2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
ASEE
Paper ID #37037

Board 79: Course Improvement of An Introduction to Programming Course in ECE: Customizing Learning Paths for Parallel Computing Topics

Isabel Alviar, University of Illinois Urbana-Champaign

Isabel received her B.S. in Computer Engineering from University of Illinois Urbana-Champaign in 2023. She is currently attending University of Illinois Urbana-Champaign pursuing her M.S. in Computer Science.

Dr. Ujjal K Bhowmik

https://ece.illinois.edu/about/directory/faculty/ubhowmik

Prof. Yuting W. Chen, University of Illinois at Urbana - Champaign

Dr. Yuting W. Chen received the B.S. degree from University of Illinois at Urbana-Champaign in 2007, and the M.S. and Ph.D. degrees from Rensselaer Polytechnic Institute in 2009 and 2011, all in Electrical Engineering. She is currently a Teaching Associate Professor in Electrical and Computer Engineering at University of Illinois at Urbana-Champaign. Her current interests include recruitment and retention of under-served students in STEM, professional development for graduate students, and curriculum innovation in computing.

Course Improvement of An Introduction to Programming Course in ECE: Customizing Learning Paths for Parallel Computing Topics

Abstract

As data sets grow larger and computational problems become more complex, parallel computing is increasingly recognized as a key solution for unlocking the potential of computer resources and achieving more efficient task resolution. Parallel computing offers many advantages, such as faster computations, significant cost savings, reduced energy consumption, and the ability to create dynamic models. Despite its widespread use in today's world, introductory Electrical and Computer Engineering (ECE) courses often do not cover this essential topic and its associated skills. To address this issue, a team at University of Illinois Urbana-Champaign has designed custom learning paths to introduce parallel computing at an earlier stage through additional learning modules, aiming to enable students to extend their knowledge while preparing them for advanced computing courses in the future. This paper focuses on the design, improvement, and assessment of the course, investigating the implementation and outcomes of optional learning opportunities created in a 200-level introduction to programming course. The paper presents the details of each extended learning opportunity and analyzes the differences in student performance on a related extra credit quiz in correlation with their course grade, the difficulty level of parallel computing topics introduced, and the lessons learned by students and instructors that can be applied to future programs.

Introduction

In most universities, students come with their own prior knowledge, skills, and conceptual understanding. Yet no matter a person's background, the earlier they are introduced to a concept, the less intimidating it would be, and curiosity begins to take hold. The Department of Electrical and Computer Engineering at University of Illinois Urbana-Champaign recognizes this and strives to create opportunities for students to build their knowledge and curiosity as early as possible. The first introductory programming class in the curriculum, ECE 220 (Computer Systems & Programming), covers numerous topics from assembly language to recursion. While there is not enough time to go in-depth on a subject as complex and important as parallel computing, its concepts are crucial in later programming courses such as ECE 385 (Digital Systems) and ECE 408 (Parallel Programming). This presents a challenge for students who may feel overwhelmed if the first time they encounter parallel computing topics is late in their upper-level classes. Additionally, there are students who already enter the class with an interest in parallel computing and wish to get further exposure. Therefore, a solution has been designed so that parallel computing can be introduced organically in ECE 220, both to enrich curious students who are interested in learning more and to prepare students to succeed in future advanced computing courses.

Extended Learning Opportunity for Honors Students

As part of the course redesign for ECE 220, an honors section was created for students who wanted to learn extra material outside of regular lectures. Five optional learning modules with infused parallel computing concepts were developed and launched in Spring 2020. For this honors section, students were expected to attend an extra discussion section once a week that covered these concepts, then complete each module, a programming assignment and report, individually by a deadline a few weeks after its release. The first two modules are very similar, involving implementation of both a private and shared parallel accumulator using C that calculates the sum of all numbers within a set of given files. After completing the first two modules, students should have a better understanding of how multithreading and synchronization work and observe the faster run-time of parallel computations over conventional sequential processing. The third module gives students exposure to linked lists and the merge sort algorithm by asking them to parallelize merge sort with pthreads. For the fourth module, students will implement a parallelized tree traversal algorithm that finds a specific element in a binary tree. They also get exposure to basic C++ programming skills such as objects, constructors, and destructors. The last learning module is optional to the honors section and involves writing the context switching code for a provided cooperative multitasking LC-3 assembly program.

While everyone is encouraged to join the honors section and complete these extended learning modules, they are non-trivial tasks, especially for students taking their first introductory programming course. Each optional coding assignment takes approximately 5 to 8 hours to complete, in addition to the regular coursework for ECE 220. Therefore, the barrier to entry is high and only a small number of students chose to complete them. However, it is important to still have this customized learning path as an option for students who *want* to learn these additional parallel computing topics, instead of forcing a heavier workload onto everyone in the course. Since the coding modules include topics unrelated to the regular coursework of ECE 220 and are not covered in assignments and assessments, the honors students would not gain any intentional grade advantage over those who did not choose that learning path. The honors section exists as a way to further encourage students who are curious about parallel computing, and hopefully influence their choice in taking related courses in the future.

Discussion on the Honors Section

When the honors section was launched in Spring 2020, a total of 23 students decided to pursue this extended learning opportunity. All 23 students completed the first two learning modules. Then, the semester was disrupted by COVID as classes abruptly shifted from in-person to online, and students were sent home from campus. Half of the students still chose to complete all four required modules and 3 even completed the optional LC-3 multitasking programming assignment as well. The honors section was not released for the semesters following due to difficulties with COVID. However, now that we have transitioned back to normal, the ECE 220 team has brought back the extended learning opportunity to a few honors students in Spring 2023 and is planning on reintroducing it fully in future semesters.

While most of the workload in the honors section is the programming modules, students are also required to complete a short report that addresses conceptual questions corresponding to the given programming assignment. To determine the effectiveness of these extended learning modules, we analyzed the questions asked for each report as well as the answers given by students in Spring 2020. For example, for the combined report on assignments 1 and 2, which involved the implementation of a private and shared parallel accumulator that calculates the sum of a given set of numbers, students were asked about the importance of using a lock to guard their result when running the program multiple times. They were also asked to compare the performance of their parallel accumulators when implemented sequentially (in programming assignment 1) versus with synchronization (in programming assignment 2). Students who answered that the sequential implementation in the first programming module was better received feedback on their report that the correct answer should be that using synchronization, such as in the second module, creates slightly better results. It is because each thread can perform the final sum operation so that the main thread can automatically output the final answer at the end of the program, instead of waiting longer from using partial sums. By analyzing commonly incorrect answers in the reports, the instructors could determine the concepts students were struggling with and needed further explanations in later discussions.

The reports also helped the ECE 220 team understand the students' thought processes throughout each module, and exemplified how there are different creative ways to solve the same problem. For example, after completing the fourth programming module, implementing a parallel algorithm that finds a specific element in a binary tree, students were asked to explain what threading model they chose, why they chose this model, and how their constructed tree looked like in their report. We summarized responses from each report and found similarities in how students approached the module. A few students described how their threading model involved setting a threshold equal to a certain number, denoting that after that number of active threads were launched, the program would have to switch to a sequential approach. Another group of students took a different approach along the lines of checking the topmost portion of the tree sequentially then assigning a set number of parallel threads after the correct depth was reached.

While the answers to these questions were helpful data points of students' conceptual understandings to the course instructors, more importantly, it also reinforced understanding to the students. Not only did they have to reflect on *what* parallelized concepts they just learned, but also *why* they were important and *how* they implemented them. With the modules consisting of a combined coding assignment and report, students were more likely to get a true grasp of parallel computing topics, instead of just blindly coding. The hope was that the students who chose to complete these extended learning opportunities in the honors section would feel more prepared, and even motivated, when being exposed to these same topics in later advanced computing courses. The original plan was to launch surveys for students entering the ECE 300 and ECE 400-level courses to see if they had completed these extended learning modules and whether the modules had any impact on piquing their interest in studying parallel computing topics further. This plan was postponed due to COVID and it will be considered again when more students are participating in the honors section in the future.

Optional Learning Module for All Students with An Extra Credit Quiz

The Electrical and Computer Engineering curriculum at University of Illinois requires students to take a sequence of two introductory courses, Introduction to Computing (ECE 120) and Computer Systems and Programming (ECE 220), as well as two advanced computing classes, Computer Systems Engineering (ECE 391) and Digital Systems (ECE 385). Since the advanced classes are designed to immediately build off the introductory ones, exposing concepts as soon as possible helps students become comfortable and even curious when they see these concepts again in more difficult forms.

Therefore, the goal of the course redesign for ECE 220 was to find creative ways to introduce the idea of parallel computing to more students early on, and not just in the form of a rigorous honors section. Beginning in Fall 2020, an optional, extra credit module on LC-3 multitasking was developed for all students enrolled in the course. Students could watch a pre-recorded video then take an asynchronous, online quiz consisting of 4 multiple-choice and true/false questions highlighting 4 different "zones" of parallel computing topics. To add elements of difficulty, students were only given one try per question and 10 minutes total to complete the quiz once it was started.

Discussion on the Extra Credit Quiz

To measure the overall learning outcomes of the extra credit quiz in introducing parallel computing topics, we evaluated two main questions:

1. Is there any difference in performance on the quiz between three groups of students divided based on their grades in the class?
2. Are there any parallel computing topics students consistently struggled with on the quiz? If so, how can the module be improved moving forward to lessen the frequency of these possible misunderstandings?

Difference in quiz performance based on overall course grades

Every semester, there are multiple sections of the course taught by different instructors. To ensure consistency and accuracy, we focus on four semesters of data from Fall 2020 to Spring 2022. The 841 students that chose to take the optional extra credit quiz across all sections and semesters are treated equally by being separated into three groups based on their overall grade in the class: Group 1 (A- or above), Group 2 (B- to B+), and Group 3 (C+ and below). For each semester, we calculated the average quiz score across groups, and compared the results and trends visually as shown in Figure 1.

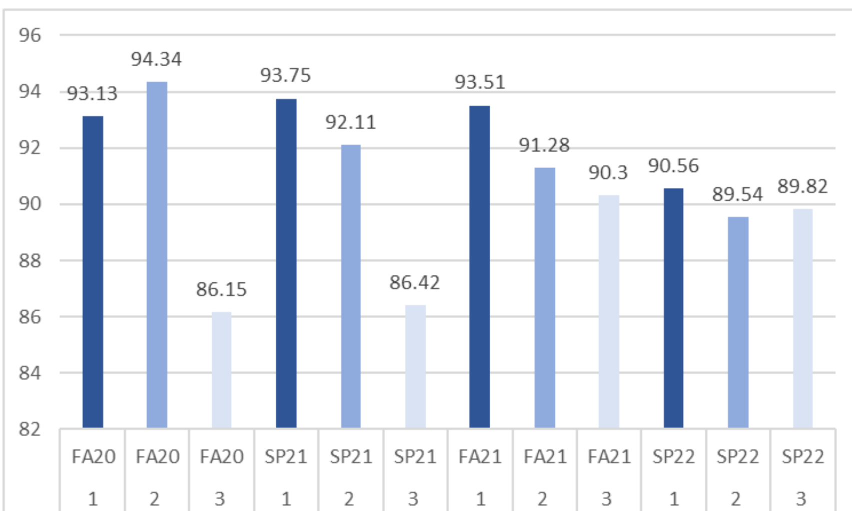

Figure 1: Average quiz score of the three groups in each semester

In Figure 1, we observed that students in Group 1 consistently scored higher on the quiz than the other two groups, except in Fall 2020 where they scored slightly lower than Group 2. However, the average quiz score for Group 1 was always an A (above 90%) across the four semesters of data. Students in Group 3 consistently performed the worst on the quiz, except in Spring 2022. There was a delta of around 1 to 7 percent between the average score of Group 3 and the next highest group, depending on the semester. Since the LC-3 Multitasking module was optional, we typically saw more participation from students in Groups 1 and 2 based on the number of students who took the quiz each semester. This could be due to many factors, including more time, motivation, interest, etc. amongst students in these groups. As time went on, there was a positive increase in scores for students in Group 3. In Fall 2020 and Spring 2021, Group 3 only averaged an 86% on the quiz, which was relatively low compared to the other groups with averages higher than 90%. By the next academic year, in Fall 2021 and Spring 2022, the difference between the results of Group 3 versus the others was small, where all three groups were averaging high scores close to each other. However, it still appeared that how well a student performed on the quiz correlated to how well they were doing in the actual class.

To verify our findings, we conducted a one-way ANOVA test amongst the scores across the three different groups, where the semester students took the quiz in was no longer a relevant variable. ANOVA tests are used to test variance, in other words, how more than 2 different groups (in this case, students divided based on class grade) responded between a particular factor (in this case, quiz scores). The idea is to take independent random samples from each group, then compute each of their sample means. Then, the variation of sample means among the groups is compared to the variation within the groups. Finally, the calculated p-value or f-statistic is used to decide whether the means of the groups are all equal or not.

Table 1: Data summary, ANOVA summary, and analysis of variance results

Data Summary				
Groups	**N**	**Mean**	**Std. Dev.**	**Std. Error**
Group 1	374	92.246	12.5463	0.6488
Group 2	272	91.3971	13.8879	0.8421
Group 3	195	88	15.5185	1.1113

ANOVA Summary					
Source	**Degrees of Freedom DF**	**Sum of Squares SS**	**Mean Square MS**	**F-Stat**	**P-Value**
Between Groups	2	2378.3888	1189.1944	6.3191	0.0019
Within Groups	838	157702.4132	188.189		
Total:	840	160080.8019			

Using a p-value of 0.05, our null hypothesis was that students in Groups 1-3 performed the same on the quiz, and our alternate hypothesis was that they did not. Since the p-value we calculated through the ANOVA test (0.0019) was less than 0.05, we rejected the null hypothesis. Therefore, we had enough evidence to conclude that the students in three different groups divided based on their course grades did *not* perform the same on the quiz. Table 1 above shows a summary of the results of the ANOVA test. When comparing the three groups, quiz results correlated with overall class performance, where Group 1 (students in the A range) scored the best, Group 2 performed the next best, and Group 3 did the worst. There are many possible explanations for this, but it is most likely that students in Group 3 are already struggling with the regular ECE 220 content, so asking them to learn more topics can create an even bigger challenge. The goal in creating the quiz was for it to be an *extended* learning opportunity, not to create another mandatory assignment that could put already struggling students at a further disadvantage in the class. Although the module was extra credit, any points scored on the quiz could only be added to the quizzes category to make up for lost points. For example, if a student scored 100% on the extra credit quiz, and their overall quiz grade was not at full credit, any points earned from the extra credit quiz would be added onto this category. This incentivizes students while also ensures that their grades cannot go above 100%, thereby giving them a large grade advantage over others who chose not to complete the extra credit quiz or did not perform as well on it. Many students, especially in Groups 1 and 2, did not need extra credit and still chose to complete the optional parallel computing module for their own enrichment.

Moving forward, it would be beneficial to create a short, optional survey to distribute at the end of the extra credit quiz or semester to measure the effectiveness of the module and understand what changes can be made to improve it to further encourage early exposure to parallel computing topics. The survey would not only serve as a source of data for students' motivations for completing the module and their future course plans in relation to parallel computing, but also allow them to provide feedback.

Difficulty of parallel computing topics

When creating the extra credit quiz as an extended learning module for ECE 220, the goal was to make the barrier to entry low so that as many students as possible were encouraged to learn parallel computing topics early on. Therefore, the quiz was short, consisting of only four questions that tested the core concepts the ECE 220 team decided were most important. Each question on the quiz corresponded to a specific "zone" of parallel computing topics, where Zone 1, or Question 1, tested concurrent tasks. Zone 2 highlighted context switching for cooperative versus preemptive multitasking. There were two different versions of questions randomly generated for Zones 1 and 2. Zone 3 expanded on preemptive context switching specifically. Finally, Zone 4 was a survey question that asked students what provided or outside materials they found helpful in preparation for the quiz.

The mean quiz score of all students from Fall 2020 to Spring 2022 was high at 90.91%. However, as shown in Table 2, there was variation between how well students performed on questions in each zone. Therefore, we analyzed the parallel computing topics students consistently struggled with on the quiz, and evaluated how the module could be improved moving forward to correct any misunderstandings. Since it was a survey question, every student scored 100% on Zone 4 as there were no incorrect answers. On the other hand, students consistently performed worst on the question for Zone 3, with a mean score of 77.65% across the four semesters. Compared to a mean score of 96.55% on Zone 2 Version 2 (Zone2_V2), this could indicate an overall misunderstanding in preemptive context switching and preserving registers amongst ECE 220 students.

Table 2: Average quiz scores per question for each zone and semester

Question	Semester	Mean	SD	n
Zone1_V1	Fall 2020	89.66	30.45	169
Zone1_V2		97.59	15.33	
Zone2_V1		92.39	26.51	
Zone2_V2		85.59	34.8	
Zone3		73.53	44.12	
Zone4		100	0	
Zone1_V1	Fall 2021	94.23	23.32	109
Zone1_V2		94.74	22.33	
Zone2_V1		92.59	26.19	
Zone2_V2		76.36	42.48	
Zone3		79.81	40.14	
Zone4		100	0	
Zone1_V1	Spring 2021	93.16	25.24	236
Zone1_V2		97.48	15.67	
Zone2_V1		89.81	30.25	
Zone2_V2		81.25	39.03	
Zone3		77.54	41.73	
Zone4		100	0	
Zone1_V1	Spring 2022	89.15	31.1	327
Zone1_V2		96.41	18.61	
Zone2_V1		86.25	34.44	
Zone2_V2		74.26	43.72	
Zone3		79.73	40.2	
Zone4		100	0	

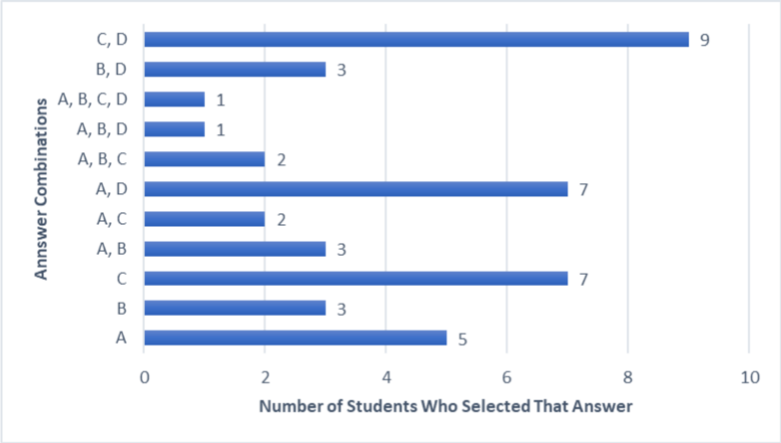

Figure 2: Common incorrect answers by students for Question 3

Since the question in Zone 3 allowed students to select all the answers they thought were correct, we also analyzed common mistakes made by students who got this question wrong. In order to do so, we extracted all of the combinations of answers given by students in Fall 2020, then counted the number of students who chose those answers. Although students could select up to any number of the four options, the correct answer choice was just D. By examining the data from Fall 2020, we found that the most common mistakes were students choosing answers C and D, A and D, or just C, as shown above in Figure 2. Quiz data from the three other semesters, from Fall 2021 to Spring 2022, reflected similar results. Therefore, we concluded that many students who got the question wrong still understood that the program status register needed to be preserved in preemptive context switching (answer choice D). However, there were some misconceptions between whether the instruction register also needed to be preserved (answer choice C), where the correct answer was that it did not need to be.

The results from the quiz allowed instructors to determine which parallel computing concepts students were struggling with or had any confusion on that needed further explanation in later versions of the pre-recorded lecture video and slides provided as part of the module.

Additionally, two different versions of questions were created for Zones 1 and 2 then randomly selected for each student who took the extra credit quiz. This was implemented to discourage the case that one student took the quiz and then shared the answers with others. Although both versions of the zones tested the same general concepts of concurrent tasks (Zone 1) and context switching (Zone 2), there were differences in performance on each question version. By examining mean scores of each question across four semesters of data, it appeared that students consistently scored lower on question version 1 in Zone 1 and question version 2 in Zone 2.

Table 3: Independent t-test results on Zone 1 and Zone 2 question scores

Zone_Question	n	df (n-1)	Mean	p-value
Zone1_V1	402	401	91.04	0.000612
Zone1_V2	443	442	96.39	
Zone2_V1	430	429	89.77	0.00001
Zone2_V2	415	414	78.07	

To verify our observations, we performed independent t-tests on Zone 1 and Zone 2 separately. An independent t-test is used to compare whether the means of two groups are different. In our case, the two groups were the two different versions of the question, where the students taking the quiz had an equal chance of getting either version due to random generation. We used a p-value of 0.05, and our null hypothesis was that the means for the two populations were equal, meaning that students performed relatively the same on both versions of the questions in each zone. Our alternate hypothesis was that the means for the two populations were not equal, or the scores for each question version were not the same. Since the p-value we calculated from running an independent t-test on the two question versions in Zone 1 (approximately 0.000612) was less than 0.05, we rejected the null hypothesis. Therefore, we had enough evidence to conclude that students performed significantly better on the second version of the question for Zone 1. Since the calculated p-value (0.00001) was also less than 0.05 for the independent t-test we ran on Zone 2, we rejected the null hypothesis. This meant that we had significant evidence to conclude that students did better on Zone2_V1 than Zone2_V2.

A possible reason for the difference in performance on question variants for Zone 1 could be due to wording, where one question's wording is more challenging or confusing than the other despite being on the same topic. The difference in performance on the question variants in Zone 2 could be due to the difficulty level of preemptive multitasking being higher than cooperative multitasking, which is generally more straightforward. For future semesters, we will continue having different variants of questions for Zones 1 and 2 to discourage academic integrity violations. However, we now also know that certain parallel computing topics are more difficult than others and require further emphasis when being taught. The ECE 220 team will work to adjust the questions and introduction video to better explain preemptive multitasking (from Zone 2) and preemptive context switching (from Zone 3).

Table 4: Number of students whose grade changed caused by extra credit quiz

Change	Number	% of students taking the quizzes
A to A+	2	1.18
A- to A	3	1.78
B+ to A-	1	0.59
B to B+	0	0
B- to B	1	0.59
C+ to B-	2	1.18
C to C+	1	0.59
C- to C	0	0
D+ to C-	0	0
D to D+	0	0
D- to D	3	1.78
F to D-	1	0.59
Sum	14	8.28

In addition to successfully introducing parallel computing topics, the optional extra credit module proved to be helpful in boosting some students' grades as well. The average quiz results were very high across all 3 groups in every semester, where scores were always above 86%. After more closely examining the data of students in Fall 2020, the first semester the learning module was released, we found that students who took the quiz added between 0.2 to 0.5 percent to their overall grade. In just one semester, we saw 14 students' grades change across levels (for example, A to A+, B- to B), as seen in Table 4 above.

The optional quiz not only allowed students to learn interesting parallel computing concepts with no risk to their grade in ECE 220, it also positively reinforced that putting in a little extra work to learn new related concepts early on can help their grades, both directly in their current class and indirectly by being better prepared for future computing courses.

Conclusion

As part of our goal to introduce parallel computing topics to students early on in ECE 220, an introduction to programming course at University of Illinois Urbana-Champaign, we created two optional and customized learning paths. The first optional path was created for the honors section of the course, in which students would learn more about parallel computing outside of the base curriculum. In addition to attending extra discussion sections, students completed programming assignments that covered topics such as multithreading, synchronization, pthreads, etc. Due to COVID and the heavier workload it entails, this option was only taken by a very small percentage of the class. In general, we found that if an extended learning opportunity such as the honors section required a lot of time, it was to be expected that a small number of students would choose this path, due to the already rigorous engineering curriculum. The second optional path was an extra credit learning module and quiz on multitasking, which all students in the class were encouraged to complete. When creating this optional module, we were mindful of all students in the course, especially those who are already struggling, by choosing a grading scheme carefully and emphasizing that the module was an extended learning opportunity, not an additional expectation or burden. Participation rate has been relatively high since the launch of this module (77.5%, 78.7%, 54.2%, and 78.6%, respectively, from Fall 2020 to Spring 2022). In the future, we plan to continue making improvements to the modules, create surveys to collect more student feedback, and conduct a follow-up study of student performance in subsequent courses to analyze the impact of these extended learning opportunities.

References

[1] S. J. Matthews, J. C. Adams, R. Brown, and E. Shoop, "Incorporating parallel computing in the undergraduate computer science curriculum," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2020.

[2] Z. Zhu, U.K. Bhowmik, Y. Wang, Z. Cheng, and Y. W. Chen, "Having it all: infusing parallel computational thinking in the lower-level computer engineering curriculum using extended learning modules," in *Proceedings of the 2021 ASEE Virtual Annual Conference*, 2021.

[3] S. K. Prasad. "NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing - Core Topics for Undergraduates (Version 2.0-Beta)." [Online]. Available: https://tcpp.cs.gsu.edu/curriculum/. [Accessed: 28-Apr-2023].

[4] Y. Patt and S. Patel, *Introduction to computing systems: From bits & gates to C/C++ & beyond*, 3rd ed. Columbus, OH: McGraw-Hill Education, 2019.

[5] M. Damian, "POSIX Threads." [Online]. Available: http://www.csc.villanova.edu/~mdamian/threads/posixthreads.html. [Accessed: 28-Apr-2023].

[6] G. Ippolito, "POSIX Thread (Pthread) Libraries." [Online]. Available: https://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html. [Accessed: 28-Apr-2023].

[7] "LibGuides: SPSS tutorials: One-way ANOVA." [Online]. Available: https://libguides.library.kent.edu/SPSS/OneWayANOVA. [Accessed: 28-Apr-2023].

[8] "LibGuides: SPSS tutorials: Working with 'check all that apply' survey data (multiple response sets)." [Online]. Available: https://libguides.library.kent.edu/SPSS/Multiple-Response-Sets. [Accessed: 28-Apr-2023].