crumb, C., Kyte, S. B., & Morton, K. (2018). Gender and racial/ethnic differences in educational outcomes: Examining patterns, explanations, and new directions for research. Handbook of the sociology of education in the 21st century, 131-152.Riegle-Crumb, C., Kyte, S. B., & Morton, K. (2018). Gender and racial/ethnic differences in educational outcomes:

Examining patterns, explanations, and new directions for research. *Handbook of the sociology of education in the 21st century*, 131-152.
[35] Burke, C., Luu, R., Lai, A., Hsiao, V., Cheung, E., Tamashiro, D., & Ashcroft, J. (2020). Making STEM Equitable: An Active Learning Approach to Closing the Achievement Gap. *International Journal of Active Learning*, *5*(2), 71-85.