Promoting Inclusivity in Computing (PINC) via Computing Application Minor

Ilmi Yoon[1], Pleuni Pennings[2], Anagha Kulkarni[1], Kaz Okada[1], Carmen Domingo[2]

[1]Computer Science, San Francisco State University
[2]Biology, San Francisco State University

Abstract— We aimed to build a new educational pathway that would provide basic training in computer science for women and students from underrepresented (UR) groups who otherwise may not take computer science classes in college. Specifically, this on-going project focused on creating a 2-year Computer Science (CS) program consisting of exciting new courses aimed at biology majors. Biology traditionally attracts large numbers of women, a significant number of students from UR groups, and has compelling needs for CS technology. The interdisciplinary program is training the next generation of innovators in the biological sciences who will be prepared to cross disciplinary boundaries. The program consists of the following: (1) computer science courses with content related to biology, (2) cohorts of students that progress through the program together, and (3) a small group peer mentoring environment, and (4) facilitated interdisciplinary research projects. Graduates from this program, referred to as "PINC" - Promoting INclusivity in Computing - will receive a "Minor in Computing Applications" in addition to their primary science degree in Biology. The program is now in its second year and thus far 60 students have participated. Among them, 73% are women and 51% are underrepresented minorities (URM). The majority of students in the PINC program stated that they would not have taken CS courses without the structured support of the PINC program. Here we present the data collected during this two year period as well as details about the Computing Application minor and programmatic components that are having a positive impact on student outcomes.

Keywords— Computing Application Minor, Inclusivity, women, underrepresented group

I. Introduction

Computer Science (CS) is one of the fastest growing areas of employment, a trend predicted to continue in the future [1]. By 2024 the US Bureau of Labor Statistics predicts that 1.1 million CS jobs will exist, but only 450,000 CS graduates will be available to fill them [2]. Unfortunately, the market's demand for individuals with CS knowledge outstrips students' interest in majoring in CS by a magnitude of 5.5, and by 2024, McKinsey & Company estimates that the US could face a shortfall of 250,000 data scientists (typically interdisciplinary scholars with domain knowledge and CS expertise to process and analyze the domain data) [3]. This problem is particularly acute in science, where big data has become central to research. From sequencing the microbiome to mapping distant galaxies, CS is at the heart of cutting-edge research. In a single year, a large lab like the one that hosts CERN's Large Hadron Collider can generate roughly 15 petabytes of data—the equivalent of roughly 4 million high-definition feature films [4]. The pace at which data is acquired is continually accelerating and yet this data is useless without someone to process and analyze it.

In addition to suffering from a dearth of interested students, the field of CS suffers from a lack of diversity in its workforce. In 2014, only 18% of bachelor degrees in CS were award to women, only 7% to African Americans, and only 10% to Hispanics/Latinos [5]—these numbers are far lower than one would expect based on these groups' share of the population. Indeed, today CS remains one of the least diverse STEM disciplines. In addition to signaling a substantial problem with educational equity [6], society has increasingly come to realize that this lack of diversity in the CS workforce makes it difficult to carry out innovative work and avoid unnecessary mistakes [7]. The effects of researchers' pre-existing biases—stemming from factors such as gender and culture—on scientific research has been well established [8]. Personal experience influences the scientific questions that get asked, the way that data are gathered, and how they are interpreted [9]. Therefore, the fact that only 2% of Yahoo employees are African American and 4% are Hispanic—numbers typical of the tech world—is a problem that affects not only underrepresented (UR) groups, but society as a whole [10].

The PINC program (Promoting INclusivity in Computing) was developed to address the given problem at San Francisco State University (SFSU). Rather than expecting women and UR students to pursue a CS major, we used an evidence-based approach to design a CS program that would help students from all scientific disciplines succeed in learning CS skills. In the next chapters, the identified barriers and the evidence-based approaches (chapter 2), PINC curricular components (chapter 3), PINC non-curricular components (chapter 4), quantitative data as well as qualitative data (chapter 5) and conclusion will be presented.

II. Background

Educational barriers prevent broader participation in CS courses.

The first step toward increasing the diversity of individuals obtaining a foundation in computing is the recruitment of a large and diverse pool of students. However, the STEM education literature highlights a number of barriers that prevent students from seeking training in CS. The combination of a lack of background knowledge and experience in CS along with the lack of a sense of belonging, will send a powerful message to potential students that they may not have what it takes to succeed in CS. This environment can trigger stereotype threat, the phenomenon in which students who fall outside of the norm unwittingly conform to stereotypes of their group, due to the stress of knowing others' low expectations of themselves. This causes students to underperform [11-14]. In addition, students who have avoided intensive quantitative courses may have challenges in developing valuable computational skills [15]. If students lack confidence in their ability to learn CS they may also develop imposter syndrome [16], which can reduce their desire to participate in CS courses. Finally, when CS classes are taught without iterative assessment then instructors may overestimate students' starting knowledge, leading to assignments that are too difficult and fail to engage students' interest [17,18].

Conversations with science students on our campus confirmed that they had experienced these barriers. Many of our students reported having felt alienated by computers and mathematics from a young age. Moreover, although some students were interested to learn CS, they were unsure which CS class to begin with. Other students reported that they had experienced disappointment after enrolling in a CS class because of one or more of the issues stated below.

- The classes seemed difficult and abstract. It was difficult to see how the skills they were learning applied to real-life problems; **(content seemed irrelevant)**
- The introductory CS courses had many students with programming experience, and thus were not true beginner classes; **(imposter syndrome)**
- They were the only one (or one of a few) women and/or URM student(s) in the room; **(stereotype threat)**
- They found themselves in an unsupportive environment, with professors announcing "only half of you will pass this class." **(stereotype threat)**

As a result, many students perceived obtaining CS skills as too daunting and elected to study something else. Other students attempted CS courses but found learning the material too onerous.

III. PINC Co-Curricular Components

A. Cohort System

Cohort system of learning represents evidence-based high impact pedagogy [22]. Cohort learning consists of a relatively small group of students – usually between 12 and 25 start and finish their degree together. Cohort programs usually have strong cohort administrators. This is necessary to organize and keep students on track through the program. Cohort programs offer community building and collaborative opportunities. Students build relationships with people who have similar goals and backgrounds. A cohort system creates camaraderie among the students, which helps them overcome academic challenges. Moreover, if the environment is diverse, it can substantially reduce stereotype threat and imposter syndrome because they will not feel as socially isolated as in a typical CS large classroom.

B. Peer-Lead Team Learning (PLTL)

All of the courses in the PINC program integrate peer-led team learning (PLTL). PLTL is a nationally recognized model of teaching and learning in which peer leaders facilitate small group learning. Students who participate in PLTL demonstrate improved course performance, retention, and attitudes about coursework, compared with those who do not [23].

The peer mentors consisted of both "peer" (e.g., CS students who are at a similar academic rank as those in the PINC program) and "near-peer" (e.g., CS students who have advanced CS training such as those in the MS program) mentors. The mixture of various academic levels among our mentors was, in part, necessitated by existing scarcity of students whom we thought would be a good fit to assume a mentoring role. Specifically, we sought CS students with not only adequate CS knowledge, but also with strong personal skills that shared our mission to increase diversity in CS education. In many cases, our mentors were women and from UR groups. We have also recruited CS Master's students who had undergraduate degrees from non-CS fields. These MS students are required to take additional undergraduate courses to be prepared for the demands of the Master's program. Thus, in terms of technical skills, they serve as peer mentors, although they are much more advanced academically.

Most of the PINC mentors had no prior mentoring experience. In order to train these peer and near-peer mentors effectively and efficiently, we implemented a series of monthly workshops facilitated by a PINC faculty member. The overall number of mentors was small (i.e., four in Fall 2016 and six in Spring 2017). For each workshop we had one of the mentors act as a facilitator and the supervising faculty assume the role of the note-taker. Everyone, including the faculty supervisor, was referred to by first-name, in order to promote an equal power structure. The goal was to create an environment where the mentors would assume ownership over the mentoring component of the program. The overall structure of the workshop was to encourage all of the mentors to brainstorm strategies that would help them to identify and resolve issues in learning the CS materiel by the PINC students. The outcomes of these workshops were shared and activities and strategies were coordinated with the course instructors. Beyond co-discovering effective tutoring methodologies to resolve specific student issues, the PINC faculty emphasized the importance to the mentors of being role models who could inspire and encourage the creativity of the PINC students who were just starting to learn CS skills. The mentors were also encouraged to share their personal stories on how their CS studies started as well as their own struggles with the material.

In Fall 2017, PINC mentors had to serve two independent PINC cohorts teaching both first and second year courses. Thus, the mentor roster doubled and twelve mentors were hired that would serve three different courses. To address the diversity of duties and academic levels of these mentors, we adjusted the co-training workshops by including several group exercises that allowed mentors to share information across courses. This was done by first hosting two workshops in which mentors from the same course discussed issues they were facing with their particular students. This was then followed by a session in which mentors were mixed across different courses and academic levels with the goal of identifying any recurrent issues and themes among their students regardless of the course they were enrolled in. This strategy was repeated twice over the course of the semester. After these workshops, the mentors met to discuss and vote on the most important goals for the mentors to address. Below are the results of this vote (mentors were allowed multiple votes):

(6 votes) Encourage students' cooperation for debugging
(6 votes) Inspire students' motivations
(6 votes) Set deadlines and tell instructors if they are not met
(5 votes) Aim for better communication between students and mentors
(4 votes) Prepare more and simpler examples for students' exercises
(3 votes) Remind students to come prepare for each mentor session

At this workshop, mentors discussed the feedback they obtained from their peers. To facilitate co-learning, all of the meeting minutes were shared among the peer mentors as well as the PINC instructors and PIs.

To support the PINC students in their independent capstone project (CSC 698a and 698b) during the second year of the program, we used an Affinity Research Group (ARG) model [24]. Each ARG consisted of 4 PINC students supervised by a CS undergraduate or graduate student who acted as the peer research mentor. These research mentors received 3 units of academic credit for participation in this program and participated in the above-described peer mentor workshop with mentors serving for other courses. The mentors use their in-depth programming/technical knowledge to facilitate learning and provide technical support to the PINC students. To enable mentors to effectively support the PINC research groups, these research mentors participated in a teaching workshop during the summer. The summer workshop was followed by weekly meetings during the semester with the PINC instructors and PIs as mentioned above.

Both PLTL and ARG support cooperative learning, positive interdependence, face-to-face interactions, individual accountability, and group processing [23,25]. In particular, the small distance between mentor/teacher and learner takes advantage of the cognitive and social congruence between the two populations [26]. PINC students likely benefitted from their interactions with peer mentors who have in-depth CS knowledge and can socialize with them as well. In turn, the peer and research mentors also gained professional development skills that are likely to make them stronger group leaders and better students.

C. Professional Development and other activities

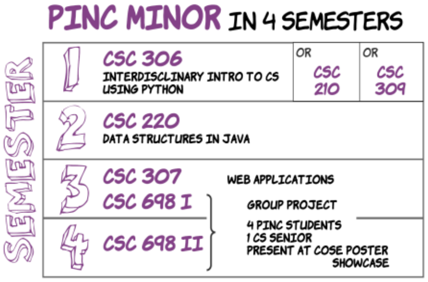
The PINC faculty hosted at least one professional development event each semester to help PINC students meet individuals in the CS workforce that can act as potential role models as well as provide examples of interesting career possibilities. We invited external guest speakers, alumni, and faculty that provided career advice, and inspired interesting research possibilities for the capstone class (CSC 698; fall 2016, spring 2017). We also hosted a poster session of 698 projects, which was then followed by a student panel that discussed internship and research opportunities along with strategies to improve their chances for being selected for these opportunities. These events also enabled PINC students in cohort 1 to share their experience with cohort 2. All of these activities helped PINC students to understand the importance of their CS training and how it aligns with specific career goals. This information helped to increase their motivation develop CS skills.

IV. PINC Curricular Components

The Computing Application minor consists of a sequence of five 3-unit courses designed to progressively develop students' computing knowledge such that they become competent to pursue interdisciplinary CS careers (Table 1). The minor begins with three lecture courses (306, 220, and 307) that provide students an opportunity to explore meaningful programming assignments that align with their interests. Once students develop a strong CS foundation, they will move on to a sequence of group research courses (698a and b) that will help them develop confidence, independence, and creativity through their interdisciplinary research projects. Thus far, roughly 30% of PINC assignments use biology-related content.

Below is a more detailed summary of the courses that constitute this CS minor.

CSC 306 *An Interdisciplinary Approach to Computer Programming***.** This introductory course teaches the basics of programming for interdisciplinary problem solving. Topics include the building blocks of programming (variables, control statements, iterative statements, arrays, functions, and abstraction) and problem solving approaches. We first used the MIT App Inventor program [27] to introduce the joy of programming (3 weeks) and then teach Python for the rest of the semester. In the PINC program, we developed biology-focused "TranslateDNA" and "BacteriaSimulation" programming assignments that were popular with the students. In PINC, we continue developing assignments that will have a broader appeal to science students across the STEM degree programs (vector and trajectory visualization for physics majors and planetary orbit simulation for astrophysics majors) that will build on students' scientific backgrounds. These assignments will be collaboratively designed with non-CS faculty from different science departments.


PINC MINOR IN 4 SEMESTERS

SEMESTER

1 CSC 306 INTERDISCLINARY INTRO TO CS USING PYTHON | OR CSC 210 | OR CSC 309

2 CSC 220 DATA STRUCTURES IN JAVA

3 CSC 307 WEB APPLICATIONS

3 CSC 698 I

4 CSC 698 II

GROUP PROJECT
4 PINC STUDENTS
1 CS SENIOR
PRESENT AT COSE POSTER SHOWCASE

CSC 220 *Data Structure and Algorithms***.** In this course, students learn important data structures, efficient algorithms and performance analysis. While this builds a very important foundation for computing competency, the materials may seemed challenging and not directly relevant to biology. Thus, similar to CSC 306, we develop class examples and programming assignments that utilize various STEM topics.

CSC 307 *An interdisciplinary Approach to Web Programming***.** This course teaches the basics of www engineering, as relevant to interdisciplinary problem solving. Topics will include developing web and database applications, HTML, PHP, Python, SQL, and MySQL databases. Programming assignments include developing citizen science web applications related to students' interests.

CSC 698a and 698b*, Research Topics in Computing Applications***.** 698a and 698b are similar to the capstone courses that seniors undertake to demonstrate knowledge of their major and develop independent research skills through experiential learning. PINC students have taken only a few CS courses before enrolling in these capstone classes. Therefore, we provide a supportive framework in which student teams choose their own projects. Peer research mentors help the student groups to stay on track. Each student is assigned to a team of 4 PINC students and 1 CS senior or graduate student. The peer research mentor provide step-by-step scaffolding to ensure that students make progress. In addition to helping PINC 2.0 students develop autonomy, these classes offer a wonderful opportunity for peer mentors to see how CS can be applied to a diverse array of STEM topics. They will also gain valuable leadership experience.

The first course, *698a*, combines lectures with the development of an independent research project. Students form teams to identify and design a group research topic. Lectures then help develop and refine the research topic. The teams identify the CS technical components needed to build their proposed solution and then study those components by building sub-component solutions. Thus, in 698a, students focus on learning about the technology related to their project. In 698b, students continue with the projects they started in 698a. The teams now integrate and develop a complete solution for the problems they identified in the prior semester. Lectures cover pertinent methodologies and software engineering practices. Students focus on implementation and experimentation. Examples of past projects include, processing images of biological cells, classifying HIV sequence data to assign subtypes, creating an educational game to help teach genetics, and detecting blood alcohol levels by using a combination of facial image processing, audio analysis, and geo location.

CSc 698 uses a project-based learning approach where real-life applications chosen by the students will drive their hands-on experiential learning. When students work on problems that interest them and on issues that they care about, the motivation to learn comes from within [28].

CSc 698a is the first course in a two-part sequence. For its inaugural offering, in Fall 2017, 21 PINC students enrolled in CSc 698a. Of the 21 students enrolled, 18 were undergraduates, one graduate, and two post-bac students. Students were majoring in Microbiology, Biochemistry, Physiology, Cell & Molecular Biology, Biotech, Pre-Nursing, and Applied Mathematics. All of these students had taken one or two lower-division CS courses in prior semesters.

At the beginning of the semester the students were organized into 6 groups, consisting of 3 to 4 students in each group. These groups were formed by the instructor based on student's (i) project interests (ii) lack of interest in any specific topic, and (iii) weekly availability. Each group was assigned a dedicated research mentor who was either a CS Senior or a graduate student. This was motivated by the well-established benefits of near-peer mentoring, and peer-led team learning [23, 24]. Every research mentor held weekly meetings for 1.5 hours with their respective group. The minutes of these meetings were drafted and submitted by each research mentor to the PINC PIs. These meetings between PINC student and their research mentors, were in addition to the regular lectures. Student attendance at lectures, and research mentor meetings were recorded, and incentivized through the final course grade.

Skills and tools that were common and useful to all the groups were taught and discussed during the lecture time. An example of such a skill was "analytical thinking" and "problem solving". To illustrate how to approach a complex problem, several example projects were systematically broken down into smaller, manageable sub-problems. By leveraging this analysis, a modular solution was designed. Students were then taught how to think of each module as a "black-box", where the focus is on defining the input and output from each "black-box", while ignoring the actual implementation details of the module. The students were asked to conduct this black- box analysis for their respective projects. This exercise had two extremely important outcomes: first, it identified the modules that they would need solve and second, it identified any misunderstandings about the functionality and organization of their modules by members of their group.

The other topics that were applicable to all of the student groups, were project planning, and version control using git. In addition to these topics, students learned a variety of technical approaches, such as, package managers, web development libraries and platforms, machine learning, image processing, data APIs, UI design tools, and string processing. They also learned non-technical skills, such as, individual and group time management, maintaining equity in member contributions and responsibilities, maintaining clear and regular communication, and identifying time to seek external help. The skills the students learned are summarized in the result section.

Table I. Curriculum Map Matrix

	SLO 1	*SLO 2*	*SLO 3*	*SLO 4*	*SLO 5*
CSC 306: *An Interdisciplinary Approach to Computer Programming*	I				
CSC 220: Data Structures	D	I			
CSC 307: An Interdisciplinary Approach to Web Programming	D	D	I	I	
CSC 698a: Interdisciplinary Design of Computing Applications for Science and Engineering	D	D	D	D	I, D
CSC 698b: *Collaborative Development of Computing Applications for Science and Engineering*	M	M	M	M	M

Note: *"I" denotes "Introduced", "D" denotes "Developed" and "M" denotes "Matured".* ***SLO 1:*** Students will demonstrate basic skills in computer programing with a major computer languages such as Java and Python. ***SLO 2:*** Students will develop simple software programs integrating core concepts of computer science, including lists, stacks, queues, trees, tables, graphs, recursion, iteration, sorting, search, and hash table. ***SLO 3:*** Students will demonstrate basic skills in setting up database applications by applying core concepts of database theory and management. ***SLO 4:*** Students will demonstrate basic skills in developing simple internet applications by applying core concepts of web-related technology, such as PHP. ***SLO 5:*** Students will design and develop a software prototype addressing relevant problems in their major studies by integrating core concepts of computer science and basic skills of database and web-related technology.

V. Results

A. Student project topics and student experience

HealthMsgU: This project aims to develop a web application that mines data on social media (primarily, Twitter) about stigmatized diseases (AIDS/HIV) and their treatments or preventions (PrEP) with the goal of informing public health researchers and practitioners about the struggles and societal challenges that patients experience. Three undergraduate students in Ecology, Cell and Molecular Bio, and Microbiology, proposed and defined this project with the help of faculty advisors from Bioinformatics and Computer Sciences. With guidance of by a CS Senior research mentor, the students developed a solid prototype for this web application that is complete with user-stories and multiple data representations. Next semester the students plan on making the web application functional. They will also introduce sentiment analysis to gauge the current public perception about medical conditions and social media.

Woozie: This project aims to develop a mobile application that estimates intoxication level of the user based on multiple inputs, such as, the number of drinks, speech quality, facial changes, and their geolocation. The goal is to use the ever-present mobile phone to facilitate responsible drinking. This project requires ground-up web development, and working knowledge of machine learning to conduct the necessary audio and image analyses. Two undergraduate students in Biochemistry, and two post bacs from Biology proposed and defined this project. Similar to the project above, the Woozie team developed a prototype for the app that will be expanded in the Spring 2018 semester.

GatorHealth: The goal of this project is a mobile app for Student Health Services (SHS) at SFSU [29]. The existing mobile app for SHS provides only two functionalities, even though the SHS website provides many features and information. The new mobile app, GatorHealth, which is being developed by four undergraduate students from Physiology, Nursing, and Chemistry, offers the full range of features and services that one would expect from SHS, such as, Appointments, Pharmacy, Services, and Resources. As part of the project the students have also successfully pitched their project idea to the administrators at SHS, and have obtained their support and buy-in. The team is confident about delivering a user-friendly app that provides a rich experience to students of SFSU by the end of next term.

CatGen: This project is focused on developing an educational web application for an introductory genetics course. The three students in this group (Biotech, Microbiology, and Cell and Molecular Bio) drew form their in-class experience to propose and design this project. Currently in Genetics, students are required to install a desktop application order to visualize the transmission of traits from parents to off springs. This application is extremely clunky and difficult to use. This experience motivated the group to develop a web application that does not require any installation, and is user-friendly. They also replaced the commonly used organism in these programs, fruit flies, with a funner organism, cats, which allowed them to offer a wider array of physical traits (coat colors, patterns, paw pad colors, eye colors, and fur length) to experiment with. They are developing an interactive app that allows students to drag-and-drop, as well as offer other educational features such as quizzes and flashcards. Future plans also include tailoring this application to middle and high school students.

HIV Subtyping Tool: This project aims to provide an efficient and effective tool for identifying HIV Subtypes present in a given genome sequence. Existing tools for this task are either efficient or effective, but not both. This group consisting of three undergraduates from Biochemistry, Cell and Molecular Bio, and Physiology, are developing a web application that is user-friendly, accurate, and a highly responsive tool for researchers working on this extremely aggressive and dangerous disease. To achieve this they are learning machine learning, and web development. A simple but powerful algorithm with Levenshtein distance [30,31] at its core, has been developed for the matching problem. A functional prototype has also been designed for a web application.

Fluotify: The goal of this project is to develop a cell tracking program for 3D images of tissue samples. Such a program can drastically improve our understanding of cell biology. The ability to study cell migration can, for example, shed light on why some cells differentiate into neurons while others don't. This project was inspired by the research being conducted by one of its members as part of her graduate thesis. This team, consisting of one graduate student in Cell and Molecular Biology, and three undergraduates in Applied Mathematics, and Biochemistry, learned image processing of 2D images to identify cell boundaries, and then count the distinct cells. The team will then track the identified cells across the z-plane, so as to work with 3D images.

Through these projects, students are demonstrating how effectively they can learn when they are committed to the project goals. The following students' quotes convey this sentiment;

"This course has offered me experience in more than just coding. We applied teamwork and with that came project and time management. We presented our PowerPoints numerous times … Aside from that, I've also gained experience in communication as we've had to keep in contact with many people who help us make this project possible."

"I learned this semester is to have an open mind to new ideas."

"This led to learning leadership skills to ensure everyone had a chance to speak, and were focused at the task at hand."

"Going in I knew CSc 698 involved project design, collaborations, and programming. I was surprised to hear that programming would not be the first step, but now it makes sense because our projects are as individual as we are."

"The style that entails weekly meetings, co-working with classmates, working on one project as opposed to multiple small ones, and developing a tool, has provided me with a learning experience I had not had at SFSU or for that matter, as a student."

Project-based courses are not easy, but students are learning important life-long lessons. One team mentioned this in particular. "While we had our share of hiccups and setbacks, we found our momentum near the end of the semester. In chemistry there is something known as activation energy which is the energy needed to push a reaction to completion. I like to think that the first half of the semester was the activation energy needed to push our project to completion."

It is also important to note that the diversity of the PINC students was also noted by others. For example, a PINC student stated the following; "At one point in our meeting with the S.H.S, the Director and supporting staff commented on how radical it was to see women in the field of Computer Science."

B. Student self-reflection CS knowledge survey

We have surveyed PINC students' self-reflection on content knowledge. Even though we observed students growth on CS knowledge through the course exams, program projects and especially through 698 projects, we also wanted to understand a bit more about their confidence level in CS. As shown in figure 1, students self-reported that the concepts that they are frequently exposed over the course work improved significantly. This survey informs faculty that terms like "abstraction" should be more frequently discussed in the class. Students who thought they know abstraction at the first semester lost their confidence in the second semester. In general, students have improved familiarity and concepts in CS foundation knowledge in steady ways and this aligns with progress in programming projects and exam performance.

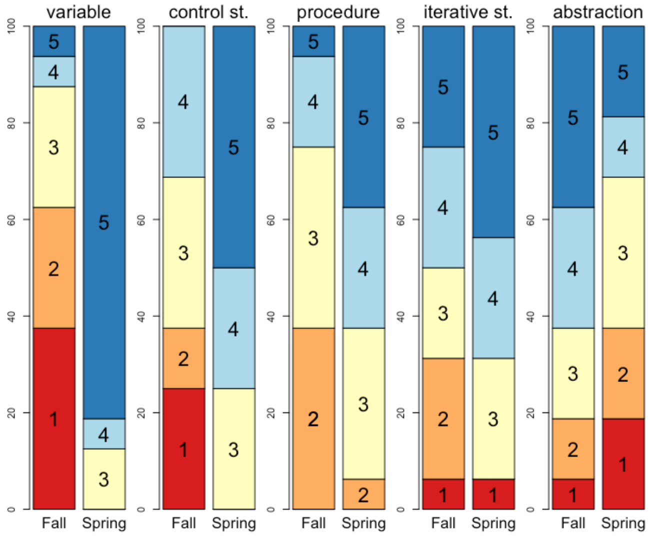


Figure 1. Student self-reflection survey on fundamental CS concepts. 1 indicates "not at all" and 5 indicates "completely comfortable". Survey was performed in Spring 2017 and Fall 2017. These surveys were

performed on cohort 1 progressively (maybe viewed as pre and post test).

C. The PINC program demonstrates that our approach to broadening CS participation works.

We launched PINC in 2016, and the results thus far demonstrate that it is effective at attracting a diverse group of students to study CS. In PINC, CS and biology faculty worked together to recruit biology majors to the Computing Applications emphasis. Thus far, two cohorts, totaling 59 students, have participated in the program. Half of these students identify as Black/African American and Latino, and three-quarters are women. This level of diversity even exceeds the diversity present within biology programs. In particular, the PINC program did an excellent job recruiting Black/African Americans, which represent a very small number of students in CS and Biology programs.

D. Recruitment of students for the PINC program

Recruitment efforts for the PINC program consisted of posters in the hallways of the biology department and announcements in the weekly Biology Department email newsletter that is sent to all Biology majors. PINC faculty also visited various classes to make announcements in person. A PowerPoint slide announcing the program was also sent via email to all biology professors in the hope that they would share this information and encourage their students to sign up for the PINC program. In the spring of 2016 (the semester before the first PINC class started), the PINC PIs organized two campus meetings to inform students about the PINC program. They also launched a website (https://pincsfsu.com/).

In all the recruitment announcements it was clear that all biology students were encouraged to apply and that no prior coding experience was needed nor a laptop as those students will be provided laptops if needed. Finally, it was advertised that CS courses would be supported by peer mentors. To reduce stereotype threat, images of women and diverse students were used on all advertisements and on the website [19-21]. Students applied to the program by filling out a "Wufoo" survey [32].

E. Demographics of the PINC students

60 students took one or more PINC classes in the first three semesters of the program (fall 2016 and spring and fall 2017). Out of these 60 students, 43 identified as female (71%), 16 as male (27%) and 1 (2%) as genderqueer. The number of female students is similar amongst biology majors (66%) but much higher compared to CS majors (16%). Thus, the PINC program reflects the gender distribution found in the biology department much more than the CS department (figure 2).

Out of 60 students, 10 were black (17%, compared to 4% (Bio) and 3% (CS)), 19 latinx (32% compared to 26% (Bio) and 15% (CS)), 5 white (8% compared to 15% (Bio) and 20% (CS)), 23 asian (38%, compared to 35% (Bio) and 38% (CS)) and 3 other (5%). The PINC program has a larger than expected number of black students, whereas the number of latinx students is comparable to the numbers among biology majors. Out of 55 students with known ethnicity, the PINC program has 28 students from UR groups, which represents half of the students. This is surprising because we didn't specifically aim to recruit UR students, and the population source was biology majors, which is approximately 30% UR.

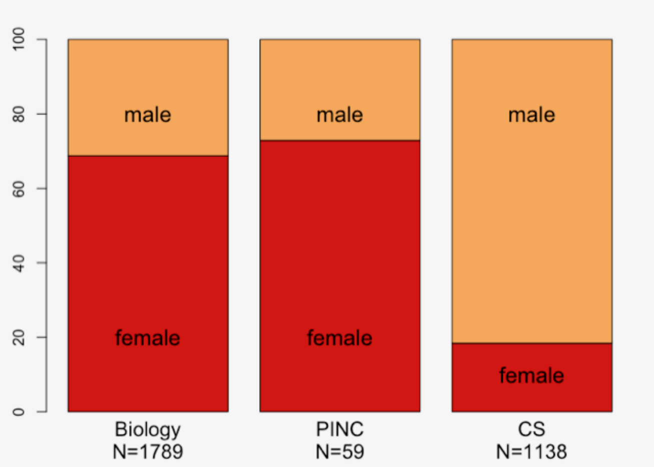


Figure 2. Gender statistics on Biology, CS and PINC program. For simplicity, male and female are used in this chart.

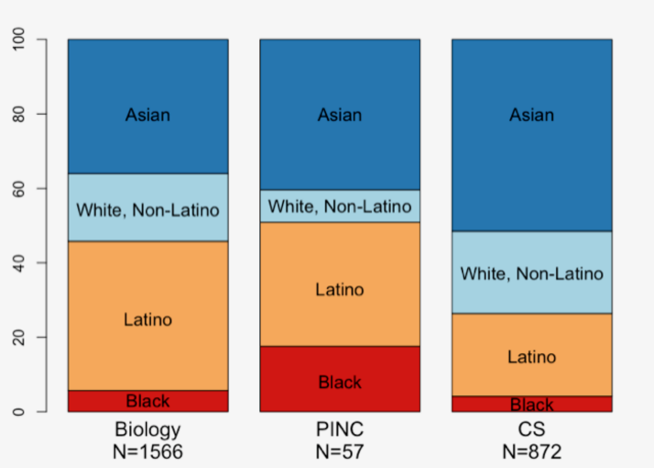


Figure 3. Ethnicity statistics on Biology, CS and PINC program. For simplicity, Asian, White, Latino and Black are used in this chart.

Since the Biology Department offers a Bioinformatics course, there was a concern that by offering CS courses that cater to biology majors, that we would potentially impact enrollments in Bioinformatics. However, this was not the case. The Bioinformatics classes were completely full in both 2016 and 2017.

The vast majority of the students in the PINC program had little to no coding experience. The application to the PINC program asked students about whether they had prior coding experience. Of the 47 students who answered this question, 18 had no coding experience at all (38%), 6 (13%) had taken a class in high school, 12 (26%) had taken a class in college (some of those were allowed to skip the first PINC class) and 11 (23%) had some other experience (e.g. online class). For three-quarters of the PINC students this was their first CS class in college.

F. Motivation to apply to and stay in the program

Once the program began, another survey was administered to the PINC students to better understand their motivation for participating in the PINC program. The survey question asked the following; "What were the primary reasons you INITIALLY decided to participate in the PINC program". 16 out of 35 respondents (45%) mentioned "job" or "career opportunities". Other motivations mentioned were "learning to code" (9 respondents, 26%) and 8 respondents mention that they plan to" apply CS skills to biological topics" (23%). In addition, their written responses indicated that our recruitment strategy helped students overcome barriers that had previously prevented them from pursuing CS education. Sample answers include the following:

- *I wanted to approach a field that I felt was intimidating for women and students in health/biology. I wanted to learn how to code, but was deterred from it at a younger age, because I did not excel in math.* (**overcoming stereotype threat, imposter syndrome**)
- *I initially decided to join PINC after learning how the biology field is trending towards solving problems with computer science answers. Also, because it is taught to high-schoolers, and I missed when they added that to the [high school] curriculum, so I feel behind.* (**overcoming irrelevant content, imposter syndrome**)

This survey also revealed that our pedagogical approach, which focused on presenting CS fundamentals in the context of students' interests and in an inclusive and nurturing manner, engaged them. Sample student comments include:

- *The project we are working on as a group is super interesting, so the thought of being able to contribute to such a cool project keeps me extremely motivated to keep learning and working with others to solve scientific problems.* (**providing relevant content to keep learners motivated**)
- *Working with others that do not have a background in CS helps me feel like I am not the only one struggling.* (**providing inclusive peer group to combat imposter syndrome, stereotype threat**)
- *My favorite experience from the PINC program is meeting with my mentor every week. The mentors I have had so far are caring and very patient with me and other PINC students.* (**providing nurturing support from mentors to combat imposter syndrome, stereotype threat)**

G. Persistence

Thus far, 60 students have taken at least one CS class. It is still too early to know how many students will complete the minor in 2018 at the time of this writing. We looked at the students who started in the fall of 2016 or spring of 2017 (n=31 students). 14 of these students are on schedule to complete the PINC minor in spring 2018, 1 student switched to the heavier 21-unit CS minor, 2 students are taking classes but do not want to complete the minor, and

14 students left the program after one or two classes.

Therefore, 15 out of 31 students are expected to complete a CS minor in 2018. In previous years only 2 or 3 students obtain a minor in CS, so this is a 5-7 fold increase.

H. Why students don't persist

Given that 14 students left the program after taking one or two CS courses, we were interested in understanding why they made this decision. It seems that some students did not have the intention of completing the minor from the onset. For example, one student was already enrolled in two other minors. Another student was finishing up her Master's thesis and left the university. We surveyed the students who left the program and received only 5 responses. Four of the five respondents mentioned the amount of time it took to take the CS classes in addition to their regular course load. For example, one student wrote: "I needed to invest more hours than I expected. I enjoyed the class, and I do feel like I learned a lot, but I was worried it would interfere with my other course work."

I. Mentors Learning Outcome

We found that both the near peer and research mentors benefited tremendously from this experience. This confirms a prior study that showed that mentors benefit as much if not more than their mentees [33]. The near peer mentor learning outcomes exceeded our expectations. In particular, the research mentors (698 group project mentor) had a very unique interdisciplinary teaching and research experience, which was very impactful. To better understand the learning outcomes, the mentors were asked to complete a self-reflection at the end of the semester. Excerpts of the research mentors self-reflections are the following:

"It was an absolute pleasure working for the PINC program this semester. It was a challenging and rewarding experience and I have learned great leadership and project management skills in my time with the program. I'm very proud of my team and the progress we have made on our project." "Until this mentoring started, I was only able to develop applications, but PINC gave me a completely different perspective. It taught me how to assist someone who is just learning computer science to build a functional product. Students from all teams had a vision that they wanted to implement, and all of the project ideas were very innovative. These things led me to understand that there is so much more to Software Engineering than just creating applications - there is a huge amount of ideas and a wide learning curve behind all of this." "From being a PINC mentor, I realized that guiding and managing people in a project is as important as developing and completing it. The journey and experience matters a lot in this. I had stages in my project where my team members were happy with every single output or every concept that they learnt. The confidence that my team gained that they will be able to carry on with the project based on what they learnt during the program was a good sense of achievement for me. It felt good to impart my knowledge on to someone who will find it helpful. Since I am a person who did not have any mentoring or project management experience before, my role here as a mentor helped me gain insight to these things." "I like learning about people and how they learn. Participating in these kinds of activities is great exposure to different viewpoints and perspectives. The PINC program was motivating to me because of the people. Meeting so many interesting people, hearing their stories and goals was really encouraging to me."

VI. Conclusion

The PINC program is a new model for how to introducing computing to women and UR science majors. This program uses high impact practices to successfully create a welcoming environment where women and UR students feel supported in taking CS classes that align well with their interests. Moreover, the student peer and research mentors also show substantial benefits from participating in the program. This model can be disseminated nationally. Our goal is to continue to increase the diversity of individuals seeking CS education so that it better reflects our national demographics. Our ability to recruit and retain women and UR students will have a substantial impact on the diversity of the CS workforce. An education that includes CS enriches individuals' critical thinking skills [54], boosting their ability to succeed in other aspects of their professional and personal lives. PINC is aiming to train students to develop competencies and transferable skills. We are developing more thorough assessment tools to study which components create the desired outcome.

Reference

[1] Bureau of Labor Statistics (2017) Occupational Outlook Handbook, 2016-17 Edition, Computer and Information Research Scientists.

[2] Tidwell A (2013) Tech companies work to combat computer science education gap.

[3] Henke N, Bughin J, Chui M, Manyika J, Saleh T, et al. (2016) The Age of Analytics: Competing in a Data-Driven World. McKinsey Global Institute

[4] Marx V (2013) Biology: The big challenges of big data. Nature 498: 255-260.

[5] Larson S (2014) Why so few women are studying computer science. https://readwrite.com/2014/09/02/women-in-computer-science-why-so-few/

[6] Flapan J (2015) Equity in computer science is vital for California. http://www.sandiegouniontribune.com/opinion/commentary/sdut-equity-in-computer-science-education-vital-for-2015oct15-story.html

[7] Ortutay B (2017) Why is there so much attention but so little progress for diversity in tech http://www.chicagotribune.com/bluesky/technology/ct-diversity-tech-little-progress-ap-bsi-20170125-story.html

[8] Medin D, Lee C, Bang M (2014) Point of view affects how science is done. Scientific American http://www.scientificamerican.com/article/point-of-view-affects-how-science-is-done/

[9] Lee D (2014) Scientists draw on personal experience to guide their curiousity. Scientific American http://www.scientificamerican.com/article/scientists-draw-on-personal-experience-to-guide-their-curiosity/

[10] Kang C, Frankel T (2015) Silicon Valley struggles to hack its diversity program. https://http://www.washingtonpost.com/business/economy/silicon-valley-struggles-to-hack-its-diversity-problem/2015/07/16/0b0144be-2053-11e5-84d5-eb37ee8eaa61_story.html?utm_term=.071fe964ea19

[11] Cheryan S, Plaut VC, Davies PG, Steele CM (2009) Ambient belonging: how stereotypical cues impact gender participation in computer science. J Pers Soc Psychol 97: 1045-1060.

[12] Murphy MC, Steele CM, Gross JJ (2007) Signaling threat: how situational cues affect women in math, science, and engineering settings. Psychol Sci 18: 879-885.

[13] Good C, Aronson J, Harder JA (2008) Problems in the pipeline: Stereotype threat and women's achievement in high-level math courses. Journal of Applied Developmental Psychology 29: 17-28.

[14] Holleran SE, Whitehead J, Schmader T, Mehl MR (2011) Talking shop and shooting the breeze: A study of workplace conversation and job disengagement among STEM faculty. Social Psychological and Personality Science 2: 65-71.

[15] Gross L (2004) The interface of mathematics and biology interdisciplinarity and the undergraduate biology curriculum: Finding a balance. Cell Biol Educ 3: 85-87.

[16] Laursen L (2008) No, You're Not an Impostor. Science Careers http://www.sciencemag.org/careers/2008/02/no-youre-not-impostor

[17] Leslie SJ, Cimpian A, Meyer M, Freeland E (2015) Expectations of brilliance underlie gender distributions across academic disciplines. Science 347: 262-265.

[18] Hulleman CS, Godes O, Hendricks BL, Harackiewicz JM (2010) Enhancing interest and performance with a utility value intervention. Journal of Educational Psychology 102: 880.

[19] Steele CM (1997) A threat in the air. How stereotypes shape intellectual identity and performance. Am Psychol 52: 613-629.

[20] Schmader T, Johns M, Forbes C (2008) An integrated process model of stereotype threat effects on performance. Psychological Review 115: 336.

[21] Johnson SE, Richeson JA, Finkel EJ (2011) Middle class and marginal? Socioeconomic status, stigma, and self-regulation at an elite university. Journal of Personality and Social Psychology 100: 838.

[22] Drago-Severson E, Helsing D, Kegan R, Popp N, Broderick M, et al. (2001) The power of a cohort and of collaborative groups. NCSALL Focus on Basics 5.

[23] Tien L, Roth V, Gates A, Roach S (2002) Implementation of a peer-led team learning instructional approach in an undergraduate organic chemistry course. Journal of Research in Science Teaching 39: 606-632.

[24] Gates A, Delgado N, Bernat A, Cabrera S (2006) Building affinity groups to enable and encourage student success in computing. Women in Engineering ProActive Network.

[25] Kephart K, Villa E, Gates A, Roach S (2008) The affinity research group model: Creating and maintaining dynamic, productive, and inclusive research groups. CUR Quarterly 28: 13-24.

[26] Ten Cate O, Durning S (2007) Dimensions and psychology of peer teaching in medical education. Medical Teacher 29: 546-552.

[27] http://appinventor.mit.edu/explore/

[28] Norman, G. T., and Henk G. Schmidt. The psychological basis of problem-based learning: a review of the evidence. *Academic medicine* 67.9 (1992): 557-565.

[29] https://health.sfsu.edu/

[30] http://www.levenshtein.net/

[31] E. S. Ristad and P. N. Yianilos, "Learning string-edit distance," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 5, pp. 522-532, May 1998.

[32] https://www.wufoo.com/

[33] Trujillo G, Aguinaldo PG, Anderson C, et al. Near-peer STEM Mentoring Offers Unexpected Benefits for Mentors from Traditionally Underrepresented Backgrounds. Perspectives on undergraduate research and mentoring : PURM. 2015;4(1):http://blogs.elon.edu/purm/files/2015/11/Riggs.GT-et-al-PURM-4.1.pdf.