DOI: 10.1109/FIE.2014.7044349

An Examination of Mathematics Preparation For and Progress Through Three Introductory Computer Science Courses

Christine F. Reilly and Emmett Tomai
Computer Science Department
University of Texas – Pan American
Edinburg, Texas, USA
Email: reillycf@utpa.edu, tomaie@utpa.edu

Abstract—**The pipeline of three introductory computer science courses at our university have low pass rates. As part of our efforts to improve the pass rates in these courses, we examine the students' mathematics preparation and their progress from one course to the next. We find that the minority of students who enter the university with good mathematics preparation are likely to succeed in our introductory computer science courses. For the majority of students, we suggest a series of interventions throughout the pipeline. Prior to taking Computer Science 1, students could gain more experience with problem solving during the Introduction to Computer Science and Introduction to Computer Engineering courses. We suggest adding precalculus as a prerequisite for Computer Science 2. For students who do not earn an A in Computer Science 2, we plan to examine whether taking a class in a second programming language before taking Computer Science 3 will improve performance in Computer Science 3. Our goal is to improve student learning and retention through these efforts, combined with continual innovation in the classroom.**

I. Introduction

At our university, as is typical at many colleges and universities, the pass rates in our introductory Computer Science courses are relatively low. Our series of three introductory courses has an average pass rate in the low 60% range. This pass rate is in the range of that found at numerous other universities around the world [1], [2], [3], [4]. The authors of this paper, along with the other instructors of these courses at our university, are concerned by such a low pass rate and would like to investigate what we can do to help our students become more successful in our classes.

In the past, we have implemented efforts to improve student performance by engaging students through interesting programming assignments [5], providing frequent feedback to students by having a short quiz after each unit instead of only having three exams throughout the semester, and by encouraging students to see the instructor and teaching assistants for individual help, among other efforts.

This paper differs from those prior efforts by focusing on the preparation of students before they enter our introductory course pipeline, and on the progress of students through the pipeline. Building upon or prior work [6], we use data gathered from our university's Registrar's Office to obtain information regarding the mathematics preparation of our students and to examine the grades students receive in our series of introductory computer science courses. Through this data analysis, we hope to discover indicators that we can use when advising students regarding what preparation they should have for our introductory computer science courses.

Many prior studies have focused on mathematics background as a factor that impacts success in introductory programming courses. In one of these studies, a questionnaire was given to 130 students in a CS1 course in order to examine the factors that might contribute to success in CS1 [7]. It was found that comfort level in the class was the best predictor of success in the course, followed by a student's math background (as measured by the number of semesters of high school math taken by a student).

Another study took the college mathematics preparation of students into account by having students who enter the university with a mathematics placement below college algebra take a CS0 course prior to taking CS1 [8]. Their CS0 course introduces programming concepts using a media–rich programming language called Scratch. The students who took CS0 in the first semester of their freshman year had a higher retention rate in CS1 than the first semester students who had the required mathematics placement to take CS1 in the same semester.

In the early and middle part of the 1980's, a number of studies examined success in introductory computer science courses in relation to mathematics background. One of these studies found that the students who did not withdraw from an introductory computer science course were older, had better high school performance, had more previous computer science experience, and had more mathematics background than those students who withdrew from the course [9]. Another study followed the first–semester freshmen who were enrolled in an introductory computer science course through to the beginning of their sophomore year. At the beginning of their sophomore year, the students who were still enrolled in computer science and related majors had higher SAT math and verbal scores, higher high school rank, and stronger high school mathematics and science backgrounds than those students who had changed to other majors [10]. Another study from this time period [11] similarly found that high school GPA and ACT scores were the best predictive factor of success in computer science, followed by the completion of higher levels of math in high school.

In this paper, we examine two factors in relation to the success of our computer science and computer engineering students: the flow of students from one course to the next in our pipeline of three introductory computer science courses, and the relationship between mathematics preparation and success in our introductory computer science classes. Section II provides an introduction to our university and an overview of our introductory course pipeline. In Section III we examine the pass rates in our introductory computer science courses, and analyze how students perform in CS2 and CS3 based on their performance in the prior course in the pipeline. We then expand our analysis of success in introductory computer science courses to examine student grades with relation to their mathematics level in Section IV. Next, Section V provides a discussion of our findings, and we conclude in Section VI.

II. Computer Science and Computer Engineering at UTPA

The University of Texas–Pan American (UTPA) is a primarily undergraduate Hispanic–serving public university located in the lower Rio Grande Valley of Texas, USA. Because the university is located close to the US–Mexico border, it serves the rapidly growing population on both sides of the border. The university currently has an enrollment of over 20,000 students, and is continuing to grow.

The Department of Computer Science at UTPA offers degree programs at the Bachelor and Master's levels. Two undergraduate degrees are offered: the CAC/ABET accredited Bachelor of Science in Computer Science, and the EAC/ABET accredited Bachelor of Science in Computer Engineering. The Computer Engineering degree is jointly offered by the Departments of Computer Science and Electrical Engineering. Computer Engineering majors select either the software or hardware track, with some different course requirements for each track. We currently have between 500 and 600 undergraduate majors, split relatively evenly between computer science and computer engineering.

Our introductory computer science sequence consists of three courses that we refer to as a pipeline. There are also a few other courses students may during their first few semesters in the majors. The pipeline courses are Engineering Computer Science I (CS1), Computer Science II (CS2), and Algorithms and Data Structures (CS3). All computer science majors and computer engineering majors from both tracks are required to take these three courses.

- **CSCI/CMPE 1370 and 1170: Engineering Computer Science I:** This course covers the fundamentals of programming using C++. Topics covered include basic programming concepts, selection, looping, arrays, functions, structures, and an introduction to object oriented programming. Students must concurrently enroll for the accompanying laboratory course where they practice these programming techniques. The corequisites for CSCI/CMPE 1370/1170 is Introduction to Computer Science (for Computer Science majors) or Introduction to Computer Engineering (for Computer Engineering Majors), and College Algebra.
- **CSCI/CMPE 2380: Computer Science II:** This course focuses on object oriented programming in C++, and presents data structures such as linked lists and binary search trees. Students also learn algorithms for sorting and searching. The prerequisite for CSCI/CMPE 2380 is CSCI/CMPE 1370.
- **CSCI/CMPE 3333: Algorithms and Data Structures:** This course builds upon the data structures learned in 2380. It covers algorithmic efficiency and complexity, presents abstract data types such as graphs, networks, trees, and priority queues, and discusses searching and sorting. The prerequisites for CSCI/CMPE 3333 are CSCI/CMPE 2380, and either Discrete Structures (for Computer Science majors), or Math for Electrical and Computer Engineers (for Computer Engineering majors). CSCI/CMPE 3333 is the prerequisite for most upper–level computer science courses.

At the same time as taking the pipeline courses, students may take other computer science courses. Computer Science majors who have taken CS1 may take: Computer Organization and Assembly Language, Computer Programming in a Second Language, and Programming in the UNIX/Linux Environment. After taking CS2, Computer Science students may take Organization of Programming Languages. Computer Engineering majors take introductory electrical engineering courses along with the introductory computer science courses. While working on the computer science introductory pipeline courses, Computer Engineering majors may take the same additional computer science classes as Computer Science majors, with the exception of Programming in the UNIX/Linux environment and Organization of Programming Languages because these courses are not required for the Computer Engineering major.

III. Pass Rates in Our Introductory Computer Science Pipeline Courses

The faculty who teach the series of three introductory computer science courses at our university have noticed that these courses typically have a low pass rate. Note that because both the Computer Science and Computer Engineering majors require a minimum grade of a C in the core courses for the major, we define a passing grade as C or better. Additionally, the faculty who teach the pipeline courses have wondered whether the students who earn a C in one of the pipeline courses are adequately prepared to advance to the next course. We obtained data from our Registrar's Office for 558 students over a four and a half year period (Fall 2009 through Fall 2013) in order to examine the pass rates in detail.

As shown in Table I, the average pass rates for all sections of CS1 and CS2 are 60%, and the average pass rate for CS3 is 63%. As expected, there is high variance in pass rates from section to section and from term to term. However, the pass rates for each of the pipeline classes are very similar regardless of which instructor taught the section. For CS1, there were six different instructors during the study period. Three of the instructors only taught one section. Of the remaining instructors, two taught seven sections and one taught six sections during the study period. As shown in Table II, all three of these instructors have similar mean pass rates for the sections they taught. A pair–wise two–tailed T–test showed no significant difference between the pass rates for these three

TABLE I. PASS RATES IN CS PIPELINE COURSES

Course	Mean Pass Rate	Standard Deviation
CS1	60.3%	0.073
CS2	60.2%	0.104
CS3	63.4%	0.072

TABLE II. CS1 PASS RATES BY INSTRUCTOR

Instructor	Sections Taught	Mean Pass Rate	Standard Deviation
A	6	59.8%	0.074
B	7	58.0%	0.073
C	7	60.2%	0.071

TABLE III. CS2 PASS RATES BY INSTRUCTOR

Instructor	Sections Taught	Mean Pass Rate	Standard Deviation
D	4	61.7%	0.035
E	4	59.5%	0.074
F	9	58.9%	0.132

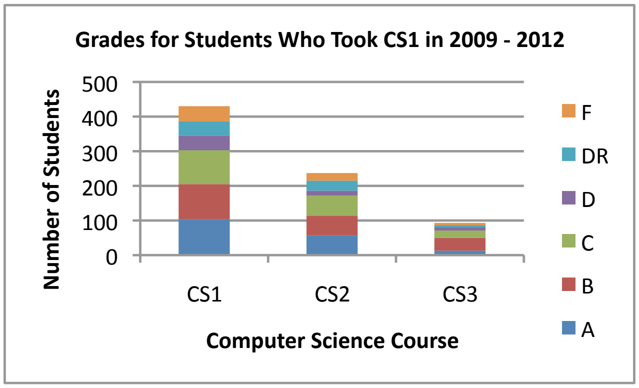

Fig. 1. Grade distribution in CS1, CS2, and CS3 for students who started CS1 during the 2009–2012 time period.

instructors ($p = 0.70$ for Instructor A versus Instructor B; $p = 0.62$ for Instructor B versus Instructor C; $p = 0.93$ for Instructor A versus Instructor C). For CS2, there were four different instructors during the study period, with one of these instructors only teaching one section. The three CS2 instructors who taught multiple sections had similar pass rates during the study period, as shown in Table III. A pair–wise two–tailed T–test showed no significant difference between the pass rates for these three instructors ($p = 0.67$ for Instructor D versus Instructor E; $p = 0.93$ for Instructor E versus Instructor F; $p = 0.60$ for Instructor D versus Instructor F). There was only one instructor for CS3 during the study period.

In order to examine the progress of students through the pipeline of introductory computer science courses, we first looked at the students who started CS1 in the 2009–2012 time period. Note that this is a subset of our data set that excludes the group of students with a reduced likelihood of having had the time to advance to CS3 during the data collection time period. There are 430 students in this subset.

Figure 1 shows the distributions of grades for the students who started CS1 during the 2009–2012 time period. The best grade each student achieved in each course is used, given the possibility that an individual student may have retaken a class multiple times. This graph shows that the CS1 pass rate during this period is 70%, the CS2 pass rate is 73%, and the CS3 pass rate is 76%. We also notice a significant attrition rate: 22% of the students who pass CS1 never took CS2 during this time period, and 50% of the students who pass CS2 never took CS3 during this time period. Table IV shows more details about this group of students, including the average number of retakes per student. From this table, we see that the number of retakes increases as students move through the pipeline.

Our next step, shown in Tables V and VI was to break the students who took CS1 and CS2 into groups based on the grade they received in each course. By grouping the students by their grade we can examine the pass rate in the next class based on the grade in the previous class.

Our data does show strikingly different pass rates in CS2 and CS3 based on the grade the student received in the previous course. We observe a 95% pass rate for the CS2 students who earned an A in CS1; a 75% pass rate for those who earned a B in CS1; and a 52% pass rate for those who earned a C in CS1. The pass rates for CS3 based on the grade in CS2 are similar: there is a 93% pass rate for CS3 students who earned an A in CS2; a pass rate of 71% for those who earned a B in CS2; and a 61% pass rate for those who earned a C in CS2.

We also used the data in Tables V and VI to examine the attrition rate based on the grade earned in a course. We see that the attrition rate increases as the grade decreases, but also notice a significant attrition rate for students who earn A's and B's.

After a few more semesters we plan to reexamine the attrition rates, especially for CS2 to CS3. It is possible that more of the students who passed CS2 during the time period examined in this study will take CS3. Our CS3 course has a prerequisite of Discrete Math, which has a prerequisite of Calculus I. Because CS1 has a corequisite of College Algebra, it is possible that a student who took CS1 in Spring 2012 has not been able to take CS3 before Spring 2014 (the end of our current data collection) due to the math prerequisite.

IV. MATH PREPAREDNESS AND SUCCESS IN OUR INTRODUCTORY COMPUTER SCIENCE PIPELINE COURSES

Because prior studies have shown that math preparedness has an impact on student success in introductory computer science courses, we examined the grades students received in our introductory pipeline courses in relation to their math level. We chose to use the students' current math level at our university as a measure of math preparedness because we could get this information from our Registrar's office and this information is readily available when we are advising students.

We looked at the students' initial math placement upon entering the university, and at their progression through the math courses at the university. A student's initial math placement can be considered as a measure of pre–university preparedness. We considered three categories of initial math placement: ready for college algebra or lower placement (alg-), ready for pre-calculus (pc), and ready for calculus or higher (cal+). We also considered the students' progress through math courses since they entered UTPA: placed at college algebra or lower, then progressed to ready for pre-calculus (alg- $\rightarrow$ pc); placed into college algebra or lower, then progressed to ready for calculus or higher (alg- $\rightarrow$ cal+); and placed into precalculus, then progressed to ready for calculus or higher (pc $\rightarrow$ cal+).

TABLE IV. STUDENTS WHO STARTED CS1 DURING 2009–2012 TIME PERIOD

Course	Number of Students	Attrition	Number Passed	Percent Passed	Average Number of Retakes per Student
CS1	430	N/A	303	70%	0.13
CS2	237	22%	172	73%	0.16
CS3	93	50%	71	76%	0.22

TABLE V. PASS RATE IN CS2 BASED ON GRADE IN CS1

CS1 Grade	Number of Students	Number Took CS2	Attrition	Number Passed	Percent Passed
A	133	114	14%	108	95%
B	113	88	22%	66	75%
C	100	60	40%	31	52%

TABLE VI. PASS RATE IN CS3 BASED ON GRADE IN CS2

CS2 Grade	Number of Students	Number Took CS3	Attrition	Number Passed	Percent Passed
A	57	40	30%	37	93%
B	57	28	51%	20	71%
C	58	23	60%	14	61%

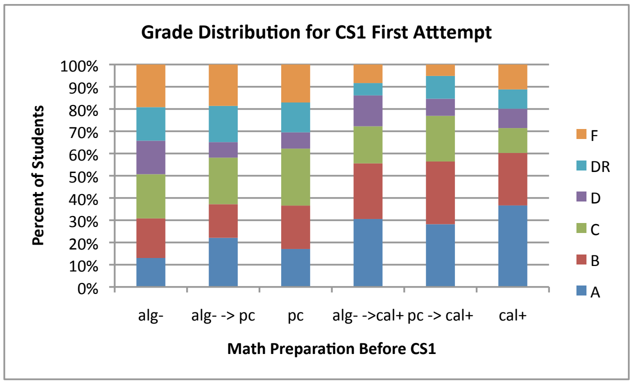

Fig. 2. Grade distribution for first attempt at CS1, grouped by math preparation

We asked two questions regarding math preparedness: Does math preparedness at the time of taking CS1 predict success in CS1, CS2, and CS3? Does math preparedness at the time of CS2 predict success in CS2 and CS3?

A. Math Preparedness at Time of Taking CS1

Figure 2 shows the grade distribution for the students first attempt at taking CS1, grouped by their math preparation category at the time of taking CS1. The corresponding pass rates are shown in Table VII. From Figure 2 and Table VII, we see that students who are ready for calculus or a higher math course in the same semester when they take CS1 have the highest first–try pass rates in CS1.

Next, in order to facilitate our data analysis, we further grouped the students into the following groups: not ready for calculus in the same semester as they first took CS1, (the alg-, alg- $\rightarrow$ pc, and pc groups; abbreviated "not cal ready"), entered the university ready for calculus (the cal+ group; abbreviated "cal ready"), and became ready for calculus while at UTPA (the alg- $\rightarrow$ cal+, and pc $\rightarrow$ cal+ groups; abbreviated "got cal ready"). The distribution of first–try CS1 grades for these three groups are shown in Figure 3.

We see that regardless of initial math placement when a student enters the university, the students who are calculus ready in the same semester when they first take CS1 have a higher pass rate than those students who are not ready for calculus when they first take CS1. However, when we look at the total number of students in each group, the students who are not calculus ready represent the majority of students who pass CS1 on the first try, and have approximately the same number of A's and B's as the students who are ready for calculus. This is shown in Figure 4 and Table VIII. The difference in pass rates, and the difference in the combined A and B grades, shown in Table VIII, is significant (chi–square, $p < 0.01$).

When we look at the math preparation of the students when they first took CS1 along with the best grades they earned in CS2, we see similar trends as the grades earned in CS1, as shown in Figure 5 and Table IX. We see a higher pass rate in CS2 for students who entered the university ready to take calculus, than those students who did not become ready for calculus prior to taking CS1. However, when looking at the total number of students in each group, we see that approximately the same number of students who were not calculus ready when taking CS1 for the first time and the group that entered the university ready for calculus pass CS2. The difference in pass rates, and the difference in the combined A and B grades, shown in Table IX, is significant (chi–square, $p < 0.01$).

By the time the students reach CS3, there are approximately the same number of students in the groups who entered the university ready for calculus and those who were not calculus ready when they took CS1. The group of students who became calculus ready prior to taking CS1 has now shrunk to 13 students, making it difficult to draw any conclusions regarding this group. Figure 6 and Table X show that the students who entered the university ready for calculus still have a higher pass rate in CS3 (85%), compared with the 78% pass rate for students who were not calculus ready when they took CS1. The small size of the group of students who became calculus ready prior to taking CS1 is too small for a reliable chi–square test. If we combine the groups of students who entered the university ready for calculus and the group who became ready for calculus prior to taking CS1 we can perform a chi–square test, and find no significant difference in the pass rates between the group of students who were calculus ready prior to taking CS1 and those that were not calculus ready (chi–square, $p = 0.82$).

Our results show that being prepared to take Calculus I in the same semester as a student takes CS1 does predict success in CS1 and CS2. However, there are still a large number of students who were not calculus ready when taking CS1 who do pass CS1 and CS2. This indicates that increasing the math prerequisite for CS1 may improve the pass rates for CS1 and CS2, but would also cause a large number of students who are likely to pass the classes to be delayed in starting the introductory computer science pipeline. It would better serve our students for us to examine other factors that would better prepare them for CS1 and CS2. A student's math preparedness at the time of taking CS1 does not have an impact on their performance in CS3.

TABLE VII. PASS RATE IN CS1 GROUPED BY MATH PREPARATION

Math Preparation Level	Math Preparation Abbreviation	CS1 First Try Pass Rate
Placed at College Algebra or lower	alg-	51%
Placed at College Algebra or lower, advanced to ready for precalculus	alg- ← pc	58%
Placed at precalculus	pc	62%
Placed at College Algebra or lower, advanced to ready for calculus	alg- ← cal+	72%
Placed at Precalculus, advanced to ready for calculus	pc ← cal+	77%
Placed at Calculus or higher	cal+	71%

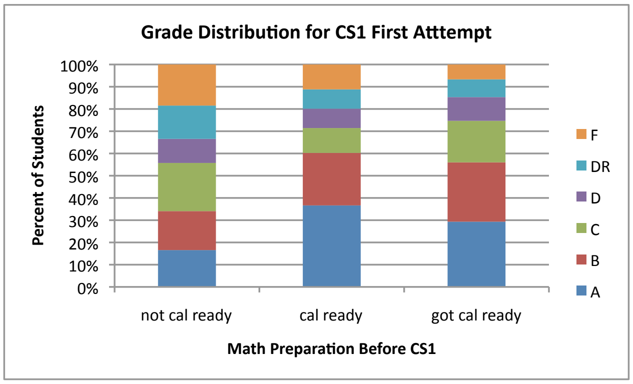

Fig. 3. Grade distribution for first attempt at CS1, grouped by math preparation

B. Math Preparedness at the Time of Taking CS2

We then looked at the students' math preparedness at the time when they took CS2. Figures 7 and 8 and Table XI show the grades in CS2 based on the students' math level in the semester when they first take CS2. We see that the majority of students are ready for calculus or a higher math course in the semester when they take CS2, and that this majority is much more likely to pass CS2 than the students who are not calculus ready. The difference in pass rates, and the difference in the combined A and B grades, shown in Table XI, is significant (chi–square, $p < 0.01$).

In Figure 9 and Table XII we show the grade distribution in CS3 based on the students' math placement on their first attempt on taking CS2. The pass rates for CS3 are 73% of the students who were not calculus ready when taking CS2 passed CS3, 85% of the students who placed into the university as ready for calculus passed CS3, and 68% of the students who became ready for calculus by the time they took CS2 passed CS3 (note that there are only 15 students in this last group). The small size of the group of students who became calculus ready prior to taking CS2 is too small for a reliable chi–square test. If we combine the groups of students who entered the university ready for calculus and the group who became ready for calculus prior to taking CS2 we can perform a chi–square test, and find no significant difference in the pass rates between the group of students who were calculus ready prior to taking CS2 and those that were not calculus ready (chi–square, $p = 0.76$).

An interesting observation regarding students who take CS2 and CS3 is that the ones who entered the university ready to take calculus have better performance, based on pass rate and grades earned, than those who became ready to take calculus while at UTPA. Entering the university with the necessary background to take calculus is an indicator of good academic preparation at the high school level.

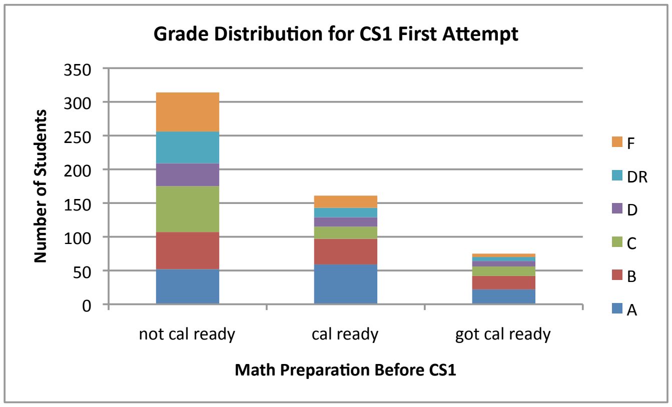

Fig. 4. Total number of first–try CS1 students, grouped by math preparation

TABLE VIII. FIRST–TRY CS1 STUDENTS, GROUPED BY CS1 MATH PREPARATION

Math Prep Abbreviation	Total Students	Number of A's and B's	Number Passed	Number Failed	Percent Passed
not cal ready	314	107	175	139	56%
cal ready	161	97	115	46	71%
got cal ready	75	42	56	19	75%

We also observe that students who were not ready for calculus at the time they take CS2 are the group with the poorest performance in CS2 and CS3. Perhaps CS2 is the point where we should examine adding a prerequisite of Calculus I.

C. Progression from CS1 to CS2

Next, we examine the progression of students from CS1 to CS2. Figure 10 and Table XIII show the numbers of students who pass and fail CS2, based on their best grade earned in the course, given the fact that some students take CS2 multiple times. We first show the number who pass/fail CS2 based on CS1 grade for all math preparation groups, then split the students into their math preparation group at the time when they first took CS2.

Students earning A and B grades in CS1 had 95% and 70% pass rates in CS2, respectively, while students earning C grades in CS1 had a CS2 pass rate of only 52%. Breaking this out further by math preparedness, the sample sizes are too small for a reliable chi–square test. Instead, we perform a Fisher's exact test for the pass/fail outcome for students who received a C in CS1. We find that only in the group of students who was not ready for calculus when they took CS1 is there a significantly lower pass rate in CS2 ($p = 0.01$).

Because of the relatively small number of students who take CS3, we are not able to examine the pass rates in CS3 based on grade in CS2 and math preparedness.

V. DISCUSSION

In Section III we showed that the overall pass rates in our introductory computer science courses are relatively low. We then examined the pass rates in CS2 based on the grade a

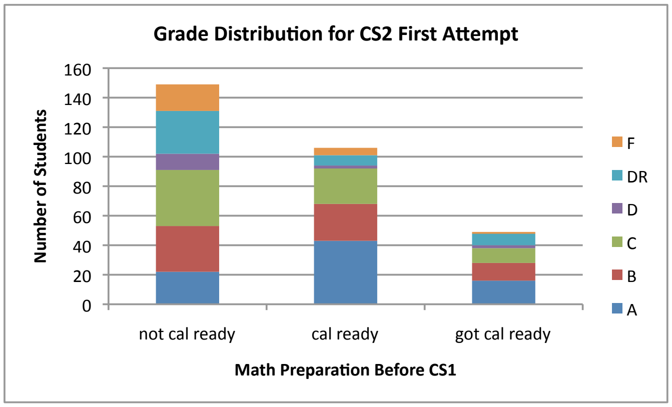

Fig. 5. CS2 grades based on math preparation at CS1 first–try

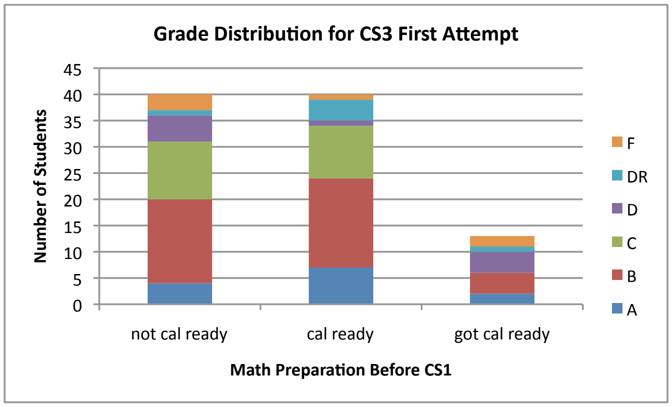

Fig. 6. CS3 grades based on math preparation at CS1 first–try.

TABLE IX. FIRST–TRY CS2 STUDENTS, GROUPED BY CS1 MATH PREPARATION

Math Prep Abbreviation	Total Students	Number of A's and B's	Number Passed	Number Failed	Percent Passed
not cal ready	149	53	91	58	61%
cal ready	106	68	92	14	87%
got cal ready	49	28	38	11	76%

TABLE X. FIRST–TRY CS3 STUDENTS, GROUPED BY CS1 MATH PREPARATION

Math Prep Abbreviation	Total Students	Number of A's and B's	Number Passed	Number Failed	Percent Passed
not cal ready	40	20	31	9	76%
cal ready	40	24	34	6	85%
got cal ready	49	28	38	11	76%

student received in CS1, and the pass rates in CS3 based on the grade a student received in CS2. As we expected, we found that students who earn a C in one course are less likely to pass the next course.

Because prior work has indicated that mathematics background and ability may be a factor in determining which students will be successful in computer science, in Section IV we examined whether a student's math preparation is related to the pass rate in our introductory computer science pipeline classes. Our reasoning is that perhaps a change in the math prerequisites for our courses would ensure that the students are better prepared and more likely to succeed in the courses.

A. *Preparation for Computer Science 1*

For CS1, we did find that the students who were ready to take calculus in the same semester as they took CS1 had a higher pass rate than the non–calculus ready students. However, the majority of our CS1 students are not ready for calculus when they take CS1, and the number of these students who pass CS1 approximately matches the number of calculus–ready students who pass CS1. Because so many non–calculus ready students are passing CS1, changing the CS1 prerequisite to precalculus does not seem like the correct approach for improving student success in CS1.

We suggest that the more constructive way to improve the pass rate in CS1 is to help students determine whether they are truly interested in and motivated for being a computer science or computer engineering major, and to help students gain additional skills prior to taking CS1. Both computer science and computer engineering have an introductory class that students should take prior to taking CS1. Perhaps we could use this introductory class to better engage and motivate students for their major. Another idea is to examine whether students who are calculus–ready have better problem solving or analytical reasoning abilities than those students who are not calculus–ready. If this is the case, perhaps adding a problem solving unit to the introductory courses would help better prepare all students for CS1.

B. *Preparation for Computer Science 2*

Most students are calculus–ready by the time they reach CS2. The students who are not calculus–ready at this point have a much lower pass rate than the calculus–ready students. Perhaps adding a prerequisite of precalculus to CS2 would be beneficial.

There are a relatively small number of students who pass CS2 but were not calculus–ready when they took CS2. We wondered whether these students were behind in math due to failing math courses or due to simply not taking the math courses. We looked into this group of students in more detail and found that only 5 of the 31 students who were not calculus–ready when they took CS2, and passed CS2 on their first attempt, had failed a college algebra or precalculus course. Therefore, we think that most of the students who are not calculus–ready at the point of taking CS2 could have become calculus–ready by this semester if they had adjusted their schedule to take math earlier in their time at the university. Because the prerequisite for CS3 is a discrete math course that has calculus I as a prerequisite, it would be beneficial to the students' schedules to be calculus–ready by the time they take CS2.

C. *Preparation for Computer Science 3*

Because all students should have taken calculus I by the time they reach CS3, we need to examine methods other than math preparation for improving student success in CS3. In Table VI we saw that the pass rate in CS3 decreases as the CS2 grades decrease. We plan to examine whether additional education on CS2–type topics prior to taking CS3 would improve the pass rate in CS3.

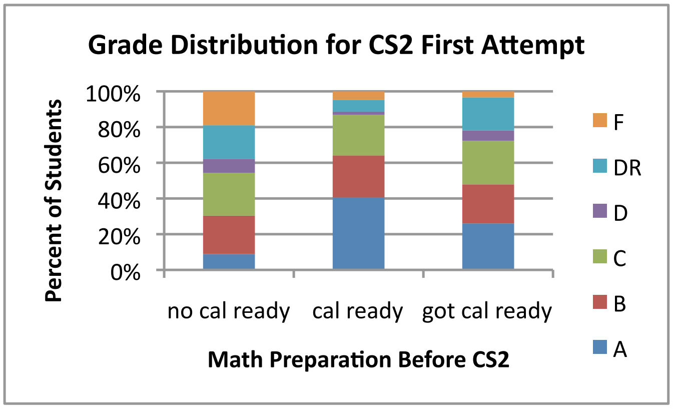

Fig. 7. CS2 grades based on math preparation at CS2 first–try

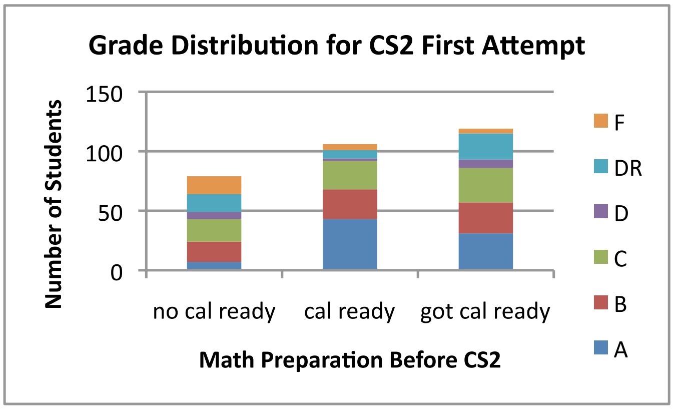

Fig. 8. CS2 grades based on math preparation at CS2 first–try

All Computer Science majors and the software–track Computer Engineering majors are required to take a class in a second programming language. Currently, we use C++ in CS1 and CS2, and we offer Java or C# as the choices for the second programming language class. The Java and C# classes focus on object oriented programming and basic data structures, and have a prerequisite of CS1. Most students take the second programming language class after CS2, although there are some that have not completed CS2 prior to taking the second programming language course. At what point in the course sequence to take the second programming language course is largely up to the student's discretion.

Perhaps we could better utilize the second programming language class as an opportunity for students to gain more experience with programming and basic data structures prior to taking CS3. We could change the prerequisite for the second programming language class from CS1 to CS2 in order to ensure that all students in the second programming language class have knowledge of basic data structures. Then we could advise students who earn a B or C in CS2 to take the second programming language course prior to taking CS3.

There are a couple of drawbacks to using the second programming language course as a bridge between CS2 and CS3. One is that the hardware–track Computer Engineering majors are required to take CS3, but are not required to take a second programming language. Another drawback is that we offer Java and C# in alternating semesters. If a student has a strong preference for one of these languages and desires to gain the additional programming experience prior to taking CS3, they may delay the semester when they take CS3 so that they can take the second programming language of their choice. This will cause additional delays in taking the higher level courses that have CS3 as a prerequisite.

TABLE XI. FIRST–TRY CS2 STUDENTS, GROUPED BY CS2 MATH PREPARATION

Math Prep Abbreviation	Total Students	Number of A's and B's	Number Passed	Number Failed	Percent Passed
not cal ready	79	24	43	36	54%
cal ready	106	68	92	14	87%
got cal ready	119	57	86	33	72%

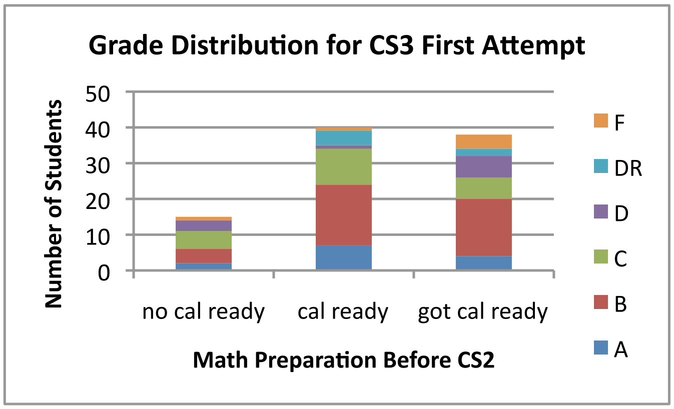

Fig. 9. CS3 grades based on math preparation at CS2 first–try.

D. Attrition Rate

Prior to conducting the data analysis described in Section III, we had not thought deeply about the attrition rate in our introductory computer science course pipeline. After observing the attrition rates shown in Tables V, VI, and XIII, we see the opportunity and need to do more internal recruiting. We were especially surprised by the number of students who earn A's and B's, but do not progress to the next class in the pipeline. We plan to analyze additional data to see whether these students are staying at UTPA and changing majors, transferring to a different university, or not continuing with their higher education. We note that the six–year graduation rate at UTPA is 46% and acknowledge that the attrition rate we observe in our classes may reflect that of the university as a whole.

VI. CONCLUSIONS

By analyzing the grades our students earn in the introductory computer science pipeline courses along with their math preparation at the time when they take CS1 and CS2 we have confirmed that both math preparation and the grade earned in the courses as a student progresses through the pipeline have an impact on student success in the introductory computer science courses. The minority of our students who enter the university with better academic preparation, as indicated by being ready to take calculus when they enter the university, have the best performance in the computer science pipeline courses. We plan to pursue mechanisms to help better prepare the majority of

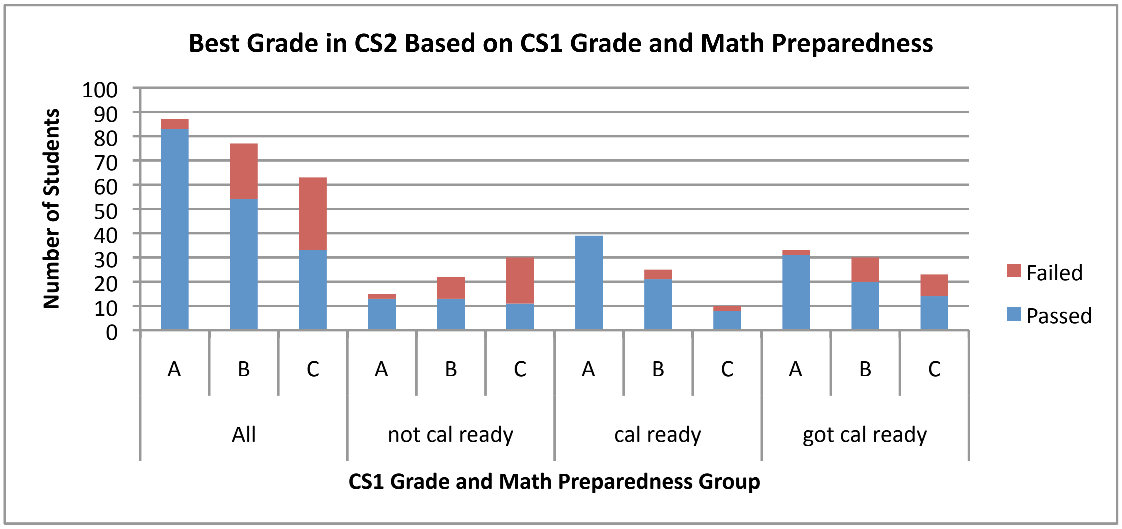

Fig. 10. Best grade pass rate in CS2 based on CS1 grade and math preparedness when starting CS2.

TABLE XII. FIRST–TRY CS3 STUDENTS, GROUPED BY CS2 MATH PREPARATION

Math Prep Abbreviation	Total Students	Number of A's and B's	Number Passed	Number Failed	Percent Passed
not cal ready	15	6	11	4	73%
cal ready	40	24	34	6	85%
got cal ready	38	20	26	12	68%

TABLE XIII. BEST CS2 GRADE BASED ON CS1 GRADE AND CS2 MATH PREPARATION

Math Prep Abbreviation	CS1 Grade	Number Passed	Number Failed	Percent Passed	Attrition Rate
All	A	83	4	95%	16%
	B	54	23	70%	24%
	C	33	30	52%	36%
not cal ready	A	13	2	87%	42%
	B	13	9	59%	41%
	C	11	19	37%	44%
cal ready	A	39	0	100%	11%
	B	21	4	84%	19%
	C	8	2	80%	41%
got cal ready	A	31	2	94%	3%
	B	20	10	67%	9%
	C	14	9	61%	15%

our students for the pipeline of introductory computer science courses.

We will examine whether we can better use the Introduction to Computer Science and Introduction to Computer Engineering courses to prepare students for CS1. One idea is to add material that will help students improve their problem solving and analytical thinking skills. We can also use these courses to better engage the students and increase their interest and motivation in becoming a Computer Scientist or Computer Engineer.

Our results indicate that adding a prerequisite of precalculus to CS2 may improve student success in CS2. In any case, this prerequisite would provide more encouragement for students to plan math classes into their schedule such that they meet the math prerequisite for CS3 in a timely manner.

Students who earn an A in CS2 appear to be well prepared for CS3. For students who earn a B in CS2, and especially for those who earn a C in CS2, we plan to examine whether taking the second programming language class between CS2 and CS3 could help improve their success in CS3.

This study brought to light the issue of the attrition rate in our introductory computer science pipeline courses, especially that of the students who perform well in the courses. We will definitely examine this issue in greater detail and hope to find ways to retain more of the top performing students in our majors.

In addition to the changes we suggest based on placement and ordering of classes, the faculty who teach our introductory computer science courses continue to learn about innovative classroom techniques. Through a combination of the changes suggested in this paper, and continual innovation in the classroom, we aim to improve the learning experience and outcomes for our students.

REFERENCES

[1] J. Bennedsen and M. E. Capersen, "Failure rates in introductory programming," *SIGCSE Bulletin*, vol. 39, no. 2, pp. 32–36, Jun. 2007.

[2] M. Guzdial, "Education teaching computing to everyone," *Communications of the ACM*, vol. 52, no. 5, pp. 31–33, May 2009.

[3] L. Porter, M. Guzdial, C. McDowell, and B. Simon, "Education success in introductory programming: What works?" *Communications of the ACM*, vol. 56, no. 8, pp. 34–36, Aug. 2013.

[4] A. Robins, "Learning edge momentum: A new account of outcomes in cs1," *Computer Science Education*, vol. 20, no. 1, pp. 37–71, Mar. 2010.

[5] C. F. Reilly and N. D. L. Mora, "The impact of real–world topic labs student performance in CS1," in *FIE '12 Proceedings of the 2012 IEEE Frontiers in Education Conference*, Seattle, Washington, USA, Oct. 2012.

[6] C. Reilly and E. Tomai, "Should college algebra be a corequisite for computer science 1?" in *Proceedings of the 2013 International Conference on Frontiers in Education: Computer Science and Computer Engineering*, Las Vegas, Nevada, USA, Jul. 2013.

[7] B. C. Wilson and S. Shrock, "Contributing to success in an introductory computer science course: A study of twelve factors," in *SIGCSE 2001*, Charlotte, North Carolina, USA, 2001.

[8] M. Rizvi and T. Humphries, “A scratch–based cs0 course for at–risk computer science majors,” in *FIE ’12 Proceedings of the 2012 IEEE Frontiers in Education Conference*, Seattle, Washington, USA, Oct. 2012.

[9] J. Konvalina, S. A. Wileman, and L. J. Stephens, “Math proficiency: A key to success for computer science students,” *Communications of the ACM*, vol. 26, no. 5, pp. 377–382, May 1983.

[10] P. F. Campbell and G. P. McCabe, “Predicting the success of freshmen in a computer science major,” *Communications of the ACM*, vol. 27, no. 11, pp. 1108–1113, Nov. 1984.

[11] D. F. Butcher and W. A. Muth, “Predicting performance in an introductory computer science course,” *Communications of the ACM*, vol. 28, no. 3, pp. 263–268, Mar. 1985.