Re-Envisioning a Computer Science Curriculum

Kees LEUNE
Department of Mathematics and Computer Science, Adelphi University
Garden City, NY 11530, USA

Salvatore PETRILLI, JR.
Department of Mathematics and Computer Science, Adelphi University
Garden City, NY 11530, USA

ABSTRACT[1]

We have engaged in a significant restructuring of our computer science curriculum. This paper describes the process that we followed and illustrates the generalizable approach through a case study. We also demonstrate that the revision had several positive outcomes that went beyond our expectations. The case study describes a computer science computer program revision that sets goals to ensure the program's long-term viability, content alignment with labor market expectations, reasonable alignment with accreditation standards, and student success, diversity, and retention. The study demonstrated an increased ability for students to personalize their educational experience, leading to a clearly identifiable program and an improved value proposition. It also documents how the process resulted in creating a one-credit orientation seminar that contributes to increased student retention and enhances diversity in the major.

Keywords: Computer Science; Curriculum; Retention; Accreditation; Diversity.

INTRODUCTION

With growing societal concerns about the cost of higher education [1], institutes of higher education must present a clear value proposition to students [2]. As such, programs must strike a balance between developing a robust theoretical foundation allowing advanced studies, and at the same time, instill a set of practical skills appropriate for the labor market for students who choose to pursue a professional career upon graduation.

This paper outlines a structured process for continuous analysis and scheduled restructuring of an undergraduate curriculum and demonstrates that the process is effective by applying it to a traditional Computer Science program. We analyze the outcomes of the program revisions and discover several unexpected findings.

Longitudinal Study

Using data spanning the period between Fall 2005 and Fall 2013, a longitudinal study of the computer science major at Adelphi University[2] was conducted. The sample consisted of $n = 95$ computer science majors. The purpose of the analysis was to examine enrollment rates and determine the extent to which student performance in mathematics classes is related to student outcomes in the introductory computer science classes. In particular, the study focused on determining if courses *Calculus 2* and *Linear Algebra* had a significant impact on *Introduction to Algorithms and Data Structures*. Additionally, the influence of *Discrete Structures* on the upper-division computer science course *Algorithms and Complexity* was examined.

In the period assessed by the study, an average of approximately nine students entered as newly declared Computer Science majors each Fall semester, resulting in a total of 70 students who declared Computer Science as their only major upon entering the university. During the assessed period, an additional 25 student either changed their major to Computer Science after starting in another major or added Computer Science as a second major. Transfer students were not included in the study. Of the 95 students, 24.2% completed the major and 38.9% dropped it. Twenty-seven of those who dropped out left the university. There is no recognizable pattern of classes that caused students to drop the major. The mean GPA at the time at which students dropped out of the Computer Science program is 2.34 (out of 4).

Pearson correlation coefficients were calculated for the relationship between *Calculus 2* grades and *Introduction to Algorithms and Data Structures* grades and between *Linear Algebra* grades and *Introduction to Algorithms and Data Structures* grades. In both these cases, a significant linear relationship existed between the two variables ($r(34) = .364, p<.05$ and $r(28) = .447, p<.05$, respectively), indicating that students who scored better in *Calculus 2* may expect to do better in the *Introduction to Algorithms and Data Structures* course. A Pearson correlation coefficient was also calculated for the relationship between *Discrete Structures* grades and *Algorithms and Complexity* grades.

---

[1] Technical proofreading and peer-editing by Thomas Marlowe, Professor Emeritus, Seton Hall University.

[2] https://about.adelphi.edu/overview/quick-facts/

A non-significant weak correlation was found *(r(22) = .126, p>.05)*, indicating that grades for *Discrete Structures* are not related to grades for *Algorithms and Complexity*.

The outcomes of the longitudinal study demonstrated significantly that the relationships between courses is highly relevant. This finding has been a key element in determining our approach to program revision and was the leading cause for developing the method described in this paper.

METHOD

The curriculum revision process was designed as a top-down faculty-driven initiative which progressed through a number of stages. In order to document a clear understanding of the program's goal and objectives, the review process began with re-envisioning the identity of the program and by analyzing its current structure. The full process is depicted in Figure 1.

When projecting the newly formulated goals and objectives onto the existing program, course sequences were analyzed, followed by detailed reviews of each course.

The reviews focused on learning objectives per course and the relationship between the objectives and the newly identified program goals and objectives, logical sequencing of courses, and potential overlap between courses. Based on these findings, faculty developed an updated course structure and wrote revised course descriptions.

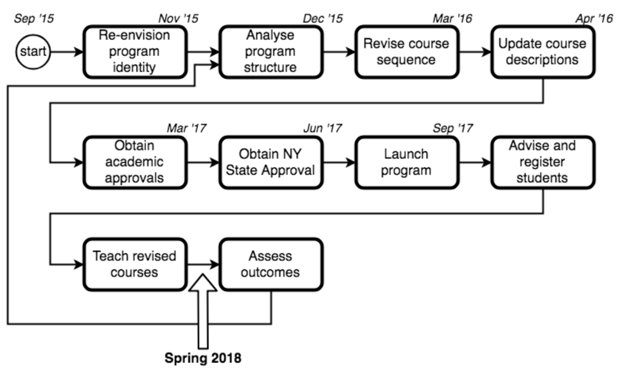

Figure 1 Program Review Process

The updated proposals then began the process of review and approval, which in most U.S.-based colleges includes a departmental review, an institution review, and, in New York, a State-level review. Once all reviews have completed, the program can be launched and students can be advised into it.

After implementing program changes, student success in achieving the stated goals must be tracked in order to make a determination as to the efficacy of the intervention. To do so, the desired learning objectives of each course should be categorized as *introducing* a goal, *reinforcing* it, or expecting *mastery*.

For all mappings, assessment rubrics were developed. The rubrics are applied to a sample of artifacts produced by students, and the outcomes form the basis of the recurring assessment. Assessment is repeated annually and will lead to curriculum reviews every five years.

CASE STUDY

The approach proposed in the previous section was tested by revising the undergraduate Computer Science program at Adelphi University.

The immediate need to revise the program was primarily driven by the changing nature of the discipline, as well as by a significant increase in the number of enrolled students. In a span of four years, the number of first-year students grew from approximately 10 students per Fall semester, to over 80 students per Fall semester.

Enrollment in computer science programs is cyclical [3]. While we are currently riding a wave of popularity of all things related to STEM-related disciplines, it is reasonable to anticipate that this will not last indefinitely. Long-term viability was a consideration.

The secondary motivation to review curriculum was the realization that the program had organically evolved and had become opaque, and, at times, illogical. While individual courses had been added to account for the changing nature of the discipline, and other courses had been removed from the schedule, a holistic program assessment had not taken place and the program had not been significantly changed since 1978.

Given the very tight market for higher education in the United States, and for the New York metropolitan area [4] in particular, it was important to ensure that the resulting program had clearly recognizable identity. Decision-making was further influenced by institutional strategic priorities, which envisions students to have greater flexibility in choosing relevant electives, graduating with a second major, and/or adopting one or more minors.

Early in the re-envisioning process, the decision was made to adopt specialization tracks in order to maximize the student's experience, increase retention, and to improves their chances of graduating on-time [5].

Revised Program

In addition to the findings from the longitudinal study, the re-envisioning process was influenced by preparation for regional re-accreditation and by recommendations made by the Association for Computing Machinery (ACM) in their 2013 Curriculum Guide [6].

Furthermore, preparing for the Middle States re-accreditation review and the longitudinal study occurred at a prime time for the Department, as a joint task force of IEEE and ACM published recommendation for curricular changes within computer science programs just after the Department's self-assessment.

The ACM/IEEE recommendations included sample course descriptions, syllabi, projects, and capstone ideas. The major thrust made by the ACM is to design a major with fewer required courses and to allow room for many more electives, which allows a student to tailor a computer science major to meet many career goals, such as Software Engineering, Information Security, Engineering, graduate school preparation, etc. The revised curriculum presented in this paper aligns very strongly with the recommendations of ACM and is grounded in our experience and assessment.

Based on earlier discussions and on the recommendations identified in the previous paragraph, the revised program set out to achieve goals such as supporting the long-term viability of the program, better content alignment with current labor market expectations, and reasonable alignment with accreditation standards, such as proposed by ABET, as well as student success, retention, and building diversity.

Lastly, the final goal of the realignment activities, and arguably one of the most important ones, was to enhance student success. By enabling students make informed decisions as early as possible, students can best plan for on-time graduation, which contributes to retention of students who wish to remain in the major. In support of these goals, a new *Computer Science Orientation Seminar* (discussed in more detail in an upcoming section) was introduced as a new mandatory first-year course. The new seminar also addressed the threat to retention caused by a lack of skills in computational problem solving in a computer science program [7].

The newly defined mission of the Computer Science program was formulated as: "To provide students with a solid foundation to think critically, to reason quantitatively, to communicate effectively both in oral and written formats, to analyze data, to design solutions, to engineer software, and to solve problems using computer science." This mission statement was further supported by a vision to "establish a supportive environment, build learning communities, mentor students, and develop internship opportunities the program will mold ethical professionals who are lifelong learners, able to harness the latest technology throughout their professional life or as they pursue graduate studies."

The mission statement and vision were subsequently refined into achievable and measurable goals and objectives. The revised goals revolved around computational problem solving, written and oral communication, practical skills, and generalizable knowledge.

A large number of changes to the curriculum were made as a result of individual course analyses. For example, courses such as *Computer Organization and Assembly Language* and *Utilities and Internals* were phased out and relevant parts of their content was merged into courses that remained; courses such as *Database Management Systems* and *Data Structures* were moved from their previous senior (resp., junior) designations to sophomore designations, and new courses such as *Operating Systems Practicum*, *Multivariable Mathematics*, and *Introduction to Cybersecurity* were introduced.

The resulting undergraduate program is structured as a collection of foundational required courses, combined with mandatory electives in one or more specialization tracks.

Table 1 lists the program-of-study. The required courses are listed in their recommended sequence. After completing foundational courses, students will have mastered basic programming techniques, understand and be able to apply basic data modeling techniques, store, retrieve and analyze data stored in relational databases, and know how to write computer programs that use efficient dynamic data structures.

All students complete the typical Mathematics requirements: *Calculus I*, *Calculus II*, and a *Discrete Structures* course. In addition, all students must complete a course in *Statistics and Data Analysis*, in which students build a proficiency in statistically analyzing data, visualizing data, analyzing and processing large data structures.

Students who have reached the end of the foundational computer science sequence will be expected to have chosen one or more specialization tracks.

The skill-focused specialization tracks include *Cybersecurity*, which provides students with detailed knowledge about offensive and defensive cybersecurity methods and techniques, *Game Development*, which teaches students how to complete technical designs and software-based implementations of computer games and multimedia solutions, and *Software Engineering*, which provides students with knowledge and skills to design, build, test, and maintain large-scale enterprise software.

In addition to these three skills-focused specialization tracks, two knowledge-focused specialization tracks are offered. They include an *Applied Sciences* track, which prepares students for a career in fields such as Engineering or Scientific Computing, and a *Foundations of Computer Science* track, which prepares students for graduate studies after completing their undergraduate degree.

Upon completion of the foundational courses and after making significant progress towards their specialization track(s), students will take a course in *Software Engineering* in which all knowledge domains that were introduced are unified into a single analysis and design course.

Lastly, students complete a year-long capstone experience. The capstone consists of a one-credit course that primarily focuses on goal definition, team formation, project scoping and planning, and of a second three-credit course in which team-based development and research initiatives are explored.

ANALYSIS

The changes proposed in this curriculum revision were implemented as of the Fall semester of the 2017/2018 academic year. While the changes are reasonably recent, the first two full years of students who participate in the revised major have passed.

Students who started prior to the academic year in which the revisions were enacted were offered a chance to transition to the new program. All but some students who had progressed in our previous program have embraced the new revised plan of study, and have transitioned to it.

Personalized Education

The strong belief that each student should be offered the potential to develop themselves to their fullest potential is an institutional strategic priority. Modern students are looking for a personalized educational experience and for a good return on investment. As such a single "one size fits all" approach to education is no longer appropriate for a private college in a highly competitive market.

After having run the revised program for two years, students have embraced the ability to tailor their experience.

The extent to which students personalize their educational experience can be captured by looking at how many students declare multiple majors, and how the additional majors are distributed. Likewise, the level of customization of the educational experience is indicated by considering the minors that were adopted. Table 2 shows a breakdown of these numbers for the Spring 2018 semester to Spring 2020 semester, which operates under the new program, compared to the Spring 2016 semester, which operated under the old program.

Despite a brief upwards development, the number of students dual majoring has mostly remained stable. Looking closer, it must be noted that most common second major that is combined with computer science is mathematics. Given the affinity between the disciplines, as well as the fact that the computer science major and the mathematics major are both offered by the same academic department, this is no surprise. Unfortunately, it is also an indicator of students not embracing the liberal arts nature of the program and mostly focusing within their own discipline.

However, we see an increase from 9.5% to almost 19% of the computer science students who have declared minors. This increase started immediately after the adoption of the restructured curriculum and appears to continue. More interesting is the distribution of the minors. The percentage of computer science students who declared a minor in mathematics has decreased somewhat, while, at the same time, the number of students adopting majors in different areas than mathematics has significantly increased. Both observations provide an indication that restructuring our curriculum did indeed result in our students' ability to personalize their educational experience.

Table 1 Revised undergraduate computer science program

Course	C	G	S	F	A
Computer Science Orientation Seminar	X	X	X	X	X
Calculus I	X	X	X	X	X
Discrete Structures	X	X	X	X	X
Intermediate Computer Programming	X	X	X	X	X
Calculus II	X	X	X	X	X
Statistics and Data Analysis	X	X	X	X	X
Database Management Systems	X	X	X	X	X
Survey of Programming Languages	X	X	X	X	X
Data Structures	X	X	X	X	X
Software Engineering	X	X	X	X	X
Senior Capstone Project	X	X	X	X	X
Principles of Programming Languages	X		X	X	
Operating Systems	X	X	X		
Operating Systems Practicum	X			X	X
Computer Networks	X	X	X		
Algorithms and Complexity			X	X	
Computer Architecture and Org			X	X	X
Introduction to Cybersecurity	X				
Computer and Network Security	X				
Applied Cryptography	X				
Game Programming		X			
Multi-variable Mathematics		X			
Artificial Intelligence		X			
Graphics & Image Processing		X			
Graphical User Interfaces		O	O		
Mobile Application Development		O	O		
Linear Algebra				X	
Number Theory				O	
Symbolic Logic				O	
Computational Mathematics				O	
General Chemistry I					X
Physics for Science Majors I					X
Physics for Science Majors II					X
Mathematical Methods in Physics I					X
Calculus III					X
Intro to ODE					X

Legend: C = Cybersecurity, G = Game Programming, S = Software Engineering, F = Foundations, A = Applied Sciences, X = required, O = elective (choose one)

Building student confidence

As mentioned earlier, a new course, the *Computer Science Orientation* seminar was developed to increase student's ability to engage in computational problem solving, and to increase retention in the major. Williams [8] has shown that writing solutions to problems and nurturing algorithmic and computational thinking increases problem solving skills. Buck [9] points out that it is important to make incoming students aware of career opportunities early on.

Table 2 Educational Diversity in CS students

	Spring ‘16		Spring ‘18		Spring ‘19		Spring ‘20	
	Absolute	Relative	Absolute	Relative	Absolute	Relative	Absolute	Relative
Enrollment	94	100.0%	127	100.0%	129	100.0%	158	100.0%
Dual major CS/Math	8	8.5%	17	13.4%	7	5.4%	5	3.2%
Dual major CS/Other	1	1.1%	4	3.1%	3	2.3%	5	3.2%
Dual major CS total	9	9.6%	21	16.5%	10	7.8%	10	6.3%
Minor in Math	7	7.4%	6	4.7%	7	5.4%	6	3.8%
Other non-math minor	2	2.1%	12	9.4%	16	12.5%	21	13.3%
Multiple minors	0	0.0%	5	3.9%	0	0.0%	3	1.9%
Total with minor	9	9.5%	23	18.1%	23	17.8%	30	19.0%

Petrilli [10] demonstrated that an orientation seminar of this nature increases retention, and better prepares all students to succeed in a mathematics or computer science major. Chen [12] states that women and minority students drop out of the computer science at higher rates as compared to white males. An additional reason for introducing the *Orientation Seminar* was to build diversity in the major. The Orientation Seminar is discussed in more detail in Leune and Petrilli [11].

First-year students were polled about their backgrounds in computational thinking. It quickly became clear that the majority of students in the program had no prior background in the computer science field, which further supported our decision to design additional course work that sets out to instill a sense of computational problem solving, and to provide a broad overview of computer science and of the information systems fields.

The *Computer Science Orientation Seminar*, which was introduced as a mandatory course for all computer science students and for all information systems students has two main goals: to provide students with a basic understanding of the computer science field, and to teach students proper “being-a-computer science-student”-etiquette.

Johnston et al. [13] have demonstrated that it is important to manage expectations appropriately. Therefore, the course introduces “learning-to-learn”-content in addition to foundational computer science knowledge. For example, the very first assignment that the students were given was to locate the computer science tutoring labs, identify the tutors, introduce themselves, and ask them for their hours. As trivial as the exercise seemed to be at first, the students' feedback was positive; it helped them navigate new buildings, learn how to ask for help, and work on scheduling. Other content that was introduced along the way included studying syllabi, writing a proper email to a professor, asking for letters of recommendations, and writing computer science papers.

All students who completed the course were invited to participate in an anonymous survey at the end of the semester. The first question asked, *Has this seminar changed your feelings towards being a computer science major?* 71% of the responses (n = 37) indicated that the seminar had changed their feelings towards the major. However, the qualitative results showed that majority of the students did not want to change their major, but rather changed their perspective towards the computer science major. One student noted “Before I added the computer science major, I wasn't very confident that it was something I wanted to do. The seminar gave me more insight on what the major was like and made me more confident in my decision to keep the major.” Another said: “The seminar has made me want to keep pursuing a computer science major. I feel that I've learned a lot more about computer science as a whole and what it takes to be a successful IS or computer science major.”

Also, this seminar showed some students that the computer science major was not what they wanted. For instance, “It did make me interested in computer science, but I realized that while it's interesting, I don't want to major in this or begin a career in this field.” This may seem to be a negative result of the course; however, it is beneficial for a student to decide early in the college career that a major is not right for them.

Table 3 Feelings about the Department and the computer science major

Statement	**Median Response**
I would feel comfortable enough interacting with most members of the Math and Computer Science Faculty	4
I have found or begun to find my passion in computer science/information systems	4.5
I am more confident in my career path now than before I started the course	4.5
I feel more comfortable seeing a professor for office hours	4.5
I feel more likely to seek out help from our department’s tutors	4

The second question asked, *Has this seminar changed your feelings towards your career goals with a degree in computer science?* Out of the total class, 58% of the responses indicated that it had changed their career goals. For instance, one student stated, "Yes, I had no idea about the aspects of computer science --- now I know I want to be a software engineer." Another student stated: "I have strongly considered a career in cybersecurity after graduation." Also, "I have become more motivated to find out about jobs that are in respect to my concentrations that I am interested in and do more research to see what the most interesting job I could have in the future."

For the remaining students, this seminar did not change their career goals. There were two extremes, either students entered this seminar knowing what they wanted to do, or they left this seminar still unsure about what they want to do. For instance, one student stated "Before taking this course, I knew I wanted to have a job in cybersecurity. This course did not change my mind." Another student stated, "Even though I have completed the course, I still don't have any sense of direction in terms of career goals as of now as a computer science major."

The questions from in second section of the survey were assessed on a Likert Scale, which had the following responses: 0 indicated *Not at All*, 1 indicated *Somewhat*, and 2 indicated *Very Much*. We cluster questions 7–15 into two categories: *Feelings About the Department and the computer science major* and *Skill Sets Acquired*. Table 3 summarizes students' self-reported perceptions of the program.

One of the main goals of this *Orientation Seminar* was to help students feel more comfortable with the department and the major, and finding their passion in computer science. The results collected from this survey indicate that the interactions with the full-time faculty have made them more comfortable with the Department and the faculty. These feelings are consistent with that found by [14], [15]. Results from the first question from the last section of the survey indicate that this was the most useful part of the course. A representative student indicated "Drs. Stemkoski and Hiller really convinced me to do more with my abilities and go further. It has opened my eyes to how much there is to do."

Students were also asked to identify what they considered were the three objectives of this *Orientation Seminar*, then to evaluate their own knowledge before and then after the seminar. From a statistical point-of-view, this is a very tricky question, because it is unknown how students will respond. However, there were three objectives that majority of the students indicated as being most important in this seminar: *Learning about our Department and What Computer Science is* ($n = 28$, 76%), *How to be a successful Computer Science Major* ($n = 25$, 68%), and *Problem Solving* ($n = 12$, 32%).

A Wilcoxon test examined the results of students' perceptions on their knowledge about the department and Computer Science before and after this seminar. A significant difference was found in the results ($Z = 4.684$, $p < .05$). Students' perceptions on their knowledge of our department and Computer Science increased significantly throughout this seminar. These quantitative results are highly supported by the qualitative results from the survey. A representative student stated "Nothing. I gained first-hand experience on what each track has to offer in terms of how they function (networking, cybersecurity, games, etc.)."

Another Wilcoxon test examined the results of students' perceptions of their problem-solving skills before and after this seminar. A significant difference was found in the results ($Z = 3.109$, $p < .05$). Students' perceptions of their problem-solving skills significantly increased throughout this seminar. Interestingly, a Spearman *rho* correlation coefficient was calculated for this relationship, resulting in a strong positive correlation ($rho(12) = .687$, $p < .05$), indicating a significant relationship between student perceptions of problem solving skills before and after taking this seminar.

Student leadership

As mentioned in before, a design goal of the program redesign was to ensure that retention benefits positively from the changes. Prior research has demonstrated that students experiencing a strong sense of community have better outcomes and higher retention [16–18].

In the newly introduced *CS Orientation Seminar*, students are actively encouraged to collaborate in groups, work with departmental tutors, and to become active participants in the CS community.

Within one year, this has led to positive, but somewhat unexpected results. Students took the initiative to form a student chapter of the Association of Computing Machinery (ACM) and were able to establish themselves as the go-to organization for all computing-related student activities on campus.

The board of the student chapter is formed exclusively by women students, which has a strong positive impact on gender-based diversity. All board members either started in the revised program, or transitioned into it.

After successfully establishing the chapter, the leadership has further moved to establish an ACM-W student chapter, which will profile itself as the student organization for *Women in Science and Computing*. As that organization launched, representation from a large variety of academic disciplines was present from the onset. Membership includes women majoring in Computer Science, Mathematics, Chemistry, Biology, Information Systems, Physics, etc.

OUTCOMES AND CONCLUSIONS

Academic curricula must evolve to remain relevant. In order to do so, they must align with the demands and requirements of today's students, which includes ensuring that programs provide a clear value proposition and enable students to graduate on-time and with a clearly identifiable degree. Curriculum review must be an ongoing process, and academic departments must acknowledge that courses may no longer directly relevant, new courses must be introduced and overlap between courses must be minimized.

The program revision process yielded re-sequenced courses, and removed courses that were exhibiting content that significantly overlapped with other subject areas. It also introduced a number of new courses in areas that were previously underdeveloped.

By implementing these changes, the overall size of the program-of-study of the Computer Science major was reduced from 78 credits to 61 credits. We strongly believe that the program's relevance and its identity have been significantly improved, and that academic rigor has been strengthened.

An intended outcome of our revision was that it would become more feasible for students to declare more than one major, and/or to declare minors, thus increasing their labor market value upon graduation and enabling them to pursue a variety of academic interests. Unintended outcomes included soft-skill development, resulting in more confident students, and students who pursued extracurricular embraced leadership experiences.

Paired with the overall credit reduction, the number of courses being offered was increased, but they were organized in a common foundation, paired with mandatory-elective courses grouped in specialization tracks. After completing the specialization tracks, all students experience common final courses, which include a *software engineering* course and a two-semester *senior seminar* capstone experience.

FUTURE WORK

Work on a curriculum is never done. This is even more true in a world that is rapidly evolving as ours is, and which has never seen rates of technology adoption as high as they are now.

Beginning in academic year 2018/2019, assessment of student success will continue by closely monitoring retention and on-time graduation. We will continue monitoring advances in the discipline and in academia, as well as in industry, and work to ensure that our programs align with those advances to the largest extent feasible.

ACKNOWLEDGMENTS

The authors would like to acknowledge the team effort by the Department of Mathematics and Computer Science at Adelphi University in designing and implementing the revised program described here.

REFERENCES

[1] L. Rothman, “Putting the Rising Cost of College in Perspective,” *Time*, 2016.

[2] G. W. McClung and M. W. Werner, “A Market/Value Based Approach to Satisfy Stakeholders of Higher Education,” *J. Mark. High. Educ.*, vol. 18, no. 1, pp. 102–123, 2008.

[3] A. Nager and R. Atkinson, “The Case for Improving U.S. Computer Science Education.,” 2016.

[4] T. DiNapoli, “The Economic Impact of Higher Education in New York State.,” 2010.

[5] E. A. B. (EAB), “Promoting Timely Degree Completion: Reconciling Student Choice and the Four-Year Graduation Imperative,” 2016.

[6] T. J. T. F. on Computing Curricula, “Computer Science Curricula 2013, Curriculum Guidelines for Undergraduate Degree Programs in Computer Science,” Dec. 2013.

[7] T. Beaubouef and J. Mason, “Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations,” *SIGCSE Bull.*, vol. 37, no. 2, pp. 103–106, Jun. 2005, doi: 10.1145/1083431.1083474.

[8] K. Williams, “Writing about the Problem-Solving Process to Improve Problem-Solving Performance.,” 2003.

[9] Robert E. Buck and Elise M. Grabner and Gary B. Roberts, “Mathematics, Curriculum, Careers,” *PRIMUS*, vol. 8, no. 2, pp. 184–190, 1998.

[10] S. Petrilli, “The Mathematics Orientation Seminar: A Tool for Diversity and Retention in the First Year of College.,” *J. Humanist. Math.*, 2019.

[11] Kees Leune and Salvatore Petrilli, “The Computing Orientation Seminar: A Model for Enhancing Diversity and Retention,” *ACM Inroads*, 2019.

[12] X. Chen, “STEM Attrition: College students’ path into and out of STEM fields.,” 2013.

[13] W.W. Johnston and J.E. Webster and C.J. Wilson, “An Introductory Research Experience in Mathematics for Undergraduates.,” *PRIMUS*, vol. 27, no. 4–5, pp. 451–460, 2017.

[14] M. Nicpon, L. Huser, E. Blanks, S. Sollenberger, C. Befort, and S. Kurpius, “The relationship of loneliness and social support with college freshmen’s academic performance and persistence.,” *J. Coll. Student Retent. Res. Theory & Pract.*, vol. 8, no. 3, 2007.

[15] Curtis V. Smith, “The Impact of Part-Time Faculty on Student Retention: A Case Study in Higher Education,” University of Missouri-Kansas, 2010.

[16] R. Fullilove and P. Treisman, “Mathematics Achievement Among African American Undergraduates at the University of California, Berkeley: An Evaluation of the Mathematics Workshop Program,” *J. Negro Educ.*, vol. 59, no. 3, pp. 463–478, 1990.

[17] B. B. Hughes, “Coming out in STEM: Factors affecting retention of sexual minority STEM students,” *Sci. Adv.*, vol. 4, no. 3, 2018.

[18] A. D. Rayle and K. Chung, “Revisiting first-year college students’ mattering: Social support, academic stress, and the mattering experience.,” *J. Coll. Student Retent. Res. Theory & Pract.*, vol. 9, no. 1, 2008.