Predicting Student Success in Computer Science – A Reproducibility Study

Amruth N. Kumar
Ramapo College of New Jersey
Mahwah, NJ 07430, USA
amruth@ramapo.edu

Abstract **(Research Category/Full paper) — A recent study conducted at the University of Oklahoma, a large research university attempted to use the grades on initial Computer Science courses to predict the success of Computer Science majors. We attempted to reproduce this study in a mid-sized liberal arts institution. We analyzed 15 years of data of students (majors as well as non-Computer Science majors) who had taken introductory Computer Science courses. We found that the better the grade on *Computer Science I*, the introductory course in the major, the better the cumulative GPA of the student upon graduation, and this applied to Computer Science majors as well as non-majors. All the students who had successfully graduated with a Computer Science degree had earned at least a C grade in the first three required courses: *Computer Science I*, *Computer Science II* and *Data Structures.* When we considered grades on six of the required courses in the Computer Science sequence, we found that students generally earned the same or lower grade on each subsequent course. Therefore, the performance of Computer Science majors on the first three courses in the required course sequence can reasonably be used as predictors of their success in the major. Finally, we found that Math SAT score was a good predictor of student success in *Computer Science I* as well as obtaining an undergraduate degree regardless of the major. Our study generalizes the results of the previous study and strengthens the results by finding that they are statistically significant. (*Abstract*)**

Keywords—Computer Science enrollment management, Predictors of success, Computer Science I

I. Introduction

Enrollment in Computer Science has been surging since 2006 [2]. This has resulted in over-subscription of Computer Science courses and shortfall of instructional staff in institutions large and small. The promise of lucrative career prospects after graduation has drawn a lot of students into the major, some with better aptitude for problem-solving that is integral to Computer Science than others.

In this context, recently, education researchers at the University of Oklahoma used transcript analysis to determine predictors of student success in Computer Science [1]. They analyzed four years of enrollment data and found that students earned higher grades in the first course than in subsequent Computer Science courses; no graduating Computer Science major had received a C in both the first two programming courses; and that GPA data used in the past to filter out students was a poor predictor of success in Computer Science.

We decided to reproduce this study at our institution for two reasons: 1) we wanted to see if we could find similar patterns in our data; and 2) we wanted to check whether the conclusions of the earlier study conducted in a large research institution extended to a mid-sized liberal arts college. Whereas joint probability distributions were used for analysis in the prior study [1], we chose to use ANOVA so that we could find the statistical significance of the results, if any.

II. Data Collection and Analysis

For this study, we collected data from the Office of Institutional Research for all the semesters from fall 2002 through spring 2017. We collected data of all the students who had taken *Computer Science I.* This is the introductory Computer Science course taken by all fresh admit Computer Science majors. It also qualifies for General Education credit. So, our data included Computer Science majors as well as non-majors.

The data we collected included:

- Demographic data: Sex (Male/Female), Race (Asian / American Indian / Black / Caucasian / Hispanic / Hawaiian / Multiracial / Unknown), and whether a first-generation student (true / false / unknown).
- Prior performance data: SAT Reading score, SAT Math score, and High school GPA where available.
- Academic data: major (grouped as Computer Science, Mathematics, Other Sciences, and non-Science majors) and the grade and the instructor with whom the student took each of the following courses – *Computer Science I*, *Computer Science II*, *Data Structures, Operating Systems, Software Engineering,* and *Organization of Programming Languages.* All these courses are required for Computer Science majors. Most non-Computer Science majors took only *Computer Science I.*
- Outcome data: Number of quarters taken to graduate, and cumulative GPA upon graduation.

In all, 2376 students had taken *Computer Science I* in those years: 881 female and 1495 male. Of these, 103 were

Computer Science majors, 205 Math majors, 404 Science majors (e.g., Biology, Physics, Chemistry, Environmental Science) and 1264 non-Science majors (e.g., Humanities, Business, Arts majors). Of these, 1369 had graduated.

For evaluation purposes, we ignored sign grades, i.e., A- was treated the same as A, B+, B and B- were all treated the same, and so on. We numerically coded A as 4.0, B as 3.0, C as 2.0, D as 1.0 and F as 0. Finally, we did not include W (withdraw), I (Incomplete), AU (Audit), DR (Dropped) and P (Pass/Fail) in the analysis since students with these grades either did not complete the course or were not subjected to the same grading criteria as the others in the course. By discounting W and DR grades, our analysis does not completely account for the significant attrition rate in *Computer Science I* (e.g., [3]). But, attrition in *Computer Science I* was not in the scope of the current study.

A. *Cumulative GPA and Grade in Computer Science I*

One-way ANOVA of cumulative GPA with the grade on *Computer Science I* as the fixed factor, considering only the students who had already graduated, yielded a significant main effect: [$F(4,1137) = 117.07$, $p < 0.001$]. Post-hoc test yielded a significant difference between all the groups except occasionally the group that failed *Computer Science I.* Table I lists the number of students who scored each grade on *Computer Science I* and their mean cumulative GPA with 95% confidence interval. Note that *the better the grade on Computer Science I, the greater the mean cumulative GPA of the cohort upon graduation*.

TABLE I. CUMULATIVE GRADE VERSUS GRADE ON COMPUTER SCIENCE I

CS I Grade	N	Cumulative Grade
F	88	2.755 ± 0.0745
D	133	2.759 ± 0.0628
C	265	2.936 ± 0.0459
B	272	3.160 ± 0.0487
A	384	3.431 ± 0.0435

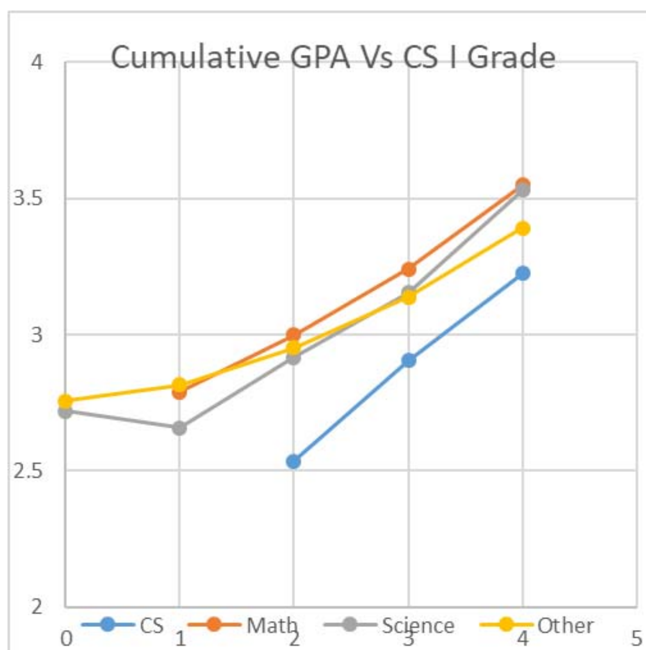
We repeated the same ANOVA analysis individually for each major and found a similarly significant main effect for Computer Science [$F(3,98) = 5.608$, $p = 0.001$], Mathematics [$F(4,200) = 26.129$, $p < 0.001$], Other Sciences [$F(4,374) = 58.538$, $p < 0.001$] and non-Science majors [$F(4,447) = 36.213$, $p < 0.001$]. Figure 1 plots the cumulative GPA against the grade on *Computer Science I* for all four majors. Whenever post-hoc tests were permitted, the differences between most pairs of groups were significant except occasionally for the groups that scored F or D. As noted before*,* the better the grade on *Computer Science I*, the better the cumulative GPA upon graduation*.* This applied not only to Computer Science majors, for whom *Computer Science I* is a mandatory entry-level course, *but also to all other majors, including non-Science majors*.

Remarkably, 82.35% of Computer Science majors who graduated earned an A in *Computer Science I,* and 10.78% earned a B grade. Only 6.86% of the majors who graduated earned a C or a D. So, *most students who successfully graduate with a Computer Science degree earn at least a B in Computer Science I.* This not only confirms the results from the earlier study [2], but also generalizes it since we did not use pair programming or separate students in *Computer Science I* by prior programming experience as was done in the earlier study.

As for checking the converse, i.e., whether any Computer Science major who earned a B or better in *Computer Science I* fail to complete Computer Science degree, our data was incomplete: cumulative GPA and number of quarters taken to graduate were reported only for those who had graduated, and not for those who had dropped out. The data recorded only the current major of a student, not a prior major that the student may have declared when entering the college.

Computer Science I at our institution had been taught by 12 different instructors during the time of this study. While all the instructors cover the same content, the approaches they take, the number and type of projects they assign, and the nature of the exams they administer are left to the discretion of each instructor. We found statistically significant difference in the grades assigned in the course by the 12 instructors [$F(11,2060) = 7.606$, $p < 0.001$]. During the period of this study, the course also went from a 3-credit lecture format to a 4-credit closed lab format (e.g., [4]). The significant effect of *Computer Science I* grade on cumulative GPA is in spite of these variations among the offerings of the course, and generalizes the result to schools where multiple instructors teach *Computer Science I.*

Figure 1: Cumulative GPA versus Grade on *Computer Science I* for students majoring in Computer Science, Math, Other Sciences and non-Science majors



B. Cumulative GPA and Grades on Other Required Computer Science Courses

Our Computer Science curriculum contains 8 required courses, 6 of which are usually taken in sequence: *Computer Science I (CS I), Computer Science II (CS II), Data Structures (DS), Operating Systems (OS), Software Engineering (SE)* and *Organization of Programming Languages (OPL).*

We repeated one-way ANOVA of cumulative GPA for Computer Science majors who had already graduated, with grades on the five other required Computer Science courses as fixed factors. Whereas each of these courses contributes to cumulative GPA, cumulative GPA also contains grades on 2 other required courses, 7 electives, 4 Math courses and 60 credits of General Education courses.

For each of the required courses, we found that *the better the grade on the course, the better the cumulative GPA*:

- *Computer Science II (CS II):* [$F(3,99) = 7.412$, $p < 0.001$]. This course was taught by 8 different instructors during the period of this study. However, no statistically significant difference was found in the course grade among the instructors [$F(7,568) = 1.698$, $p = 0.107$].
- *Data Structures (DS):* [$F(2,96) = 18.797$, $p < 0.001$]. Even though this course was taught by 4 different instructors, no statistically significant difference was found in the grades awarded by them [$F(3,379) = 1.885$, $p = 0.132$]
- *Operating Systems (OS):* [$F(3,96) = 31.313$, $p < 0.001$].
- *Software Engineering (SE):* [$F(3,94] = 13.028$, $p < 0.001$].
- *Organization of Programming Languages (OPL):* [$F(4,84) = 24.609$, $p < 0.001$].

Table II lists the percentage of the students who graduated with Computer Science degree, who earned each of A/B/C/D/F grade on the six required courses. Note that those who earned an F repeated the course at another institution and transferred the credits in order to subsequently graduate. Their transferred grade is not included in the analysis.

TABLE II. THE PERCENTAGE OF STUDENTS WHO GRADUATED WITH COMPUTER SCIENCE DEGREE THAT EARNED EACH GRADE IN THE REQUIRED COURSES

	A	B	C	D	F
CS I	82.35	10.78	5.88	1.0	
CS II	54.00	32.00	13.00		1.0
DS	45.45	39.39	14.14		
OS	29.00	30.00	34.00	7.00	
SE	28.57	47.95	22.44	1.02	
OPL	19.10	31.46	31.46	12.35	5.61

From Table II, it is clear that a minimum grade of C in *Computer Science I, Computer Science II* and *Data Structures was required to successfully graduate with a degree in Computer Science.* This analysis considered only the students who successfully graduated with a Computer Science degree. Students who perform poorly in the early courses of Computer Science typically transfer to Information Systems major offered by business school. Their data was not included in the current analysis, but was used in the analysis of the previous section.

C. Pairwise Comparsion of Grades on Required Courses

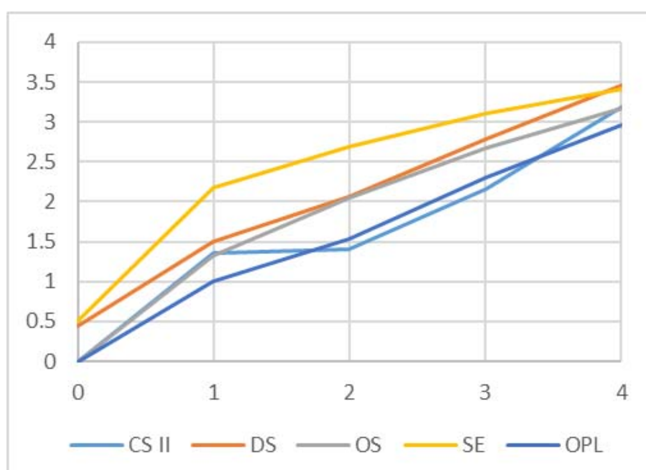
We compared the grade on each course in the required course sequence with that on the next course in the sequence to track the performance of the students as they progress through the Computer Science major.

One-way ANOVA of the grade on *Computer Science II (CS II)* with the grade on *Computer Science I* as the fixed factor yielded a significant main effect [$F(4,568) = 64.556$, $p < 0.001$]. In Table III, the row for *CS II* lists the mean grade on *CS II* for the students who scored the grade in the column header on *CS I*, e.g., students who scored an A in *CS I* scored a mean grade of 3.18 in *CS II*. Those who scored B in *CS I* had a mean grade of 2.16 in *CS II*. The same data is presented in the line graph in Figure 2.

TABLE III. MEAN GRADE ON EACH COURSE TABULATED AGAINST THE GRADE ON THE PREVIOUS COURSE

	A	B	C	D	F
CS II	3.18	2.16	1.41	1.36	0
DS	3.46	2.78	2.07	1.50	0.45
OS	3.17	2.68	2.04	1.33	0
SE	3.42	3.10	2.69	2.18	0.5
OPL	2.97	2.30	1.53	1.00	0

Figure 2: Mean grade on each course plotted against the grade on the prior course.



One-way ANOVA of the grade on *Data Structures (DS)* with the grade on *Computer Science II* as the fixed factor yielded a significant main effect as well [F(4, 371) = 74.6, p < 0.001]. For example, in Table III, students who earned an A in *Computer Science II (CS II)* had a mean grade of 3.46 in *Data Structures (DS),* and those who scored a B in *CS II* had a mean grade of 2.78 in *DS.*

Similar significant main effect was found when analyzing the grade on:

- *Operating Systems (OS)* against that on *Data Structures (DS)*: [F(4, 192) = 17.02, p < 0.001]
- *Software Engineering (SE)* against that on *Operating Systems (OS):* [F(4, 162) = 27.393, p < 0.001]
- *Organization of Programming Languages (OPL)* against that on *Software Engineering (SE)*: [F(4, 119) = 9.642, p < 0.001].

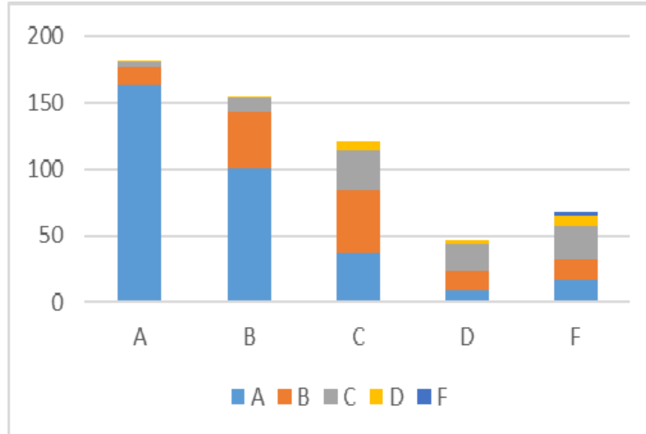
In other words, we found a significant main effect when comparing the grade on each of 5 required courses against that on the prior required course in the curriculum. As is clearly illustrated by Figure 2, the mean grade of the students who graduated with a Computer Science degree monotonically decreased from each required course to the subsequent required course in the sequence (except for the handful of students who failed a course). This provides justification for the use of the grade on the first three required courses*:* *Computer Science I*, *Computer Science II* and *Data Structures* as predictors of success in the major:

While the above analysis considered only mean grades, we found a similar pattern in the number of students when cross-tabulating grades on each pair of required courses. Table IV lists the cross-tabulation of *CS I* and *CS II* grades. Note that most of the students who earned an A in *CS I* earned either an A or B in *CS II.* Most who earned a B in *CS I* earned a B or C in *CS II.* Most who earned a C in *CS I* earned a C, D, or F in *CS II.* This is further illustrated by the stacked column chart in Figure 3, wherein, each column represents the distribution of *CS I* grade earners earning a particular grade in *CS II*. These results mirror the results of the earlier study [1].

TABLE IV. NUMBER OF STUDENTS WHO EARNED EACH OF ***CS I*** AND ***CS II*** GRADES

CS I Grade	*Computer Science II (CS II) Grade*				
	A	B	C	D	F
A	**163**	**101**	37	9	16
B	14	**42**	**47**	15	17
C	3	10	**31**	**19**	**24**
D	2	2	6	4	8
F	0	0	0	3	3

Figure 3: Distribution of *CS I* grade earners in each grade of *CS II.*



We repeated the same cross-tabulations for the other pairs of courses in the required course sequence. Figure 4 illustrates the cross-tabulation of *CS II* and *Data Structures (DS)*; Figure 5 illustrates *Data Structures (DS)* and *Operating Systems (OS);* Figure 6 illustrates *Operating Systems (OS)* and *Software Engineering (SE);* and Figure 7 illustrates *Software Engineering (SE)* and *Organization of Programming Languages (OPL).*

In all the cross-tabulations except the last one (SE and OPL), we observe the same trend as found in *CS II* → *DS* grades: A → A/B, and B → B/C. But, in a slight reversal, we note that C → C/B. In the last cross-tabulation of SE and OPL, we find that A → A/B; B → B/C and C → C/D. This pattern of grades that are usually same or lower in subsequent required courses as observed for most of the students supports our earlier assertion that grades in the first three courses of the required sequence: *CS I, CS II,* and *Data Structures* are good predictors of success in the major.

Figure 4: Distribution of *CS II* grade earners in each grade of *Data Structures (DS)*

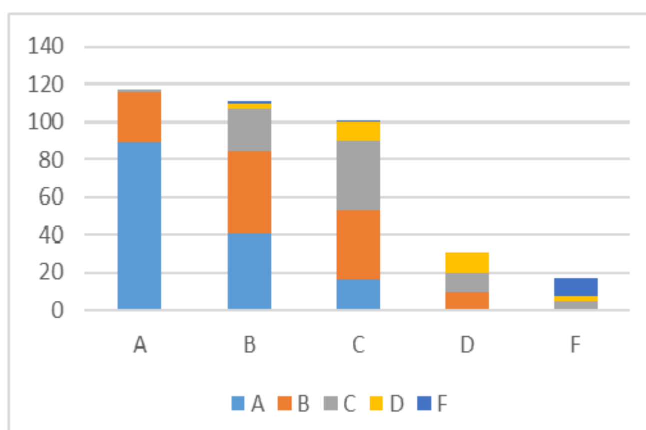


Figure 5: Distribution of *Data Structures (DS)* grade earners in each grade of *Operating Systems (OS)*.

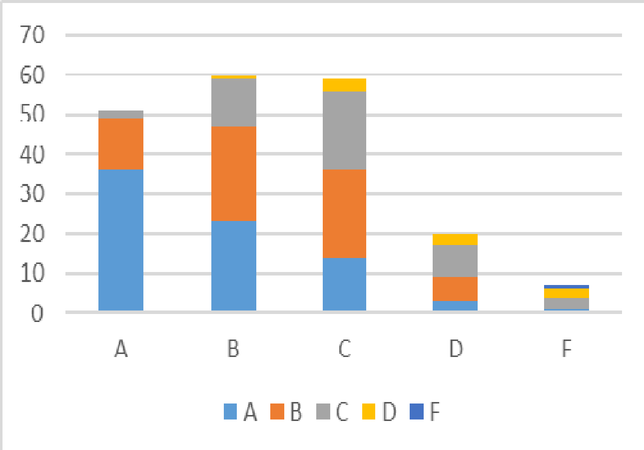


Figure 6: Distribution of *Operating Systems (OS)* grade earners in each grade of *Software Engineering (SE)*.

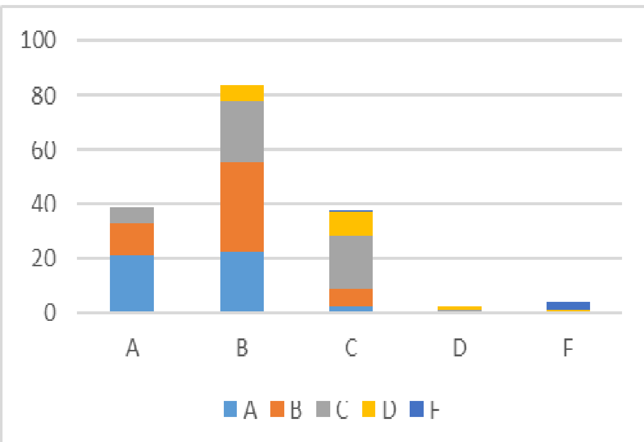
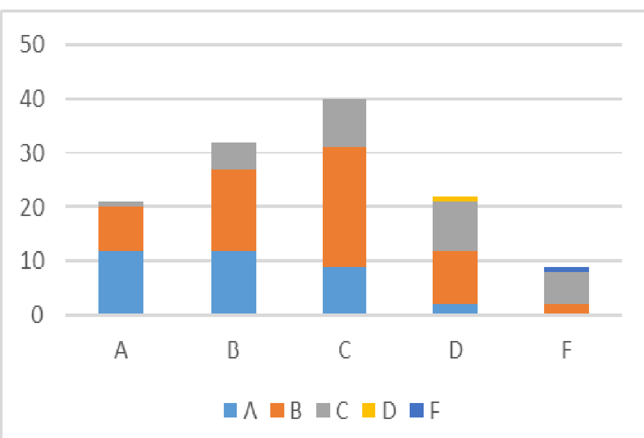


Figure 7: Distribution of *Software Engineering (SE)* grade earners in each grade of *Organization of Programming Languages (OPL)*.



D. Factors influencing performance in Computer Science I

We wanted to find out who did well in *Computer Science I*. We conducted a univariate ANOVA analysis of the grade in *Computer Science I* with sex, race, major and SAT Math score as fixed factors. This analysis included all the students, whether they had graduated or not.

We found no significant main effect for sex [$F(1,893) = 0.24$, $p = 0.624$] or race [$F(9,893) = 0.994$, $p = 0.444$], i.e., we found no significant difference in the grades of the two sexes and the multiple racial groups. We did find a significant main effect for major [$F(3,893) = 4.302$, $p = 0.005$]: Computer Science majors earned a much higher grade than the students of the other majors, as shown in Table V. Note that the number of Computer Science majors is less than in earlier calculations because of missing sex/race/SAT Math score data for some Computer Science majors.

TABLE V. ***Computer Science I* grade by Major**

Major	N	Mean ± 95% Confidence Interval
Computer Science	77	3.844 ± 0.279
Mathematics	170	2.960 ± 0.203
Science	279	2.658 ± 0.160
Non-Science	367	2.100 ± 0.150

Similarly, we found a significant main effect for SAT Math score [$F(45, 893) = 2.972$, $p < 0.001$]. The correlation between SAT Math score and *Computer Science I* grade was 0.442 and significant at the 0.01 level.

When we similarly analyzed cumulative GPA with sex, race, major and SAT Math score as fixed factors, we found a significant main effect for (and only for) SAT Math score: [$F(47, 981) = 3.90$, $p < 0.001$]. The correlation between SAT Math score and cumulative GPA was 0.608 and significant at the 0.01 level. *So, SAT Math score is a good predictor of success in Computer Science I as well as of obtaining an undergraduate degree regardless of the major.*

III. Discussion

We found that the better the grade on Computer Science I, the better the cumulative GPA of the student upon graduation, and this applied to Computer Science majors as well as non-majors. We speculate that this may be due to the centrality of problem-solving to all majors – problem-solving is one of the two skills covered in *Computer Science I,* the other being programming. If so, it further supports the importance of computational thinking [5] in higher education for all majors.

Our analysis showed that all the students who had successfully graduated with a Computer Science degree had earned at least a C grade in the first three required courses: *Computer Science I*, *Computer Science II* and *Data Structures*. These three courses lay the foundation for all the subsequent courses in the Computer Science curriculum. So, it is not surprising that the promise of academic success in Computer Science can be detected using grades so early in the curriculum. We think that such early detection is rather a boon – we can advise students better about their prospects of succeeding in Computer Science major based on their performance in just the first three semesters.

When we cross-tabulated grades of Computer Science majors on pairs of successive required courses, we found a recurring pattern: most of the students earned the same or lower grade on the next course in the sequence. This result extends the pattern identified in the prior study wherein researchers found that students did better in the first class than subsequent classes [1]. In addition, it justifies using the grades on the first three required courses not only as predictors of success in Computer Science major, but also as litmus tests for advising whether a student should continue in the major. *Students who do not earn at least a C in these three courses should be encouraged to revisit their choice of the major.* This may sound draconian, but it is in the best interests of the student. These three courses count towards a minor in Computer Science, and will not go to waste. Students who do poorly on these three courses often end up with a cumulative GPA of less than 2.0 which will prevent them from graduating. Providing an early warning after these three courses will help such students avoid the fate of not being able to graduate even after four years of higher education.

Finally, we found that SAT Math score was a good predictor of student success in *Computer Science I* as well as of obtaining an undergraduate degree regardless of the major. However, we do not plan to use this in recruitment or enrollment management decisions, since SAT scores are affected by stereotype threat [6], and are known to disadvantage women and minorities [7].

Reproducibility is a core principle of scientific research. It refers to the ability to draw the same results using different instruments, methods, protocols and/or participants [8]. Its importance is highlighted by a 2015 study of reproducibility of 100 results published in 2008 in three top Psychology journals, that found that only 35 of the 100 results could be reproduced at a statistically significant level [9]. Therefore, our effort to reproduce the results of the earlier study [1] is an important line of enquiry in engineering education research in its own right. That it was able to confirm the results of the earlier study lends additional credence to the results.

Our study strengthened the results of the earlier study by finding that the results were statistically significant. It generalized the results of the earlier study by finding that the results found in an engineering school were also applicable to a liberal arts school; and the results were found even when the circumstances of *Computer Science I* were different, i.e., pair programming was not used, and students were not segregated into sections based on their prior programming experience.

Acknowledgment

Partial support for this work was provided by the National Science Foundation under grants DUE-1432190 and DUE-1643945.

References

1. Trytten, D.A. and A. McGovern. "Moving from Managing Enrollment to Predicting Student Success." *Proc. Frontiers in Education (FIE 2017)*, Indianapolis, IN, October 2017.
2. Computing Research Association, Generation CS: Computer Science Undergraduate Enrollments Surge Since 2006. 2017.
3. Beaubouef, T. and Mason, J. 2005. Why the high attrition rate for computer science students: some thoughts and observations. SIGCSE Bull. 37, 2 (June 2005), 103-106.
4. Kumar, A.N. Closed Labs in Computer Science I Revisited in the Context of Online Testing. *Proc. Of SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2010).* Milwaukee, WI. March 2010. 539-543
5. Wing, J.M. Computational Thinking. CACM Viewpoint, March 2006, pp. 33-35.
6. Steele, C.M. A threat in the air: How stereotypes shape intellectual identity and performance. American Psychologist, 52, 1997, 613-629.
7. Rosser, P. The SAT Gender Gap: Indetifying the Causes. Center for Women Policy Studies, Washington D.C. 198.
8. Drummond, C. Replicability is not Reproducibility: Nor is it Good Science. Proc. Evaluation Methods for Machine Learning Workshop, 26^{th} ICML, Montreal, Canada, 2009.
9. Open Science Collaboration, Estimating the reproducibility of psychological science, Science, Vol. 349(6251), 28 August 2015