Promoting Diversity in Computing

Anagha Kulkarni, Ilmi Yoon, Pleuni S. Pennings, Kazunori Okada, and Carmen Domingo
San Francisco State University
San Francisco, CA, USA
{ak,ilmi,pennings,kazokada,cdomingo}@sfsu.edu

ABSTRACT

In this paper we present a pilot program at San Francisco State University, Promoting INclusivity in Computing (PINC), that is designed to achieve two goals simultaneously: (i) improving diversity in computing, and (ii) increasing computing literacy in data-intensive fields. To achieve these goals, the PINC program enrolls undergraduate students from non Computer Science (non-CS) fields, such as, Biology, that have become increasingly data-driven, and that traditionally attract diverse student population. PINC incorporates several well-established pedagogical practices, such as, cohort-based program structure, near-peer mentoring, and project-driven learning, to attract, retain, and successfully graduate a highly diverse and interdisciplinary student body. On successful completion of the program, students are awarded a minor in Computing Applications. Since its inception 18 months ago, 60 students have participated in this program. Of these 73% are women, and 51% are underrepresented minorities (URM). 74% of the participating students had nominal or no exposure to computer programming before PINC. Findings from student surveys show that majority of the PINC students now feel less intimidated about computer programming, and vividly see its utility and necessity. For several students, participation in the PINC program has already opened up career pathways (industry and academic summer internships) that were not available to them before.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; **CS1**; **Computing literacy**;

KEYWORDS

diversity, gender gap, X+CS, cohort-based programming, near-peer mentoring, project-driven learning

ACM Reference Format:
Anagha Kulkarni, Ilmi Yoon, Pleuni S. Pennings, Kazunori Okada, and Carmen Domingo. 2018. Promoting Diversity in Computing. In *Proceedings of 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'18).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3197091.3197145

ITiCSE'18, July 2–4, 2018, Larnaca, Cyprus

ACM ISBN 978-1-4503-5707-4/18/07…$15.00
https://doi.org/10.1145/3197091.3197145

1 INTRODUCTION

San Francisco State University (SFState) is one of the most diverse universities in the United States with a long-standing commitment to attracting, retaining, and graduating a highly diverse student body. Of the students reporting their gender and ethnicity in Fall 2016, 58% were female, and nearly 50% identified with an underrepresented minority groups (Black/African Americans, American Indians, Alaskan Natives, Native Hawaiian, Pacific Islander, and Latino). These trends carry over to the College of Science and Engineering (COSE) at SFState where 49% are female students, and 48% are underrepresented minority group (URM). However, when we further zoom into the Computer Science department (CS) under COSE, the trends are starkly different. Only 15% of the students are female, and 28% are URM. Unfortunately, these numbers are representative of the trends across the country in CS. In 2014, only 18% of bachelor degrees in CS were awarded to women, only 7% to African Americans, and only 10% to Hispanics/Latinos[1], these numbers are far lower than one would expect based on these groups' share of the population [2, 7]. Indeed, CS remains one of the least diverse STEM disciplines. In addition to signaling a substantial problem with educational equity, this lack of diversity in the CS workforce makes it difficult to carry out innovative work and avoid unnecessary mistakes[2].

This problem of gender and ethnic disparity in Computer Science becomes even more acute when we factor in that the demand for computer scientists and programmer greatly exceeds the *supply*. US Bureau of Labor Statistics estimates that by 2024 1.1 million CS jobs will exist, but only 450,000 CS graduates will be available to fill them [1]. The dominant force driving this demand is the unprecedented amount of digital data being generated by nearly every field of study. Data scientists and programmers who have domain knowledge and CS training are sorely needed to take on the big data challenges arising in many fields. These observations are the motivations for a brand new pilot program, Promoting INclusivity in Computing (PINC), underway at SFState[3]. One of the key ideas of PINC is to bring computer science to students majoring in other fields, instead of expecting them to change their career course to CS. We also see PINC as a *back-door* solution to the problem of gender and ethnic inequalities in CS. We recruit PINC students from majors such as Biology, and Biochemistry which enjoy fairly high enrollments by URM and women. We hope that PINC will slowly shift the demographics in CS and help shatter the prevalent stereotypes about CS in regards to (i) its difficulty, (ii) its abstraction from real-world, and (iii) its lack of diversity.

[1] https://readwrite.com/2014/09/02/women-in-computer-science-why-so-few/
[2] http://www.chicagotribune.com/bluesky/technology/ct-diversity-tech-little-progress-ap-bsi-20170125-story.html
[3] https://pincsfsu.com

2 PROGRAM DESCRIPTION

The PINC program is designed around three pedagogical techniques: (i) cohort-based program structure, (ii) near-peer mentoring & affinity research group, and (iii) project-based learning. We describes these techniques and our motivation to adopt them in subsequent sections. We start by describing the PINC curriculum.

2.1 PINC Curriculum

The PINC curriculum consists of five 3-unit courses designed to progressively develop students' computing knowledge such that they become competent to pursue interdisciplinary CS careers. The program begins with three lecture courses (306, 220, and 307) that provide strong CS foundation through rigorous but meaningful programming assignments that align with their interests. CSc 306 (An Interdisciplinary Approach to Computer Programming) teaches the basic building blocks of programming. CSc 220 (Data Structure and Algorithms) introduces data structures, algorithms and algorithmic thinking, and basic performance analysis. CSC 307 (An interdisciplinary Approach to Web Programming) teaches the basics of Web programming covering HTML, PHP, Python, SQL, and MySQL databases. After these three courses, students apply their CS learnings to develop a solution to a problem of substantially large scope. This is done through the final two courses (698a and b) where the focus is on group projects. As students progress through these five courses, the expected learning outcomes (Section 2.1.1) and their corresponding competence level are mapped out in Table 1. Students who successfully complete the program are awarded a minor in Computing Applications. We are keenly aware that fairly limited CS learning can be achieved in five courses. However, the goal here is to get more students (especially minorities) *hooked* to CS, and set them on a path where they feel confident and prepared to explore CS further. We see PINC as just the beginning of a very rich and fulfilling journey for these students.

2.1.1 Student Learning Outcome. At the end of the PINC program, a student will be able to:

(1) demonstrate basic skills in computer programing using a widely used language, such as, Java or Python.
(2) develop simple software programs that employ core concepts, including lists, stacks, queues, trees, tables, graphs, recursion, iteration, sorting, search, and hash table.
(3) program database applications by applying core concepts of database theory and management.
(4) develop simple internet applications using web development technology.
(5) design, develop, and manage software prototype creation process with the goal of solving a problem from their field, or providing an improved solution to the problem.

2.2 Cohort-based Program Structure

Cohort-based programs organize students into relatively small groups (12 to 25), where students from each cohort start and finish their degree together [8]. Cohort programs offer community building and collaborative opportunities with people who have similar goals and backgrounds. This creates camaraderie among the students, and helps them overcome academic challenges. Our motivation for adopting a cohort-based structure came from knowing the commonly held myth by non-CS students about introductory CS course(s). "The introductory CS courses had many students with programming experience, and thus were not true beginner". This type of imposter syndrome is especially prevalent among female students, URM, and first-generation college students [16]. Cohort-based organization where majority of the students in the class are non-CS majors, provides a robust mechanism to combat such biases. Moreover, if the environment is diverse, it can substantially reduce stereotype threat [11, 14, 18, 19], and the social isolation that minorities feel in a typical CS classroom [4, 9].

2.3 Near-peer Mentoring & Affinity Research Group

Peer-led team learning (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 [12, 13]. PINC program integrates a form of PLTL, *near-peer mentoring*. Seniors and graduate students in the CS department who had demonstrated strong academic performance, and strong interpersonal skills were selected as mentors for the PINC courses. We refer to these students as near-peer mentors, and not peer mentors, because they have received significantly more CS training than PINC students. The near-peer mentors meet with the students that they were paired up with every week. These weekly session provided dedicated *office hours* for the PINC students where they ask questions about topics covered in class, and about homework assignments. Mentors also used these sessions to go over topics that were introduced during lectures but needed reinforcing. This format was followed for the first three courses of the program (306, 220, 307).

To support the PINC students in their capstone project (CSC 698a and 698b) during the second year of the program, we used an Affinity Research Group (ARG) model [10]. Each ARG consisted of three to four PINC students supervised by a CS Senior or graduate student who acted as their research mentor. These research mentors received 3 units of academic credit for participation in this program. Each group met with their mentor every week for at least an hour to define, plan, develop, and manage the project. The mentors used their CS training and experience to facilitate learning and provide technical support to their group.

Both PLTL and ARG support cooperative learning, positive interdependence, face-to-face interactions, individual accountability, and group processing [12, 15]. In particular, the small distance between mentor/teacher and learner takes advantage of the cognitive and social congruence between the two populations [20]. This mentorship component of PINC is designed to mitigate the *CS difficulty* stereotype, where students come in with a preconceived notion that CS courses are impossibly hard and thus bad for their GPA. The mentors also enhance the community building aspect of the program, and strengthen the support network for the students.

Many of the PINC mentors had no prior mentoring experience. We provided them ongoing training through a series of monthly workshops facilitated by one of the PINC PIs. The goals of these

Table 1: Curriculum Map with SLOs ("I" denotes *Introduced*, "D" denotes *Developed*, and "M" denotes *Matured*. SLOs are defined in Section 2.1.1).

	SLO 1	SLO 2	SLO 3	SLO 4	SLO 5
CSC 306: An Interdisciplinary Approach to Computer Programming	I				
CSC 220: Data Structures & Algorithms	D	I			
CSC 307: An Interdisciplinary Approach to Web Programming	D	D	I	I	
CSC 698a: Topics is Computing I	D	D	D	D	I
CSC 698b: Topics in Computing II	M	M	M	M	D

workshops were to (i) to create an environment where the mentors assume ownership over the mentoring component of the program, (ii) to develop strategies to identify and resolve learning challenges that their mentees were facing, and (iii) to co-discover effective tutoring methodologies to resolve specific student issues. Mentors were also encouraged to share their personal stories on how their CS studies started as well as their own struggles with the material. Until now, 81% of the mentors have been female or URM.

2.4 Project-based Learning

The third pedagogical strategy that PINC adopts is project-based learning [3, 17, 21] where real-life, interdisciplinary project topics chosen by the students themselves drive their hands-on and collaborative learning experience. These projects provide them a unique opportunity to apply their learnings from the previous courses to a complex problem over an extended period of time (2 semesters). This promotes deeper understanding of the core concepts, better appreciation of the breadth of the field, and longer retention of concepts and skills [5, 6]. These courses are instructed by one of the PINC PIs who provides *distant supervision* for each project. The goal is to provide creative independence and a scaffolding opportunity to the students while also (i) keeping them on track, (ii) bringing in domain and technical experts as per projects' needs, and (iii) conducting periodic evaluation of each student and project. Section 3.4 describes six group projects that were defined and prototyped during the inaugural offering of CSC 698a in Fall 2017. In Spring 2018, these project have continued to progress toward fully developed and thoroughly tested applications.

3 PRELIMINARY FINDINGS, AND LESSONS

The PINC program launched in Fall 2016 with focus on Biology as the source department, and has been active every semester since. Biology students were chosen for this pilot because 1. the student population in this major has been consistently diverse, 2. two of the PINC PIs are from Biology department, and 3. the need for computing literacy is becoming acute in this field as it becomes increasingly data-driven. The PINC program was advertised to the students via the means of posters in the hallways of the Biology department, through weekly email newsletters, and through presentations during classes. In all the recruitment announcements it was highlighted that no prior coding experience was needed, it was not necessary to own a computer, and that all courses would be supported by mentors and PINC PIs. To reduce stereotype threat, images of women and diverse students were used on all advertisements and on the website [14, 18, 19].

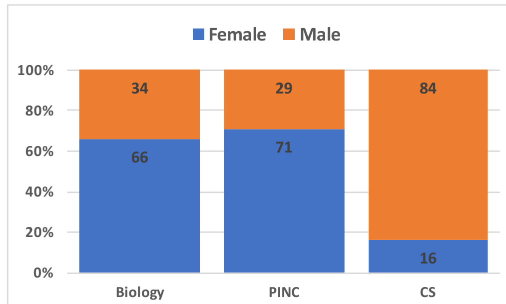

Figure 1: Gender distribution of students in Biology department, PINC program, and CS department

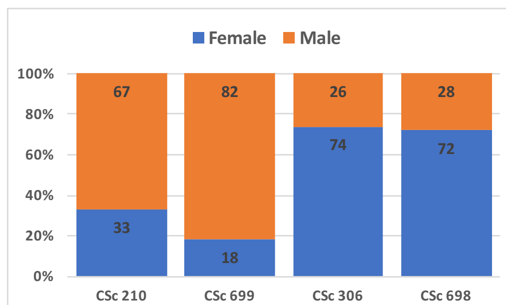

Figure 2: Gender distribution of students at the start and end of the CS bachelors program, and PINC program. For CS bachelors, start: CSc 210, and end: CSc 699. For PINC program, start: CSc 306, and CSc 698.

3.1 Impact on Gender and Ethnic Diversity

Figure 1 provides the gender distribution of students in the Biology department, the PINC program, and the CS department at SFState. As one would expect, the composition of PINC program is more representative of the source department, Biology. Although, PINC program has slightly higher percentage of females than even the Biology department. Compared to the CS department, PINC program has a starkly different gender distribution. Female students only account for a sixth of the population in the CS department.

Figure 2 compares the gender distribution at the start, and end of the program for PINC and CS Bachelors. CSc 210; Introduction to Computer Programming, is the first course that CS majors have to take, and is comparable to PINC's first course: CSc 306. CSc 699 is an

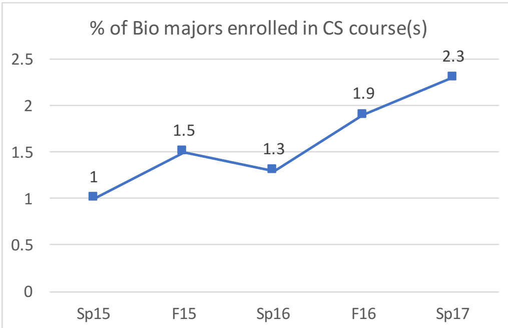

Figure 3: Percentage of Biology majors enrolled in 1+ Computer Science course(s). Pre-PINC period: Sp15 - Sp16.

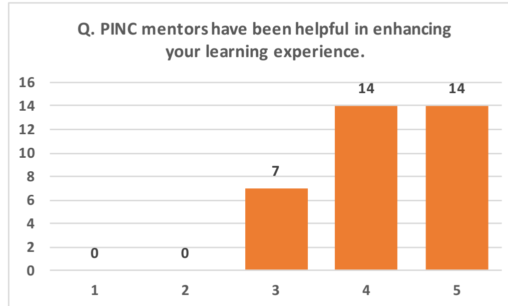

Figure 4: Survey results for question assessing utility of PINC mentors. Five-level Likert scale, 1 corresponded to *Strongly Disagree*, and 5 to *Strongly Agree*

independent study course that CS seniors take, and is comparable to PINC's CSc 698. Figure 2 shows that the already small population of female students at the start of the CS program (CSc 210) shrinks by 15 percentage points by the end of the program (CSc 699). Whereas for the PINC courses, the drop in female students (2 percentage points) is fairly small.

Similar trends are observed in the numbers of URM students. In the traditional CS bachelors program, the starting population of URMs is already small (34%) and it drops to 30% by upper-division courses. Whereas in the PINC program, the URM students account for 56% of the initial enrollment and this number persists till the final courses of the program.

We plan to use the gender and URM distribution in PINC as an inspirational tool for CS students. Toward this goal, we plan to organize periodic informal events that bring together PINC and CS students. Seeing a large and thriving female population can motivate other female students and URMs, especially who are at early stages in the program, to continue on.

Overall these trends indicate that PINC is attracting, and retaining female and minority students in a computer science program. Reaffirming these trends with more data and statistical analyses will become possible as more data is gathered over time.

3.2 Impact on Increasing Computing Literacy

Figure 3 plots the percentage of Biology majors that were enrolled in one or more CS course before the PINC program launched (Spring'15 through Spring'16), and since the launch (Fall'16 and Spring'17). A small but encouraging increase in the percentage of students taking CS courses is observed since the start of the PINC program. In Fall'16, 20 Biology students were enrolled in a CS course. That number increased to 39 by Spring'17.

Typically, 92.3% of Biology majors at SFState do not take a single CS course. 6.3% enroll in one CS course, mere 1.0% enroll in two, and 0.5% enroll in three or more CS courses. This indicates that there is low precedent among Biology students at SFState about taking multiple CS courses. In fact, the vast majority of the students never enroll in any CS courses. The PINC program asks Biology students to take a major departure from this trend, as they have to earn credits for five CS courses to successfully complete the program. As such, it was not trivial to recruit Biology students to PINC program. We ran an intense ad campaign, and accommodated strong students even when they could not commit to completing the the program. The goal was to gain the critical mass necessary to make the program viable, and to promote the program to a student population that was not primed to computing.

In spite of this background, 15 out of the 31 students who started PINC program in Fall'16 or Spring'17, are on schedule to complete the program in Spring'18 or Fall'18. These 15 students have successfully completed three to four CS courses out of the five in PINC curriculum, by the start of Spring'18. This is encouraging for two reasons. One, it is an unprecedented accomplishment in the Biology department. Second, students were able to achieve this while taking their major courses and GEs.

A deeper look into student attrition from PINC, showed two common reasons: (i) student finished their bachelors program, and (ii) could not balance the demands of the major and the PINC program, and thus had to choose. The first reason can be handled simply by selecting students who are at least four semesters away from graduation. Once the program is established we can be more selective and enforce this enrollment criterion. For the second problem (balancing major and PINC program) we are working on a multipronged solution. First, we are in talks with the Biology department to have a few of the PINC courses counted toward the Biology bachelors program. Second, we plan to offer some PINC courses during summer so as to spread out the workload for the students. Third, demonstrate the benefits of the PINC program. Toward this goal, we are organizing poster sessions and social events where senior PINC students will present their work to students who are just starting the program, and also to Biology Freshmen, and Sophomores.

Overall, we strongly believe that the PINC program has initiated a path through which computing literacy can be improved even in majors with low precedent for computing education.

3.3 Utility of PINC Mentors

Figure 4 provides the distribution of student responses to the following survey question: "PINC mentors have been helpful in enhancing your learning experience." Student responses were measured on a five-level Likert scale where 1 corresponded to *Strongly Disagree*, and 5 to *Strongly Agree*. 80% of the students agreed or strongly agreed that mentors improved their learning experience. When asked about one thing that they liked best about the program, some of them responded with: *"Having a mentor is a great help. I really appreciate their input.", "We get to work with experienced mentors who are very knowledgeable", "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.", "One thing I like are the weekly mentor meeting."* The complete survey and the results are made available on the program website[4]. 94% of the students enrolled in the capstone project-based course (CSc 698a) felt that having a dedicated mentor assigned to each project was crucial to the success of the project. All the signs indicate that mentors play a crucial role in making PINC successful.

At the same time, the mentors themselves benefit tremendously from this experience. Here are some of the responses that mentors provided on their exit survey. *"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. 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."*

3.4 Utility of Project-based Learning

The first of the two capstone courses (CSc 698a) ran for the first time in Fall'17. 21 students enrolled in the course were organized into 6 groups, consisting of 3 to 4 students in each group. These groups were formed by the instructor based on students' (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. In addition to the weekly lectures, each group met with their mentor once a week for 1.5 hours. Skills and tools that are useful to all the groups, such as, analytical thinking, problem solving, project planning, and version controlling, were taught and discussed during the lecture time. Over the period of just one semester, the groups have defined and prototyped six excellent and ambitious projects (briefly described below). They have done so fairly independently with periodic inputs from the instructor, and close guidance from the mentors. Each group presented their project topic and expected project timeline, in the form of a poster, early on in the term. This was followed by two more presentation opportunities at different stages of the project. These provided a forum for the students to receive feedback and to practice their presentation skills. In a survey conducted at the end of the semester, 83% of the students felt that working on a group project improved their learning. 100% of the students agreed that they could take on complex problems when working on a group project. Also, all of them expect this experience to help them in industry jobs. Only 6% thought that working on the group project lowered their productivity. A brief description of the six group projects defined by the students from CSc 698a is given next.

HealthMsgU: This project develops 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 identifying the struggles and societal challenges that patients experience. Three undergraduate students in Ecology, Cell and Molecular Bio, and Microbiology have developed a solid prototype for this web application that is complete with user-stories and multimodal data presentations.
Woozie: This project aims to develop a mobile application that estimates intoxication level of the user based on multiple inputs:- the number of drinks, speech quality, facial changes, and their geolocation. The goal is to promote responsible drinking using the ever-present mobile phone. This project requires ground-up mobile 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. The Woozie team developed a prototype for the app that will be expanded in the Spring 2018 semester.
GatorHealth: This project develops a mobile app, GatorHealth, for Student Health Services (SHS) at SFState. 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. Thus further developing communications and interpersonal skills.
CatGen: This project is focused on developing an educational web application, CatGen, for an introductory genetics course. The existing application that is commonly used in Genetics courses is a standalone desktop application that is extremely clunky and difficult to use. This experience motivated the CatGen group (three students from Biotech, Microbiology, and Cell and Molecular Bio) 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 more admired and complex 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.
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,

[4]https://github.com/PenningsLab/PINCSurveyData

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.

Fluotify: This project develops a cell tracking program for 3D images of tissue samples. The ability to study cell migration can, for example, shed light on why some cells differentiate into neurons while others don't. This team, consisting of one graduate student in Cell and Molecular Biology, and three undergraduates in Applied Mathematics, and Biochemistry, had to learn basics of image processing and how to use an image processing software (OpenCV) for this project.

Note that all of the above projects are inherently interdisciplinary, and required the students to obtain at least working knowledge of a variety of computing tools and technologies, such as, web and mobile development frameworks (Flask, PhoneGap) and libraries (Bootstrap), IDEs (Android Studio), image processing software (OpenCV), 3^{rd} Party APIs (Twitter, Google Maps), and classification algorithms from machine learning (scikit).

3.5 Utility of Cohort-based Structure

We see clear indication of the benefits that students derive from the cohort-based structure of the program. Students appreciate the community building aspect of cohorts: *"I really enjoy the small group experiences (how small the classes are). This has helped me form better relationships with my teachers and peers and allow me to get a more hands on and one on one approach to learning CS.", "The interaction we have with our mentors and classmates, it feels like a close community"*. They feel more comfortable in a more cohesive group with similar backgrounds: *"Working with others that do not have a background in computer science help me feel like I am not the only one struggling", "Being able to work with other Biology majors"*. They use the community as their support system to get them through the challenges: *"I have enjoyed the camaraderie of working with a cohort. We all get to suffer together!", "I really enjoy working with my cohort and sharing the pain of coding!", "I love the ability to interact with my classmates and having a mentor to guide me through my struggles."*

4 CONCLUSIONS AND FUTURE WORK

PINC is a new model for CS education with two key objectives: to improve gender and ethnic diversity in computing, and to increase computing literary among non-CS majors. This program uses evidence-based practices to successfully create a welcoming environment where URM and women 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 URM students will have a substantial impact on the diversity of the CS workforce. An education that includes CS enriches individuals' critical thinking skills, boosting their ability to succeed in other aspects of their professional and personal lives. PINC trains students to develop competencies and transferable skills. When asked if computer science makes them a better biologist, more than 80% of the PINC students responded highly positively. This indicates that students are starting to see computer science as an enabler. We plan to leverage this momentum and promote computing literacy to other science majors.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Bill Hsu, Dr. Alegra Eroy-Reveles, Dr. Kimberly Tanner, Emily Shindledecker, and Brooke Weinstein for their valuable inputs and support. The authors would also like to thank the anonymous referees for their thoughtful comments. This work is supported in part by the Technology Pathways Initiative: http://www.cawit.org/universities/.

REFERENCES

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

[2] Monica Babes-Vroman, Isabel Juniewicz, Bruno Lucarelli, Nicole Fox, Thu Nguyen, Andrew Tjang, Georgiana Haldeman, Ashni Mehta, and Risham Chokshi. 2017. Exploring Gender Diversity in CS at a Large Public R1 Research University. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education.* ACM, 51–56.

[3] Stephanie Bell. 2010. Project-based learning for the 21st century: Skills for the future. *The Clearing House* 83, 2 (2010), 39–43.

[4] Sylvia Beyer, Kristina Rynes, and Susan Haller. 2004. Deterrents to women taking computer science courses. *IEEE technology and society magazine* 23, 1 (2004), 21–28.

[5] Phyllis C Blumenfeld, Elliot Soloway, Ronald W Marx, Joseph S Krajcik, Mark Guzdial, and Annemarie Palincsar. 1991. Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational psychologist* 26, 3-4 (1991), 369–398.

[6] Beverley E Crane. 2009. *Using WEB 2.0 tools in the K-12 classroom.* Neal-Schuman Publishers.

[7] Adrienne Decker and Monica M McGill. 2017. Pre-College Computing Outreach Research: Towards Improving the Practice. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education.* ACM, 153–158.

[8] Eleanor Drago-Severson, Deborah Helsing, Robert Kegan, Nancy Popp, Maria Broderick, and Kathryn Portnow. 2001. The power of a cohort and of collaborative groups. *Focus on Basics* 5, 2 (2001), 15–22.

[9] Allan Fisher and Jane Margolis. 2002. Unlocking the clubhouse: the Carnegie Mellon experience. *ACM SIGCSE Bulletin* 34, 2 (2002), 79–83.

[10] Ann Gates, Nelly Delgado, Andrew Bernat, and Sergio Cabrera. 2006. Building affinity groups to enable and encourage student success in computing. *Women in Engineering ProActive Network* (2006).

[11] Geneva Gay. 2010. *Culturally responsive teaching: Theory, research, and practice.* Teachers College Press.

[12] David K Gosser, Mark S Cracolice, JA Kampmeier, Vicki Roth, Victor S Strozak, and Pratibha Varma-Nelson. 2001. *Peer-Led Team Learning: A Guidebook.* Virginia: Prentice Hall.

[13] Susan Horwitz, Susan H Rodger, Maureen Biggers, David Binkley, C Kolin Frantz, Dawn Gundermann, Susanne Hambrusch, Steven Huss-Lederman, Ethan Munson, Barbara Ryder, and others. 2009. Using peer-led team learning to increase participation and success of under-represented groups in introductory computer science. In *ACM SIGCSE Bulletin*, Vol. 41. ACM, 163–167.

[14] Sarah E Johnson, Jennifer A Richeson, and Eli J Finkel. 2011. Middle class and marginal? Socioeconomic status, stigma, and self-regulation at an elite university. *Journal of personality and social psychology* 100, 5 (2011), 838.

[15] KL Kephart, EQ Villa, Ann Quiroz Gates, and Steve Roach. 2008. The affinity research group model: Creating and maintaining dynamic, productive and inclusive research groups. *Council on Undergraduate Research Quarterly* 28, 4 (2008), 13–24.

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

[17] Thom Markham. 2011. Project based learning a bridge just far enough. *Teacher Librarian* 39, 2 (2011), 38.

[18] Toni Schmader, Michael Johns, and Chad Forbes. 2008. An integrated process model of stereotype threat effects on performance. *Psychological review* 115, 2 (2008), 336.

[19] Claude M Steele. 1997. A threat in the air: How stereotypes shape intellectual identity and performance. *American psychologist* 52, 6 (1997), 613.

[20] Olle Ten Cate and Steven Durning. 2007. Dimensions and psychology of peer teaching in medical education. *Medical teacher* 29, 6 (2007), 546–552.

[21] John W Thomas. 2000. A review of research on project-based learning. (2000).