The Relationship between Prerequisite Proficiency and Student Performance in an Upper-Division Computing Course

Sander Valstar, William G. Griswold, Leo Porter

University of California, San Diego

{avalstar,wgg,leporter}@eng.ucsd.edu

ABSTRACT

While it is widely believed that taking a class's prerequisites is critical for success, less is known about how proficiency with the prerequisite knowledge from those courses affects performance in later courses. Specifically, it is unclear how well students understand material from prerequisite courses and whether that understanding may impact their outcomes in the subsequent course. Additionally, in subsequent courses, do students strengthen their knowledge from prerequisite courses and, if they do, does that improvement matter for the subsequent course? This study examines the prerequisite knowledge of 208 students in an upper-division data structures class at a large North American research university. Prerequisite proficiency on entry to the course was surprisingly low, with nearly a third of students demonstrating low proficiency and only a quarter high proficiency. Students modestly improved their proficiency during the term, lifting a third of those with low proficiency to at least medium proficiency. Overall, final exam performance was significantly correlated with prerequisite knowledge. For those with low initial proficiency, improvement in proficiency was significantly correlated with performance on the final. These results suggest that more attention needs to be placed on reinforcing prerequisite knowledge for those with low proficiency.

KEYWORDS

Prerequisites, data structures, student performance

ACM Reference Format:

Sander Valstar, William G. Griswold, Leo Porter. 2019. The Relationship between Prerequisite Proficiency and Student Performance in an Upper-Division Computing Course. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19), February 27-March 2, 2019, Minneapolis, MN, USA*. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3287324.3287419>

1 INTRODUCTION

Although prerequisites have been shown to be valuable for student outcomes outside computing [9, 18], their value has not been empirically studied in computing. Studying the importance of prerequisite success takes on greater meaning in light of institutional pressure to reduce failure rates, as well as recent discussions about reducing prerequisites to speed time to degree [19].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCSE '19, February 27-March 2, 2019, Minneapolis, MN, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-5890-3/19/02...\$15.00

<https://doi.org/10.1145/3287324.3287419>

Many instructors have a story about the student who somehow enrolled in their course without a required prerequisite, suffering dire consequences. But what happens when a student has taken the prerequisite and just did poorly in that class—does low prerequisite knowledge relate to poor performance, and if so, to what extent? What percentage of students enter courses with insufficient proficiency of required prerequisites? Do students improve on their deficiencies in prerequisite knowledge during the course?

To answer these questions, we measured the prerequisite knowledge of 208 students in an upper-division data structures class through a pre- and post- survey. From these results, we examined changes in proficiency over the term, correlated pre-term and post-term proficiency with the students' final exam scores, and analyzed which prerequisite content was most salient for student success.

We expected that proficiency with prerequisite knowledge would be generally high as the prerequisites for this course are reported to be rigorous and demanding. Yet we also had concerns that the very low rates of D's and F's (< 7% combined) in the prerequisites could mean that some instructors were giving passing grades to students who are not truly prepared for the next step. Other possible factors include fading knowledge for those not taking these courses back-to-back [3] and cheating in prerequisite courses [11, 15].

The results were both sobering and encouraging. Prerequisite proficiency on entry to the course was far lower than anticipated. We classified student prerequisite proficiency as low (under half of the questions correct), medium (50-70%), or high (over 70%). Nearly a third of students (29%) demonstrated low proficiency and only 27% demonstrated high proficiency. On the other hand, students improved their proficiency over the term, by 8 percentage points on average, with a number of low-proficiency students improving to at least medium proficiency by the time of the final.

Not entirely surprising is our finding that final exam performance is correlated with prerequisite proficiency, 0.366 (Pearson correlation coefficient) for the pre-term test, 0.384 for the post-term test, and 0.434 for the two taken together. Improvement in prerequisite proficiency during the term mattered: for students demonstrating low proficiency on the pre-term prerequisite test, their improvement on the post-term test was correlated (0.371) to their performance on the final. All of these correlations were statistically significant. Not all prerequisite knowledge was found to be equally important, however, nor was proficiency uniform over the subject matter. For example, a student's ability to perform a basic runtime analysis was the strongest predictor of performance on the final. We discuss which topics were of most importance in detail.

Taken together, these results show the importance of proficiency with prerequisite knowledge for future performance in computing, and highlight the possible benefits of putting more attention on the reinforcement of prerequisite knowledge.

2 BACKGROUND

Course prerequisites are commonly used in curricula to ensure students have the necessary prerequisite knowledge to succeed in subsequent courses. In computing, course prerequisites are described in the ACM curricula recommendations CS 2013 [12]. Prerequisites can play a variety of roles in the curriculum, including ensuring students have the proper background knowledge or sufficient maturity, that they display a certain degree of commitment to a program, or to convey program requirements [19]. In particular, the role of mathematics and introductory computing prerequisites have been well discussed [19]. There is pressure, however, to reduce the depth of prerequisite chains to aid with students' time to degree [19]. In view of these discussions, the role prerequisite proficiency plays in student success in computing courses deserves further study.

Outside computing, Martin found that grades in prerequisite courses, specifically statistics and mathematics, correlated with student performance in an agriculture price analysis course [9]. Expanding on the work of Martin in agriculture, Vitale et al. found that cumulative GPA and grades in prerequisite courses predict student grades in a Farm and Agribusiness Management course [18].

In computing, direct connections between prerequisite course grades or knowledge from prerequisite classes has not been similarly established. However, there has been substantial work examining a myriad of factors that determine students success in introductory programming courses. These factors include prior knowledge, gender, motivation, self-efficacy, and achievement goals [8, 13, 21, 23]. Although each of these factors have shown promise, Wilson et al. recently cautioned that these relationships may not always replicate [20], which has led to large replication efforts [22]. Our work expands on our understanding in computing by establishing a direct relationship between prerequisite course knowledge and success in upper-division computing courses.

The study most closely related to ours is that of Muralidharan et al. [10]. Muralidharan et al. examined student performance in Hindi and mathematics with 8th grade students in India, finding that a surprising majority of students are lacking prerequisite knowledge from prior grade levels. Their study goes on to show that an intervention program targeting prerequisite knowledge can significantly improve outcomes. In our study, we similarly examine student prerequisite knowledge entering an Advanced Data Structures course, finding that incoming students demonstrate a wide range of proficiency and that improvement in prerequisite proficiency during the course is correlated with improved final scores.

3 RESEARCH QUESTIONS

Our study has five primary research questions:

- RQ1. Are most students arriving with sufficient prerequisite knowledge to start subsequent courses?
- RQ2. Do students, particularly those with low preparedness, improve their proficiency in prerequisite content as they progress through subsequent courses?
- RQ3. Does incoming prerequisite proficiency impact final exam scores?
- RQ4. If there is a relationship between incoming prerequisite proficiency and final exam scores, which prerequisites are most meaningful?

- RQ5. Does improvement in prerequisite proficiency translate to improved final exam scores?

4 STUDY DESIGN

To understand the influence of prerequisite proficiency on final exam performance we chose to study an instance of the course "Advanced Data Structures" (ADS), an early upper-division course in our CS program. This is an excellent candidate for our study because it has a wide range of prerequisites including software, theory, and hardware. As we were interested in actual proficiency at the time of taking the course, not just performance in the prerequisite courses, we tested proficiency at the beginning and the end of the term. Our study has institutional Human Subjects approval.

4.1 Course Context

The ADS course we studied was taught at a large North American research university on the quarter system. This 10-week course is designed for CS majors and minors, and is one of the gateway classes to other advanced courses in the major. The goal of the class is for students to learn how to program and analyze high-performance data structures. The data structures studied include trees, hash tables, Huffman coding, and graphs. Related topics include memory management, pointers, and recursion. Assignments are programmed in C++. The course has three principal prerequisites, all lower-division courses: introductory data structures (CS2), advanced discrete math, and computer organization.

4.2 Data Collection

In order to be able to study the relative levels of prerequisite proficiency we needed a tool to broadly examine the prerequisite knowledge of the students in this class. We chose to test the prerequisite knowledge of students by means of two surveys. In a pre-study for the present study, we ran the surveys online. However, we found a significant mismatch between student correctness in the online survey versus questions which were subsequently given in class. This suggested that students collaborated or used outside aids while completing the survey online. So, for this study, we performed the data collection with an in-class survey, on paper. The first survey was taken during the first day of class and the second survey was taken during the last week of class, before the final exam. Both surveys consisted of 15 multiple-choice questions and students were given 20 minutes for each survey. This format was chosen to balance the needs of data capture with the time taken away from the class lecture time.

4.3 Survey Content

The following courses are prerequisites for majors for the upper-division data structures class at our institution:

Table 1: Survey questions listed by question topic and related prerequisite courses.

Q	Topic	Related courses
1	Pass by value/reference	CS1
2	Valid Heap not BST	CS2 - Basic Data Structures
3	Valid Heap not BST	CS2 - Basic Data Structures
4	Valid BST not Heap	CS2 - Basic Data Structures
5	Neither Heap nor BST	CS2 - Basic Data Structures
6	Best List implementation	CS2 - Basic Data Structures
7	Runtime analysis	CS2 - Basic Data Structures and Adv. Discrete Math
8	Invariants	Discrete Mathematics and Adv. Discrete Math
9	Tracing recursion	CS1 and Adv. Discrete Math
10	Assembly	Computer Organization
11	Bitwise arithmetic	Computer Organization
12	Little-endian addressing	Computer Organization
13	Pointers	Computer Organization
14	Probability	Discrete Mathematics
15	Tracing recursion	CS1 and Adv. Discrete Math

Table 2: Low, medium, and high groups on Survey 1.

Group	Threshold	# students	% students
lowS1	$0 \leq \text{score } S1 < 8$	61	29.3%
mediumS1	$8 \leq \text{score } S1 \leq 10$	90	43.3%
highS1	$10 < \text{score } S1 \leq 15$	57	27.4%

Lacking validated assessments for direct prerequisite courses, we selected questions for our survey from those used in prior research studies on these topics [4, 6, 7, 14]. In cases where such a question was not available, we asked the instructor of the corresponding prerequisite course to supply us with two easy and two medium difficulty multiple-choice questions for us to use.

The two surveys were designed to be essentially equivalent, matched question-for-question. To limit the effects of memorization between the two surveys, small changes were made to the questions in the second survey, making sure that the changes did not make a question harder or easier. For example, on a tracing question, it was important that the number of statements that needed to be traced stayed the same. Table 1 lists the question topic and associated prerequisite courses for each survey question. Publicly hosted PDF versions of both surveys can be found online [2]. Because questions 2-5 are topically related, we reversed the order of these questions between the first and second survey. In the table (and the rest of this paper), the order of questions 2-5 for Survey 2 has thus been reversed so that they align with the equivalent questions in Survey 1. For example, question 5 on Survey 2 is treated as question 2 in the table.

4.4 Population

The first survey was taken by 365 students, the second by 236 students, and the final by 363 students. A total of 209 students took both surveys and 208 of those students took both surveys and the final exam. Unless stated otherwise, the statistics in this paper are based on these 208 students.

Figure 1: Distribution of Scores on Surveys 1 and 2. Respectively: avg=8.6 (57%), 9.8 (65%); median=9, 10; std dev=2.7, 2.7.**Table 3: Low, medium, and high groups on Survey 2.**

Group	Threshold	# students	% students
lowS2	$0 \leq \text{score } S2 < 8$	44	21.2%
mediumS2	$8 \leq \text{score } S2 \leq 10$	73	35.1%
highS2	$10 < \text{score } S2 \leq 15$	91	43.8%

Figure 2: Distribution Score Change. The average score change was 1.2 points (8%), with a median of 1 and a standard deviation of 2.7.

5 RESULTS

5.1 RQ1: Early-Term Prerequisite Proficiency

For Survey 1 students scored an average of 8.6 out of 15 points with a median of 9 and a standard deviation of 2.7. Figure 1 shows a bar chart of the distribution.

Table 2 shows the amount of low-, medium-, and high-performing students for Survey 1. We define low performance as below 50% (below 8) and high performance as above 70% (above 10). As listed in Table 2, just over 29% of students answered fewer than 50% of the questions on prerequisite topics correctly.

In sum, the students entering a subsequent course like ADS exhibit a wide distribution of prerequisite proficiency, with about a third having considerably lower proficiency than expected.

5.2 RQ2: Change in Proficiency during Term

For Survey 2 students scored an average of 9.8 out of 15 points with a median of 10 and a standard deviation of 2.7. Figure 1 shows a bar chart of the score distribution for Survey 2. Table 3 shows the number of low-, medium-, and high-performing students for Survey 2 using the same thresholds as Survey 1.

Table 4: Low, medium, and high groups on the Final Exam.

Group	Threshold	# students	% students
lowFinal	$0 \leq \text{score} < 59$	64	30.8%
mediumFinal	$59 \leq \text{score} \leq 65.5$	76	36.5%
highFinal	$65.5 < \text{score} \leq 75$	68	32.7%

Figure 3: Comparison of Survey 1, Survey 2, and final scores. Low, medium, and high scores are defined in Table 4. A data point's size conveys the number of students under the point.

The score improvement between Surveys 1 and 2 was 1.2 (8%) on average, with a median of 1 and a standard deviation of 2.7. The scores on Survey 2 were significantly different compared to Survey 1 with a p-value less than 10^{-4} . As seen in Figure 2, most students improved their score by 0, 1, or 2 points out of 15. This resulted in over a quarter of the low performers on Survey 1 moving to medium or high proficiency. Overall, these results demonstrate that for the prerequisites in this study, as students progress in a subsequent course, they improve their prerequisite proficiency.

5.3 RQ3: Relationship between Prerequisite Proficiency and Final Exam Scores

The average score on the final exam was 61.8 points out of 75 (82%), with a median of 63 (84%) and a standard deviation of 6.9 (9%). For both surveys, the students' scores are significantly correlated to their final exam performance. For Survey 1 the correlation to final exam grades is 0.366, for Survey 2 the correlation is 0.384, and for the sum of the scores for Survey 1 and 2 the correlation is 0.434. The p-values for these Pearson correlations are all below 10^{-7} .

The same correlations are illustrated in Figure 3. We binned students based on their final exam performance as low, medium, or high performers. To obtain a balanced grouping we used the two tertile points of the score distribution as thresholds. The reason the group sizes are not perfectly in balance despite using the tertile points as thresholds is because multiple students received the same final score. Notable is that these thresholds are at 79% and 87% of a full score, so on average students did much better on the final exam than on our prerequisite surveys. Table 4 shows the groupings.

In Figure 3, each student is plotted in the scatter plot with the x axis being their score on Survey 1 and the y axis being their score on

Survey 2. The color and shape of the data point indicates whether the student performed low, medium, or high on the final. The size of the data point indicates the number of students represented by it. Note that in cases when there were students from multiple performance levels in the same point, multiple data points appear (e.g., a large green triangle and a small yellow circle). Where these multiple points occur, the color may appear as combinations of others (e.g., orange is a combination of red and yellow).

Figure 3 hence visualizes the correlation between prerequisite proficiency and final exam performance. Most strikingly, it shows that most of the high performing students on the final were those who performed well on both Survey 1 and Survey 2 (upper right quadrant). Overall, we find a strong correlation between prerequisite proficiency and final exam performance.

5.4 RQ4: Which Prerequisites Relate to Final

To gain insight on whether prerequisite knowledge can be unbundled into more- and less-relevant topics, we performed a few per-question statistical analyses. Table 5 lists the correlation of the individual survey questions to the final exam scores, providing a hint as to which prerequisites might be important for performance on the final. Significantly correlated questions are listed in bold. Question 7, which concerns runtime complexity analysis, is the highest (significantly) correlated question for both Survey 1 and 2.

Some questions may be harder than others, reducing correlation at the extremes because there is little to distinguish high and low proficiency. Additional signal might be teased out of such questions by using conditional probabilities. For example, getting the right answer on a really hard survey question might be associated with high performance on the final, yet getting it wrong might say little about doing poorly on the final. Such distinctions are captured by the concepts of Sufficiency and Necessity [5]. A high likelihood of high performance when getting a survey question right is interpreted as meaning that the prerequisite knowledge is *sufficient* for success. A low likelihood of high performance when getting a survey question wrong is interpreted as meaning that the knowledge is *necessary* for success.

We express sufficiency as a likelihood ratio—the ratio that one outcome is more likely than another—specifically the positive likelihood [5]:

$$\text{Sufficiency} = \frac{P(E|H)}{P(E|\sim H)}$$

where the variable E stands for Evidence (in our case answering a question correctly), H stands for Hypothesis (the student falling in the top 50% of the final exam scores), and $E|H$ means E given H (all students that answered the survey question correctly and were in the top 50% of the final exam scores).

Necessity is the negative likelihood, that of finishing in the top half of the final if a student answers a question *incorrectly*, that is, $\sim E|H$. Since we are interested in the association between answering the question incorrectly and being in the *bottom* 50% on the final exam, $\sim E|\sim H$, we use inverse necessity (inverse negative likelihood):

$$\text{Inverse Necessity} = \frac{P(\sim E|\sim H)}{P(\sim E|H)}$$

The question most strongly associated with high final exam performance when answered correctly (sufficiency) is question 12 for both Survey 1 and 2. Its sufficiency of 2.10 on Survey 2 means

Table 5: Statistics for survey questions: percentage of correct answers and comparison to final exam scores: correlation with p-values, sufficiency, and inverse necessity. Bolded values indicate significant correlation for $\alpha = 0.05$.

Q	Subject	Survey 1					Survey 2				
		correct	correl	p-val	suff	nec ⁻¹	correct	correl	p-val	suff	nec ⁻¹
1	Pass by value/reference	64%	0.050	0.470	1.13	1.25	71%	0.103	0.139	1.24	1.68
2	Valid Heap not BST	44%	0.006	0.937	0.99	1.00	44%	0.065	0.350	1.27	1.20
3	Valid Heap not BST	56%	0.190	0.006	1.36	1.46	75%	0.193	0.005	1.21	1.80
4	Valid BST not Heap	62%	0.055	0.434	1.04	1.07	83%	0.126	0.069	1.14	1.90
5	Neither Heap nor BST	76%	0.108	0.120	1.15	1.55	90%	0.063	0.370	1.03	1.37
6	Best List implementation	61%	0.032	0.646	1.13	1.21	65%	0.071	0.307	1.06	1.12
7	Runtime analysis	61%	0.329	0.000	1.85	2.56	69%	0.284	0.000	1.34	1.92
8	Invariants	34%	0.009	0.895	0.89	0.94	41%	0.072	0.299	1.05	1.03
9	Tracing recursion	78%	0.119	0.087	1.05	1.17	77%	0.211	0.002	1.16	1.65
10	Assembly	56%	0.200	0.004	1.43	1.57	43%	0.235	0.001	1.59	1.40
11	Bitwise arithmetic	58%	0.155	0.025	1.25	1.35	67%	0.217	0.002	1.36	1.86
12	Little-endian addressing	21%	0.194	0.005	1.91	1.18	27%	0.252	0.000	2.10	1.30
13	Pointers	69%	0.244	0.000	1.47	2.35	69%	0.203	0.003	1.32	1.87
14	Probability	55%	0.215	0.002	1.34	1.42	72%	0.265	0.000	1.36	2.19
15	Tracing recursion	70%	0.198	0.004	1.16	1.40	88%	-0.049	0.479	1.02	1.12

Table 6: Correlation between change in survey score and final exam score, broken down by the groups defined in Table 2. Significant correlations are in bold for $\alpha = 0.05$ and underlined for $\alpha = 0.005$.

Group	avg. impr.	med. impr.	σ	corr.	p-val
lowS1	2.90	3	2.8	<u>0.371</u>	<u>0.003</u>
mediumS1	1.01	1	2.4	0.114	0.284
highS1	-0.37	0	2.0	0.306	0.020

that students who answered it correctly had more than twice the probability of ending up in the top half of the class than in the bottom half. Question 12 was deemed difficult by the authors of this paper, and the students performed worst on it.

The question most highly associated with low final exam performance when answered incorrectly (inverse necessity) is question 7 on Survey 1 and 14 on Survey 2. Question 7 was deemed by the authors as solvable with just a basic proficiency. Question 7 stands out for being strong for sufficiency, inverse necessity, and correlation. As such, we can safely (and unsurprisingly) conclude that early-term proficiency in runtime analysis is crucial to good performance in an Advanced Data Structures class. The second highest question for inverse necessity was question 13, a basic question on pointers. As the data structures class jointly teaches C++ programming, the importance of having a background in C is unsurprising. The high necessity for question 14 on Survey 2, regarding basic probability, was a surprise, as ADS does not explicitly use probability. Question 10, regarding assembly language, inexplicably exhibits a similar pattern. The necessity of an apparently unrelated topic raises the possibility that there are dependences that remain to be understood or may be indicative of an intervening variable (e.g., study skills, mathematical reasoning, maturity, etc.).

Overall, we find that most prerequisite topics exhibit positive correlations and modest sufficiency and necessity. Some related topics that exhibit the strongest values, like runtime analysis, make sense. However, the high inverse necessities for unrelated topics leave many open questions.

5.5 RQ5 - Relationship between *Improvement in Prerequisite Mastery and Final Score*

The correlation of score change between Surveys 1 and 2 for the entire class was small and insignificant at 0.02, p-value of 0.77. Examining the course on a whole can mask effects for subpopulations, and we were curious if low performers on Survey 1 saw a significant benefit from improving their prerequisite proficiency. Examining student performance based on the groups defined in Table 2, lowS1 and highS1 have significant correlation, as shown in Table 6. We do not find significance for mediumS1, however their 1-point improvement is small. The scores of the high performers barely moved, interestingly. The movement and correlation for lowS1 is especially strong, improving a median 3 points, with a correlation of 0.371 and a p-value of 0.003. This effect can also be seen in the left half of Figure 3, which contains those students who performed poorly on Survey 1. For those students, if they continue to do poorly on Survey 2 (students in lower-left corner) they tend to have poor final exam scores, whereas students who improved on Survey 2 (students in upper-left corner) they tend to have better final exam scores.

In sum, most of the improvement in proficiency is due to the improvement of those with low entering proficiency, and that improvement had a significant correlation with their final score.

6 DISCUSSION

In this section we reflect on our research questions, results, limitations, and threats to validity.

6.1 Findings

The relatively low early-term prerequisite proficiency that we identified, combined with the correlation between proficiency and final exam performance, shows that many students will indeed be struggling. This is especially worrying as Muralidharan et al. [10] found that virtually no learning at the 8th grade math level is occurring for students who lack proficiency with the lower grade math materials. They found intervention with a focus on proficiency with lower grade math materials is required in order for students to make progress with their learning at the 8th grade math level.

In our study, we find that students do improve their prerequisite proficiency over the term, however they do so only marginally on average. But we also find that the amount of improvement in prerequisite proficiency is especially correlated to final exam performance for students with low early-term prerequisite proficiency.

One group we did not discuss was the 16 students who dropped the course (and hence are not included among our 208 participants). Their average early-term proficiency was 7 percentage points lower than those who finished the course, with 43% demonstrating low proficiency and only 12% high proficiency. While not statistically significant (likely due to the small sample size), this result suggests that prerequisite proficiency may be even lower at the start of the term than we found here. Moreover, it may indicate that lack of proficiency may be a reason for students dropping the course.

Given these findings, we suggest two possible courses of action: (1) final exams should be carefully designed to ensure that students who pass have achieved proficiency before proceeding to the next course (see discussion of course grades in the next section) and/or (2) teachers of upper-division computer science courses, especially those with many prerequisites, should consider helping students reinforce their prerequisite knowledge at the beginning of the term.

6.2 Why Students Lack Prerequisite Knowledge

Grades in Prerequisite Courses. In this study, we found that prerequisite proficiency in an upper-division Computer Science class is much lower than one might expect, given the students achieved passing grades in prerequisite courses. When we analyzed the failure rate of the prerequisite courses for Advanced Data Structures [1] we found that at our institution, on average, only 2.9% and 3.6% of students will receive a D or F (respectively) with the remainder of the students received a C or higher. (A 'D' grade presently allows students to progress at our institution). Our study highlights that not just 3.6% of students but likely 29% of students are failing in their prerequisite proficiency. We suspect there might be some form of grade inflation occurring and the consequence is that we are sending under-prepared students into subsequent classes.

Multiple Prerequisite Paths. At our institution, we offer non-majors options for prerequisite courses to provide flexibility. For example, non-majors can complete either a computer organization course that follows CS2, teaching C and assembly programming, or a non-major's CS1 course in C. Either course should teach students the necessary knowledge in C (e.g., pointers) to begin a course in C++, however one may do so in more detail. Additionally, one interdisciplinary major at our institution (with a large number of required units and few students) requires the Advanced Data Structures course but waives the Computer Organization prerequisite. As such, part of the effect seen in this study may be due to students bypassing prerequisites or due to some prerequisite courses being inferior to others as preparation for this course. Further inquiry is the topic of ongoing future work.

Unrealistic Expectations. Prior work in CS1 has shown that students often underperform instructor expectations. For example, instructors expect students completing CS1 to be able to complete the rainfall problem when, in reality, few students are so capable [16]. It is possible that this mismatch is again at play, where our survey questions on prerequisites were viewed as relatively easy

by the authors when in reality, they may be difficult for students. This underscores the importance of using meaningful assessments at the end of courses (e.g., concept inventories [17]) to measure student learning in order to better understand their abilities.

6.3 Limitations and Threats to Validity

Limitations. Generalizability from our study is limited as we only studied one course (ADS) and one term. Because correlation is not causation, the correlations found might not imply that increased prerequisite proficiency causes increased performance on the final. However, the fact that increased proficiency over the term was correlated with increased final exam performance for low- and high-performing groups provides compelling complementary evidence.

Underlying Factors. Although our survey focused on prerequisite knowledge, the relationship between the survey and the final exam may not be entirely related to prerequisite content. There may be intervening variables as perhaps prerequisite proficiency is related to students' general study habits or motivation in the field. Moreover, perhaps "better" students took the survey seriously and hence did better on the survey than those who did not. However, evidence that questions on particularly critical prerequisite topics were most strongly correlated with student exam performance suggests that at least some portion of the effect found related to the content itself.

Threats to Validity. The two surveys we used as a measuring tool are not validated and the questions were drawn from many different sources. Some survey questions may have been too hard for all students, making them less useful in our study. Although most students answered all questions, students were given a relatively short amount of time to complete them. As such, students may have hurried through the last questions and hence impacted findings. In addition, those who completed the survey were those who were in class on the two days the survey was given, so the results may be biased for the population of students who frequently attend class.

7 CONCLUSION

In our study of prerequisite proficiency in an upper-division advanced data structures course, we find that students varied in their proficiency to an unexpected degree, that prerequisite proficiency correlated with higher final exam scores, and that students improve on their understanding of prerequisites in a way that correlates to improved final exam scores. In sum, these findings support more care by instructors in ensuring students have fundamental proficiency with their course content before allowing them to move on to subsequent courses. In turn, instructors who teach courses with multiple prerequisites may wish to provide students with resources to buttress their mastery of prerequisites early in the term. Additional study is needed to understand whether these findings replicate across multiple institutions and across multiple courses, however, these results offer a promising glimpse of how prerequisites relate to student success in upper-division computing courses.

8 ACKNOWLEDGEMENTS

The authors wish to thank our reviewers for their helpful suggestions and the course instructor, Hau Nguyen, for his assistance. This work was supported in part by NSF award 1712508.

REFERENCES

- [1] Course and professor evaluations (cape). <http://cape.ucsd.edu/>. Accessed: 2018-11-05.
- [2] Prerequisite surveys. <https://paperdata.page.link/surveys>. Accessed: 2018-11-05.
- [3] A. D. Baddeley and A. D. Baddeley. *Your memory: A user's guide*. Carlton Books New York, NY, USA, 2004.
- [4] H. Danielsiek, W. Paul, and J. Vahrenhold. Detecting and understanding students' misconceptions related to algorithms and data structures. In *Proceedings of the 43rd Technical Symposium on Computer Science Education*, pages 21–26, 2012.
- [5] R. O. Duda, P. E. Hart, and N. J. Nilsson. Subjective bayesian methods for rule-based inference systems. In *Proceedings of the June 7-10, 1976, National Computer Conference and Exposition, AFIPS '76*, pages 1075–1082, 1976.
- [6] T. Götschi, I. Sanders, and V. Galpin. Mental models of recursion. SIGCSE '03, pages 346–350, New York, NY, USA, 2003. ACM.
- [7] S. Hamouda, S. H. Edwards, H. G. Elmongui, J. V. Ernst, and C. A. Shaffer. A basic recursion concept inventory. *Computer Science Education*, 27(2):121–148, 2017.
- [8] A. Lishinski, A. Yadav, J. Good, and R. Enbody. Learning to program: Gender differences and interactive effects of students' motivation, goals, and self-efficacy on performance. In *Proceedings of the 2016 Conference on International Computing Education Research*, pages 211–220, 2016.
- [9] M. A. Martin. Course prerequisites and undergraduate student performance. *NACTA Journal*, 33(1):38–42, 1989.
- [10] K. Muralidharan, A. Singh, and A. J. Ganimian. Disrupting education? experimental evidence on technology-aided instruction in india. Technical report, National Bureau of Economic Research, 2016.
- [11] S. E. Newstead, A. Franklyn-Stokes, and P. Armstead. Individual differences in student cheating. *Journal of Educational Psychology*, 88(2):229, 1996.
- [12] A.-C. J. T. F. on Computing Curricula. Computer science curricula 2013. 2013.
- [13] V. Ramalingam, D. LaBelle, and S. Wiedenbeck. Self-efficacy and mental models in learning to program. *SIGCSE Bulletin*, 36:171–175, 2004.
- [14] J. Sheard, A. Carbone, R. Lister, B. Simon, E. Thompson, and J. L. Whalley. Going solo to assess novice programmers. pages 209–213, 2008.
- [15] Simon and J. Sheard. Academic integrity and computing assessments. In *Proceedings of the Australasian Computer Science Week Multiconference*, pages 3:1–3:8, 2016.
- [16] E. Soloway. Learning to program= learning to construct mechanisms and explanations. *Communications of the ACM*, 29(9):850–858, 1986.
- [17] C. Taylor, D. Zingaro, L. Porter, K. C. Webb, C. B. Lee, and M. Clancy. Computer science concept inventories: past and future. *Computer Science Education*, 24(4):253–276, 2014.
- [18] J. Vitale, S. Wanger, and D. Adams. Explaining student performance in an undergraduate agricultural economics classroom. *NACTA Journal*, 54(1):2, 2010.
- [19] H. M. Walker. Prerequisites: Shaping the computing curriculum. *ACM Inroads*, 1(4):14–16, Dec. 2010.
- [20] C. Watson, F. W. Li, and J. L. Godwin. No tests required: Comparing traditional and dynamic predictors of programming success. In *Proceedings of the 45th Technical Symposium on Computer Science Education*, pages 469–474, 2014.
- [21] B. C. Wilson and S. Shrock. Contributing to success in an introductory computer science course: a study of twelve factors. *SIGCSE Bulletin*, 33:184–188, 2001.
- [22] D. Zingaro, M. Craig, L. Porter, B. A. Becker, Y. Cao, P. Conrad, D. Cukierman, A. Hellas, D. Loksa, and N. Thota. Achievement goals in CS1: Replication and extension. In *Proceedings of the 49th Technical Symposium on Computer Science Education*, pages 687–692, 2018.
- [23] D. Zingaro and L. Porter. Impact of student achievement goals on cs1 outcomes. In *Proceedings of the 47th Technical Symposium on Computing Science Education*, pages 279–296, 2016.