A Cascading Mentoring Pedagogy in a CS Service Learning Course to Broaden Participation and Perceptions

Yasmin B. Kafai, Jean Griffin*, Quinn Burke, Michelle Slattery+, Deborah A. Fields, Rita M. Powell*, Michele Grab*, Susan B. Davidson, & Joseph S. Sun*
University of Pennsylvania Graduate School of Education
*Department of Computer and Information Science
+Peak Research
3700 Walnut Street
Philadelphia, PA 19106

kafai@upenn.edu, griffin@seas.upenn.edu, burkeqq@cofc.edu, peakresearch@usa.net, deborah.fields@usu.edu, rpowell@seas.upenn.edu, mgrab@seas.upenn.edu, susan@cis.upenn.edu, sunj@seas.upenn.edu

ABSTRACT

This study reports on the design, implementation, and evaluation of a service-learning course based on a "cascading mentoring" model linking together the faculty, administration, and undergraduates of an urban university's computer science department with local high school students. We present findings from surveys and post-interviews that illustrate undergraduates' and high school students' experiences in the program and how their perceptions of computing and mentoring changed based upon the outreach. In our discussion, we focus on the institutional and conceptual challenges of implementing the community service course within the university's computer science department, while also highlighting the learning opportunities for streamlining such a model for future iterations.

Categories and Subject Descriptors
K.3.0 **[Computers and Education]:** General

General Terms
Human Factors

Keywords
Community service, mentoring programming

1. INTRODUCTION

Despite years of interventions aimed at broadening participation in computer science, research [7, 8] continues to report barriers to participation for entire populations of our citizenry. Mired in the longstanding view that programming is an innate skill meant only for a select few [4], computer science's failure to attract a broad range and number of participants directly correlates with the wider public's equally narrow perception of the field. Yet what has received considerably less attention in this dual focus on participation and perception is the role of effective pedagogy (the third "p") to introduce CS as a tenable course of study and potential career field. Based on prior research [16, 17, 24] citing the unique potential of service-based learning as a pedagogical bridge across the rift between undergraduate and K-12 computing curricula, this paper investigates how college and also high school students can undertake both the role of teacher and learner to create a reciprocal rather than hierarchical mentoring relationship within the surrounding community.

Current discussions of service learning [14, 15] stress the pedagogical importance of reciprocity to generate more authentic and longstanding relationships, and the cascading model we present here emphasizes the interchangeability of the mentor-mentee roles, grounding undergraduates' CS-knowledge in hands-on activity while also introducing underprivileged high school students to the academic and career potential of computing. Developed through a partnership between a university's computer science and education departments, the cascading model emphasizes not only CS-based content but also the modes of delivery for such content. For the past two years we implemented the cascading model with four different cohorts of undergraduates through CS-based service learning courses, each of which was paired with accompanying workshops or a summer camp for local high school students. In our presentation of the cascading model, we report on (1) its implementation on both the undergraduate and high school levels as a means to attract a wider range of youth to CS, (2) its function as a pedagogical tool in shifting perceptions of what it means to mentor, and (3) the resultant changing perceptions of CS as a course of study and career. In the discussion we address the challenges and opportunities of developing and implementing this type of a service learning course and how the cascading model may be additionally streamlined for future iterations.

2. BACKGROUND

Despite a recent report from the Computing Research Association (2012) citing a fourth straight year of increased CS undergraduate enrollments, there continues to be significant concern over the development of a viable "pipeline" into CS when it comes to attracting and retaining a diverse range of students [2]. Problems in participation are very much problems in perception as Kelleher and Pausch [18] aptly point out in their taxonomy of introductory programming environments, citing the underlying social and cultural stigmas surrounding programming (and not technical

SIGCSE'13, March 6-9, 2013, Denver, Colorado, USA.

"know-how") as the field's most immediate issue. This conclusion points to a wider issue within the CS community. For while there have been great strides in making the field more accessible technologically, CS severely lacks such innovation in terms of classroom pedagogy, still doggedly relying on "top-down" lecture-style delivery as its default form of instruction on both the undergraduate and high school levels [10, 12].

In this paper, we focus on the role of pedagogy that is inherently connected to these interrelated issues of participation and perception. There are two elements to the pedagogical approach we present with the cascading model. First, there is the overall design of the program, a *service learning* course, in which learners' acquisition of CS concepts and skills is directly tied to the authentic social connections they develop outside of the college classroom. Formally defined by the U.S. National and Community Service Act of 1990 as academically-integrated learning through active participation in community outreach, service learning has demonstrated the potential to enhance students' academic comprehension while also fostering an understanding and caring about other individuals [20]. In terms of CS, service learning coursework has been found to particularly appeal to women and underrepresented minorities, helping dispel the entrenched view among some communities that computer science is asocial and inapplicable [23, 24]. While CS service learning programs have been implemented in some universities [9], research on service learning is not highly visible in either the computer science education community or in the service learning community [24]. One of the goals of our study is to contribute to this growing body of knowledge around CS-based service learning and to promote such outreach as a viable way to broaden participation in and perceptions of computing.

Second, there is the particular instructional design of the service-learning approach, namely *mentoring* through a reciprocal "learning-by-teaching" model. While scholars and practitioners do not agree on a single definition of mentoring, there is consensus aligned with the popular view that mentoring involves acting as a guide, advisor, and/ or counselor to a mentee [22]. Previous discussions of the learning benefits in mentoring have included improved self-esteem, increased opportunities to interact with peers, increased social competence [3] and civic participation [26]. Within the CS department, engaging undergraduates in campus-based mentoring experiences has demonstrated success in raising students' sense of self-efficacy [5] as well as modestly increasing enrollment numbers [11]. Noticeably absent however from such research (and of particular interest to us), is the notion of the mentor as learner. Little research expands the continuum of mentoring roles from learners to teachers and thus would be more inclusive of a view that sees mentoring as a reciprocal rather than a hierarchical relationship. Focusing on mentoring as a process by which one transforms from learner to teacher offers the possibility of a more equitable and reciprocal relationship that opposes the deficit perspective prevalent in many mentoring and service-learning outreach efforts [25, 26], and for this reason may be especially promising in working with underrepresented groups in CS with whom issues of participation and perception are especially prevalent.

3. METHODS

3.1 Mentors and Mentee Participants

Mentors. Our cascading mentoring model was developed through a partnership between the university's computer science and education departments via a service learning course for undergraduates. During the two years, 16 undergraduate students (75% male, 25% female; 38% African American males, 13% Asian females, 50% Caucasian) completed the EAS 285 service learning course and participated in mentoring activities to broaden participation among surrounding local high school students. More than half (56%) reported that they were considering or had declared CS or CE as a major or minor in college; other majors included bioengineering, electrical engineering, and chemical bio-molecular engineering. All undergraduates had taken at least taken one introductory CS course that was a requirement for enrolling in EAS 285.

Mentees. On the high school side, 54 students (33% females, 67% males; 37% Caucasian, 33% African American, 20% Asian, 4% Hispanic, 2% Native American/Alaskan Native, 4% other) participated in the program as mentees. All participants were 14-18 years of age and were high school students in grades 9-12.

3.2. Service Learning Course

The EAS 285 Teaching Computer Science Basics was the community service learning course offered in spring '10, summer '10, spring '11, and summer '11. It was co-taught by a CS faculty member (Griffin) and a school of education doctoral student (Burke). Initially undergraduates undertook the role of mentees, learning about educational theories and practices as well issues surrounding K-12 CS education. Of particular significance was the theory of Constructionsim [21] and grounding CS learning in the design and building of tangible objects that could be then shared with others, be it video games, digital stories, or even programmable plush dolls and clothing accessories. Undergraduates also learned several new educational technologies: the Scratch introductory programming language, and, depending on the course offering, the electronic textile kit Lilypad Arduino, Processing (with Java), or Python. They studied software engineering principles, which then informed the design and development of their educational software projects. In addition to completing reading, writing, and programming assignments, the undergraduates created a variety of lesson plans, which included "unplugged" activities, programming projects and puzzles, demos, tutorials, and Jeopardy-style quiz games. In the spring, workshops were held at two local high schools. During the summer, *Boot Up!* week-long intensive computing camps were held for high school students.

3.3 Data Sources and Analyses

Surveys. All participants, undergraduates and high school students, completed pre- and post- surveys. Depending upon access to computers, some participants completed paper surveys, while others completed surveys online. The survey questions included demographic items, interest ratings, confidence with computing, and satisfaction with participating in the camp, course, or workshop. Both mentors and mentees also completed pre- and post-workshop surveys on their experiences and computing attitudes. The undergraduates took an additional survey, using Retrospective Pre-Test (RPT) methodology [19], in which they rated their level of interest on five items by reflecting back to how they felt before their mentoring experience and then to rate how they felt now. Surveys were analyzed with 2-tailed paired samples t-tests (95% CI) to determine the significance of differences in ratings before and after computing experiences.

Post Interviews. We also conducted post interviews with undergraduates about their mentoring experience in-person or over the telephone, all within one week after the actual workshop experiences. These recorded interviews were then transcribed and analyzed in terms of the three themes identified in an earlier study

[17] that specifically describe the transition from mentee to mentor in terms of the (1) social challenges of mentoring, (2) facilitating others' understanding, and (3) reinforcing one's own learning through the mentoring process (see Figure 1).

Reflections	Sample Interview Statement
Social Issues	[B]e yourself….Don't obsess about the image of how a mentor should be. You know, be malleable and you know, work with the students. Don't, like, assume that you're coming from the university, which is a very nice institution, and you were probably one of the better students in your high school, like realize that not everyone's going to be like you. *(Undergraduate Senior, following the Spring 2011 Course)*
Facilitating	What I've done in the past 3 or 4 months would likely shape sort of what I would do as a mentor going forward. I would really try and help students focus on more of the technical aspects of what they are doing. It's really nice to be able to work with students one-on-one rather than in a big group. I do like the style of teaching where you give them a project, where they have some ultimate goal that they have to work towards on their own. They have to be self-driven and then they come to you for questions rather than you providing just a full lecture. *(Undergraduate Senior, following the Spring 2011 Course)*
Learning	I knew I'd be able to help out because a lot of the stuff we did last year, but also went there as a mentor, but I went there to learn too, so I was like, in the same shoes as the other students because I was learning some stuff too. So we could learn together and I can help them know what I know. *(High School Junior, following the Summer 2011 CS Camp)*

Figure 1: Themes & Sample Interview Responses

In addition, we interviewed five high school students who participated as mentees in workshops at one high school as 9th graders and then became mentors to a new crop of students as 10th graders. They served with us for an entire year. At the end of the year, we conducted hour-long interviews with these students to capture their reflections on their experiences on how they saw their roles as a mentors socially develop, what they learned about mentoring, and what they learned about computing through their mentoring experiences.

4. RESULTS

4.1 Service Learning Course and Mentoring Experiences

Undergraduate Mentors. In the pre-survey, mentors listed a number of areas/ skills that they hoped to learn through the experience. Roughly half of the respondents listed creative design skills, a third listed a greater understanding of how technology works, and 39% expressed the hope to learn whether computer science would be a good fit as a career. In terms of the mentoring experience, the two primary reasons undergraduates chose to become a mentor were to develop leadership skills (89%) and to develop their own computing skills (79%). When the undergraduates were asked to rate their experience in the service-learning course, all agreed that (1) their learning of CS principles was reinforced through teaching; (2) their overall awareness of the field of computer science education increased; and (3) their capacity to serve as role models to aspiring young technologists was very much heightened by the high school students' willingness to turn to them for support. They also reported improvements in their abilities to problem solve, deliver effective presentations, teach hands-on skills, make positive changes in the community through leadership activities, and set clear boundaries with participants. Eight of the eleven skills improved significantly ($p<.05$) as the result of their mentoring experience. Through this service learning experience, student mentors described their experiences as not just helping them develop mentoring skills but also "life skills" such as being able to think on one's feet, adapting as needed, setting clear and realistic boundaries, and using small-talk as a way to get to know others and develop a relationship. These ratings very much support the feasibility of having undergraduate students serve as mentors.

High School Mentees. We also asked high school students mentees what they hoped to learn from the camp or workshop experience. High school students listed creative design skills (71%), how technology works (60%), and if computer science would be a good fit as a career (50%). The camps or workshops overall were rated as "effective" or "very effective" by 91% of high school mentees and 83% of them reported that their level of participation in the camp or workshop was "active" or "very active." While 51% of high school mentees were considering computer science or computer engineering as a major or minor in college before the camp or workshop, this number increased to 65% after the mentoring experiences. The mentees agreed or strongly agreed with the following statements on the post-surveys: "I know more about computing now" (81%); "I increased my knowledge of computing" (89%); "I have a better understanding of concepts like computational thinking" (82%); "I have more confidence about computing" (85%). According to high school students' ratings, both the mentors (41%) and the program (37%) had an impact on their decisions about college major. High school mentors considered their most important role models to be the undergraduate mentors (54%) over the adult teachers (24%).

4.2 Computer Science Career Interests and Computing Attitudes

Undergraduate Mentors. After participation as a mentor, undergraduates' career interests increased across all dimensions: computer science in general, teaching computer science, mentoring younger students, majoring or minoring in computer science in college, and working in computer science. Mean interest rating scores ranged from 3.1 to 4.3 on a 5-point scale, from "extremely low" (1) to "extremely high" (5). Interest in teaching computer science was moderate at the start of the course, but increased significantly ($p<.01$) by the end of the course. Not surprisingly, interest in mentoring younger students was high at the start of the course, but increased significantly ($p<.05$) by the end of the course. While a greater number of undergraduate mentors considered declaring a computer science or computer engineering major or minor in college (from 62% pre-mentoring to 77% post-mentoring), this increase was not significant. In contrast to career aspirations, students' computing attitudes such as "computer jobs are boring" or "women can have jobs in computing" did not change significantly in the positive direction after the mentoring experience. Mentors rated their level of agreement for 15 items on a 4-point scale from "strongly disagree" to "strongly agree." We predicted that computing attitudes would become more positive with more computing experience, yet 8 of 15 attitudes changed in a negative direction, but not significantly so.

High School Mentees. We found that participation in the summer camp or school workshop significantly impacted ($p<.05$) high school students' consideration of computer science or computer engineering as a major or minor in college. In the pre-survey just over half of the high school mentees (51%) were interested as compared to 65% in the post-workshop survey. Much of this appears to be the influence of the undergraduate mentors, all of whom were rated as either “effective” or “very effective” by all mentees who completed the post-workshop survey. Mentees' attitudes towards computing were investigated with 15 items on a 4-point scale from “strongly disagree” to “strongly agree.” Like the undergraduate mentors, high school mentee's ratings went down, rather than up on items such as: “computing is fun; women can have jobs in computing; men can have jobs in computing; I like computing; I can become good at computing; I think computing is useful; I want to find out more about computing; and I am interested in a career in computing.” Their attitudes did not shift significantly with the exception of one item: “I am good at computing.” In contrast to the undergraduates, high school mentees all rated themselves higher after being mentored but again, none of these shifts were significant.

4.3 Reflections on Cascading Mentoring

Undergraduate Mentors. Our analyses of the 16 interview transcripts revealed that all undergraduate and high school mentors reflected upon this transition from mentee to mentor in terms of all three themes with reflections about facilitating (n=27) as the most frequent in occurrence, followed closely by social observations (n=25) and then by direct references to their own technical and pedagogical learning throughout the process (n=19). This is not altogether surprising as all participants spent the majority of the time reflecting upon the mentorship role and the high school students they encountered over their respective workshops and camps. Interestingly though, the undergraduate participants were much more likely to reflect on the social conditions of the students whom they were mentoring—particularly in terms of comparing their mentees' classroom environments to their own previous high school experiences.

High School Mentors. The five returning high school students who spent an entire year mentoring others spoke reflectively about what they had learned both about mentoring and about computing through their experiences. Initially, most thought that mentoring would be easy because they would simply need to tell students what to do and that they would be in positions of authority. Later however, they realized that mentoring was a complex mixture of figuring out students' interests, helping them become interested in computing, and helping them discover the answers to their own problems rather than telling them what the answers were. For instance, Corbin (a pseudonym, as are all names herein) explained that many students who needed help did not recognize that they needed help, while others thought they needed more help than they actually did. Jackson came to the realization that providing expertise with a joke or a “nudge” worked better with mentees than “telling them.” Or as Jackson explained, “Instead of trying to point them where I think they should go… I tried to allow them to learn in the same way that I did, which was sort of use whatever appealed to you for whatever reason and just feel your way around. And if you had any questions about what anything did, to the best of my knowledge, I would help you understand.”

These high school mentors also described how they had learned things about programming (specifically with Scratch, the software they used) directly from their mentoring experiences. As William put it, some kids would ask him questions about things that they were interested in doing, “so I had to learn it myself to be able to show them how to do it.” Or as Chase described, “[S]ometimes people would come to me with more complex things that I've never actually done before and I would figure it out with them.” William provided an example of this: “Jenny was one of the people who um, I had to teach about variables… [S]he needed a solution to a problem and I had dabbled a little bit in variables before, so I went onto that part of Scratch and started experimenting with it, and then I got the hang of it and we were able to solve the problem.” Finally, in an example of the possibilities of a cascading mentoring model, Corbin described his experiences with undergraduate mentors in the same workshop where he too was a mentor of younger high school students than himself: “I felt like I was a mentor being mentored,” he said. The high school mentors expressed gratitude for what they learned from the undergraduate mentors, most of whom knew more about programming or mathematics than they did and who were able to help the high school mentors follow their own interests in programming, as they in turn helped the more novice high school mentees design and implement their programs. This learning from others while teaching at the same time highlights the benefits of the cascading model of mentorship.

5. DISCUSSION

We examined the opportunities and challenges in integrating cascading mentoring into a CS service learning to address the role of pedagogy—the third “p”—that is inherently connected to issues in broadening participation in and perceptions of computing. Our findings suggest that implementing cascading mentoring is a promising but also complex pedagogical effort in addressing the multiple challenges associated with broadening participation and perceptions of computing.

In terms of opportunities, we found that the service learning course was successful, in particular in recruiting a high percentage of underrepresented groups of students, both on the mentor and mentee side. The cascading mentoring model also proved to be successful in broadening participants' interest in computer science and developing a more sophisticated perspective of what it means to mentor, teach and learn. It provided rich opportunities for participants to engage in reciprocal mentoring, a process that created a more equitable relationship between mentors and mentees. While many efforts in broadening participation focus on raising interest in computing and increasing participants' ideas of computing, it is equally important to promote different pedagogies that address learners' needs, interests, and backgrounds. With the increasing need for technology fluency, there is a primacy placed on promoting content. Indeed, there is a great need in building technical skills and conceptual knowledge of computing, particularly among underrepresented groups who often do not have access to courses and activities involving computation. By the same token, by just focusing on content, we leave out instrumental insights from the pedagogical side developed over the last 30 years that content coverage alone is not sufficient, but rather learning needs to start with students' informal knowledge and take advantage of scaffolded instruction [6]. As evident in both the mentor and mentee pre- and post-workshop surveys and mentor interviews, while computing was a significant reason participants chose to be part of the service learning outreach, it certainly was not the only reason for participating nor the only skill set the undergraduates and high school students developed. Participants cited leadership ability, time management skills, and creative design as both the reason for and takeaways from the experience, and it is clear that they very much valued the practical (e.g., “life skills”) reasons for service learning coursework as well. The cascading mentoring model illustrated that by “turning the tables”—namely, by having mentees become mentors—we provided undergraduates and returning high school participants with powerful insights about their own learning through their own nascent teaching efforts. Namely, the design of the service learning course turned them into reflective practitioners or learners who became more finely attuned to the needs of others and their own expertise. After all, helping someone to learn in an efficient and respectful manner is of value beyond schooling and is especially crucial to succeeding in the workplace, where much learning about and with technology does not, in fact, happen through top-down content delivery but in far less formal interactions with others that directly relate to knowing how to effectively offer (as well as graciously receive) assistance.

The feasibility of having undergraduates and high school students shift from mentee to mentor and become impactful mentors was demonstrated for both groups of students. The impact of mentors on students at all levels was significant in terms of decision for considering a college major. Mentoring abilities improved significantly, as did confidence, interest in teaching computer science, and interest in mentoring. Minority and underrepresented groups participated in numbers far exceeding their normal representation in computing. The effects of learning by teaching were also significant, increasing student interest in both teaching computer science and mentoring younger students and reinforcing student learning through teaching. Broadly speaking, our findings support the assertion that this sense of having been in the "same shoes" is an imperative in terms of developing future mentors out of the mentee role [14, 15]. The mentor reflections of returning high school students suggests that those who engaged in the actual workshops (rather than just the undergraduate coursework) may very well have a claim to better understand how mentees feel and engage during such outreach. Of course, while some of this may simply be attributed to a similarity in age (high school students reflecting upon other high school students), future iterations of the cascading model need to be mindful that its currents need not be simply "down-pours" but potentially "up-pours" as well. That is, contrasting the mentoring styles and experiences of returning high school participants with those of the college undergraduates offers an ascending insight within a cascading model.

There was, however, one conflicting finding that deserves further discussion. While we were able to confirm some of the expected positive changes in college and career aspirations, we also recorded more stagnant views on computing attitudes among both undergraduate mentors and high school mentees. It may simply be that attitudes change more slowly over time, and in the case of high school students, they had much less understanding of computing before they became mentors so their pre-camp workshop ratings might have been inflated. An alternative explanation is that these post-workshop survey ratings reflected improved understanding of computing, rather than negative attitudes. After all, as the aphorism reads, we tend to only begin to understand a topic when we realize how little we ourselves know. It is difficult to accurately rate one's feelings on a pre-workshop survey when one may not yet fully understand what he or she is actually rating. This suggests that it might be more appropriate to assess attitudes with a retrospective pre-test methodology so that respondents have had the exposure, training, or learning required to accurately rate their pre and post experiences.

We faced several challenges, mostly concerning institutional and administrative issues. One of the biggest challenges was in recruiting undergraduates to the EAS285 class. Although a relatively high percentage of African American males enrolled, the overall enrollment numbers were low, averaging only four undergraduates in four course offerings. The course was an engineering elective course but we were unable to get it rostered as part of the core CS curriculum. This administrative integration would have been critical for many undergraduates who expressed interest in the course but couldn't fit the course in their otherwise booked schedule with required courses and little room for electives. The remarkably busy schedules characteristic of undergraduate curricular life proved to be another challenge—and integrating these schedules into the far more regimented schedules of middle and high school students only compounded this challenge. The undergraduate mentors typically needed two college course periods in order to have enough time to travel and visit their high school mentees.

Perhaps one of the biggest challenges in bringing the cascading model to fruition is the issue of moving from seeding to scaling up the model. We largely underestimated the time it would take to implement and maintain a mentoring cascade over several iterations. While the service learning course was offered both years as a traditional spring semester-long course as well as a more compact summer-based course, the service-learning component of each course varied considerably. Whereas undergraduates in the spring course enacted the cascading model through on-site mentoring at a public middle school and two high schools, undergraduates from the summer course served as mentors at an on-site intensive week-long summer *Boot-Up!* camp. This unique mix of off-site and on-site mentoring certainly offered us a glimpse into the range of service-learning possibilities, but there needed to be more attention directed to weighing and adjusting mentoring styles based on whether undergraduates were on site at public schools or whether the public school students were on the university campus. Going forward we would like to develop curricula that specifically address the tacit yet remarkably significant role of "place" when it comes to mentoring students, especially those in underserved communities. To a degree, the range of locales also affects the generalizability of our results as the context in which the cascading mentoring can occur may be a tacit variable affecting the overall shifts in attitudes toward computing and mentoring.

Moving ahead into the third and last year of implementing the cascading mentoring model, we made adjustments to address some of the challenges. Due to persistently low enrollment, we discontinued EAS 285 and introduced a new service learning course in spring 2012, called CIS 101 *Compute to Create*, an official pilot of the CS Principles project [1]. As an introductory CS course for non-engineering students with no prerequisite, the new course satisfies a formal reasoning requirement (as does introductory calculus). As we had hoped, enrollment in CIS 101 was high; it filled to capacity (15 undergraduate students) with a minimum of advertising. The service component of the new course involved some mentoring of high school students but significantly less so (only three hours per semester) and did not include the depth of CS engagement that high school students experienced in the summer camp and school workshops. Instead, mentoring activities focused on short weekend and school day workshops as well as public computing demonstrations at the outdoor Philadelphia Science Festival. While these reductions in time and engagement allowed us to reach a much larger number of K-12 students than previously, we need to examine in more detail the differences in impacting and broadening participation and perceptions in computing.

There is an urgent need to think about how to create communities of learners, whether it is college classroom, or as in our case, across schools. This of course is no easy feat, particularly because the social and academic divide between high school and college is already so significant for students from underserved communities. However, such a network of communities of learning—formal and informal and across grade levels—represents an essential first step to begin to systematically address these longstanding issues of participation in and perception of CS, as well as a means to both distill and distribute a pedagogy of best practices.

6. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation (CIS-0940511). Any opinions, findings, and conclusions or recommendations expressed in this chapter are those of the authors and do not necessarily reflect the views of the National Science Foundation or the University of Pennsylvania.

7. REFERENCES

[1] Astrachan, O., Briggs, A. 2012. The CS Principles Project. *ACM Inroads*, 3, 2, 38-42.

[2] Barker, L., McDowell, C., and Kalahar, K. 2009. Exploring factors that influence computer science introductory course students to persist in the major. *ACM SIGCSE Bulletin,* 41, 2, 282-286.

[3] Barton-Arwood, S., Jolivette, K. and Massey, N. G. 2000. Mentoring with elementary-age students. *Intervention in School and Clinic*, 36, 1, 36-44.

[4] Biggers, M. Brauer, A. and Yilmaz, T. 2008. Student perceptions of computer science: A retention study comparing graduating seniors with CS leavers. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education* (Portland, OR USA March 12-15). ACM, New York, NY, 402-406.

[5] Boyer, K., E. Thomas, A. Rorrer, D. Cooper, and Vouk, M. 2010. Increasing technical excellence, leadership and commitment of computing students through identity-based mentoring. Paper presented at the *Proceedings of the 41st SIGCSE Technical Symposium on Computer Science Education* (Milwaukee, Wisconsin, USA March 10-13). ACM, New York, NY, 167-171.

[6] Bransford, J., Brown, A., and Cocking., R. 2000. *How People Learn*. Washington, DC: National Academy Press.

[7] Bruckman, A., Biggers, M., Ericson, B., McKlin, T., Dimond, J., DiSalvo, B., Hewner, M., Ni, L. and Yardi. S. 2008. “Georgia Computes!”: Improving the computing education pipeline. In *Proceedings of the 41st SIGCSE Technical Symposium on Computer Science Education* (Chattanooga, Tennessee, USA March 03-07). ACM, New York, NY, 86-90.

[8] Carter, L. 2006. Why students with an apparent aptitude for computer science don't choose to major in computer science. *ACM SIGCSE Bulletin*, 38, 1, 27-31.

[9] Dodds, Z., and Karp, L. 2006. The evolution of a computational outreach program to secondary school students. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA March 01-05). ACM, New York, NY, 448-452.

[10] Donaldson, S. 2011. Teaching computer science in a big questions framework. *Journal of Computing Science in College*, 26, 5, 72-79.

[11] Egan, M. and Johnson, M. 2010. Service learning in introductory computer science. Paper presented at the Proceedings of the 15th Annual Conference on Innovation and Technology in Computer Science Education (Bikent, Ankara, Turkey June 26-30).

[12] Goode, J., Chamna, G., and Margolis, J. 2012. Beyond curriculum: The exploring computer science program. *ACM Inroads*, 3, 2, 47-53.

[13] Guzdial, M. 2004. Programming environments for novices. In S. Fincher and M. Petre (Eds.), *Computer Science Education Research* (pp. 127-154). Philadelphia, PA: Taylor & Francis.

[14] Hart, S. 2006. Breaking literacy boundaries through critical service-learning: education for the silenced and marginalized, *Mentoring and Tutoring,* 14, 1, 17-32.

[15] Howard, R. W. 2006. Bending towards justice: service learning and social capital as means to the tipping point, *Mentoring and Tutoring,* 14, 1, 5-15.

[16] Hug, S., Thiry, H., and Tedford, P. 2011. Learning to love computer science: Peer leaders gain teaching skill, communicative ability and content knowledge in the CS classroom. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (Dallas, TX USA March 09-12). ACM, New York, NY, 201-206.

[17] Kafai, Y. B., Desai, S., Peppler, K., Chiu, G. and Moya, J. 2008. Mentoring partnerships in a community technology center: A Constructionist approach for fostering equitable service learning. *Mentoring & Tutoring*, 16, 2, 194-201.

[18] Kelleher, C., and Pausch, R. 2005. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37, 2, 83-137.

[19] Lamb, T.A. 2005. An imperfect but useful tool: The retrospective pretest. *The Evaluation Exchange*,*11*(2). Retrieved July 5th, 2012 at http://www.gse.harvard.edu/hfrp/eval/issue30/ ttp://www.gse.harvard.edu/hfrp/eval/issue30/spotlight.html.

[20] McIver, W., and Rachell, T. 2002. Social informatics and service learning as teaching models. *IEEE Technology and Society Magazine*, 21, 3, 24-31.

[21] Papert, S. 1980. *Mindstorms: Children, Computers, and Powerful Ideas*. New York, NY: Basic Books.

[22] Roberts, A. 2000. Mentoring revisited: A phenomenological reading of the literature. *Mentoring and Tutoring*, 8, 2, 145-170.

[23] Rosmaita, B. 2007. Making service learning accessible to computer scientists. In *Proceedings of the 38th Technical Symposium on Computer Science Education* (Covington, KY, USA March 07-10). ACM, New York, NY, 541-545.

[24] Sanderson, P. 2003. Where’s (the) computer science in service-learning? *Junior Computer Science College*, 19, 1 (Oct. 2003), 83-89.

[25] Villalpando, O., and Solorzano, D. G. 2005. The role of culture in college preparation programs: a review of the research literature. In W. Tierney, Z. Corwin and J. Colyar (Eds.), *Preparing for College: Nine Elements of Effective Outreach*. Albany, NY: State University of New York Press.

[26] Walker, G. 1996. Social change one on one: the new mentoring movement, *American Prospect,* 7, 27, 75-81.