Increasing Minority Youths' Participation in Computing through Near-Peer Mentorship*

Michael J. Lee
Department of Informatics
New Jersey Institute of Technology
Newark, NJ 07102
mjlee@njit.edu

Abstract

It is critical to focus on diversity and increasing participation of underrepresented groups in computing. To address this need, we must better understand minorities' access to role models and mentors, especially at a young age, as research and practice shows that these relationships can affect students' self-efficacy and motivation in the educational fields and careers they choose to pursue. We provided a 9-Saturday programming camp to middle school students in Newark, New Jersey with near-peer mentors (first year, college student instructors) to learn more about the younger students' initial access to role-models and mentors, and how an intervention might change this. Our camp served a total of 28 minority students (17 males and 11 females; grades 5-7) from a low-income, urban area. We found that when asked at the beginning of the camp, our middle students largely reported that they did not have any role-models or mentors in computing. However, at the conclusion of the camp, these same students indicated that they developed strong connections with their near-peer mentors and even saw them as role-models. These findings highlight the need for more mentorship opportunities for students of all ages, and the importance of providing resources and support to help develop and nurture these connections.

1 Introduction

Computing jobs continue to be in significant demand across the world [23], with programming jobs being among the fastest growing career areas [4]. Unfortunately however, many youth—especially those from underrepresented minority groups—are not pursuing related educational opportunities or careers [23]. Without adequate representation from all groups, we run the risk of severe inequities and bias in software design and use [24]. Therefore, we must investigate new learning opportunities that both engage a wider range of the population and sustain their learning and engagement over time so that we can better address the lack of diversity and low supply of technology workers.

This project attempts to address inequities in computing by learning more about low-income, minority, middle school students' access to mentors and role-models in computing, and how a 9-Saturday programming camp with university student instructors might affect this. We surveyed middle school students, before and after they were exposed to programming with a debugging game, website design curriculum, and block programming environment. We specifically chose to focus on middle school students, as research has shown that this is the age when youth begin to form strong, lasting ideas about their future careers [6, 8, 10]. Our goal was to provide underrepresented minority students with a positive experience with programming along with similarly-aged instructors, to develop productive mentee-mentor relationships, and produce evidence that these types of activities yield benefits beyond learning how to code.

2 Related Work

2.1 Middle Schools

Although many computing related enrichment activities target high school students, research has shown that youth begin to form ideas about future careers earlier, during their formative years in middle school [6, 8, 10]. Many educational research efforts targeting younger children have focused on how to engage them in Computer Science (CS) by using programming environments that are user-friendly and have the potential to lower the cognitive threshold for novice programmers, such as Scratch (e.g., [21]), Alice (e.g., [13, 14]), Gidget [17, 18, 20], and others [9]. Overall, studies repeatedly found that these types of environments are effective in the acquisition of basic programming skills and concepts (e.g., [16, 20, 21]) and support computational thinking. For example, Meerbaum-Salant et al. [22] found that Scratch helped middle schoolers learn most of the targeted CS concepts. Lee & Ko [20] found similar results with Gidget, where users showed significant, measurable learning gains for targeted CS concepts. In our study, we aim to learn more specifically about underrepresented minority middle school students and how a coding camp using many of these tools with similarly-aged instructors might affect their views about access to computing mentors and role-models.

2.2 Near-Peer Mentorship

Research shows that self-efficacy about an activity can be positively influenced by seeing someone similar to oneself being successful in similar activities [2]. On the other hand, seeing someone who is dissimilar or who has disparities in experience can cause one to believe that those skills are beyond their reach and are therefore disinclined to invest the effort needed to pursue them fully [1]. Therefore, youth are more likely to imitate those who they perceive as similar to themselves [2]. Denner [6] and D'Souza et al. [7] found that role-models are an important motivator for middle school girls' and high school students' interest in computing, respectively. Clarke-Midura et al. [5] found that middle school students relate particularly well to near-peer mentors—mentors that were only a few years older than the students and not necessarily the most skilled with CS—increasing the younger students' self-efficacy and interest in computer science. Unfortunately, many female and minority students may not have role-models or mentors in computing that they can relate to directly. Based on these observations, we aim to learn more about minority youths' access to role-models and/or mentors, and how this might affect their pursuit of computing-related education and careers.

3 Method

3.1 Participants and Instructors

We ran a 9-Saturday camp at a local K-8 public school in Newark, New Jersey. The school principal recruited students by recommendations from the 6th and 7th grade teachers at her school, who made announcements about the camps in their classes. We did not specifically recruit for underrepresented minorities, as the school's demographic makeup reflects the community it serves, composed of 52% Hispanics, 47% African Americans, and 1% others; 82% of these students are low-income—eligible for free or reduced-price lunch.

For the camp, we had access to one classroom, the cafeteria, equipment (e.g., laptops with WiFi access), middle school teachers (one male and two female technology and mathematics teachers), and first-year undergraduate student instructors. The instructors ran all the events and taught the middle school students, while the teachers were available primarily for classroom support (e.g., to help keep the children on task) and did not provide any computing-related instruction. The day was split into two parts, a morning session and an afternoon session. The camp included 11 instructors (4 females and 1 male in the morning session; 1 female and 5 males in the afternoon session). The instructors all took a university shuttle to and from the middle school, and all of the middle school students were dropped off and picked up by their respective parents/guardians.

3.2 Procedure

Each of our nine Saturdays consisted of a seven-hour day including breakfast, a morning programming activity, lunch, and an afternoon (block) programming activity. Breakfast was 15 minutes and lunch was 45 minutes. On the first day before starting any programming activities, students filled out a pre-test questionnaire. On the (ninth) final day, students filled out a post-test questionnaire before concluding the camp. Each of the questionnaires took 20 minutes to complete. The game/HTML (morning) and block programming (afternoon) activities lasted approximately three hours each, every Saturday.

3.2.1 Morning Session

Since one's first experience with code is important [15], we chose two different activities for the morning sessions. On the first day, students played *Gidget* [20] (www.helpgidget.org)—a free, online, educational programming game—which has been shown to be engaging for a wide range of programming novices and a good introduction to (text-based) programming for after-school programs [11, 19]. We encouraged our middle school students to complete as many levels as they could during the first day, and allowed them to continue playing the game from home and during any free time on subsequent camp days. For all other days, we had students complete code.org's *CS Discoveries*' curriculum, which was designed for and used by thousands of middle school students. This curriculum focuses on teaching students about HTML, CSS, best practices for posting on the internet, and ultimately for them to create a personal website.

3.2.2 Afternoon Session

We used Harvard University's *Creative Computing for Scratch* curriculum [3] in our afternoon sessions to give the middle school students experience with a block programming language. Scratch has been used extensively to engage youth with block programming through animation authoring and storytelling [21, 22]. This curriculum focused on teaching students introductory programming concepts and to ultimately create a personal music video.

3.2.3 Guest Speakers

Every even week (i.e., Weeks 2, 4, 6, and 8), a guest speaker came to talk about their computing-related job for 30 minutes after lunch. All speakers were recruited to have grown up, live(d), and work in the local metropolitan area. We had a total of four speakers, all African American/Black professionals (three males and one female) who worked in game/app development, computer security, disaster response, or research, respectively. The guests spoke about their background, education, and career, giving examples of their responsibilities (and sometimes demoing some of their tools or products). After the guest finished speaking, students asked them questions for 10-15 minutes.

3.2.4 Pre- and Post- Test Questionnaires

The middle school students completed a pre-test questionnaire before the start of the camp, and a post-test questionnaire as the final activity on the last day of the camp. The questionnaires consisted of three questions asking students if they wanted to go to college, and whether they would attend more classes *at* school or *outside* of school to learn more about programming. Next, we asked them the following questions about mentorship and role-models:

1. *Do you know of anyone famous related to computing? If so, who?*
2. *Do you have a role-model or role-models related to computing? If so, who?*
3. *Do you know anyone personally who has a computing related job? If so, who?*
4. *Have you ever had a mentor for anything related to computing? If so, who?*

Then, only for the pre-test (since we did not expect responses to change for the duration of the camp), we collected demographic information (i.e., grade level, ethnicity, gender, and eligibility for free/reduced lunch), and asked four yes/no questions examining if they: owned a smartphone, had prior programming experience, and had access to a computer and/or internet at home (since the availability of computing resources has been reported to be significantly related to household income and inversely affected by minority status [12]).

4 Results

It took the middle school students approximately 20 minutes to complete each of the pre- and post- test questionnaires. In total, we had 28 students ($n = 28$) participate in our event (10-13 years old). The camp included 17 boys and 11 girls (one 5th grader, twelve 6th graders and fifteen 7th graders; see Figure 1). All of the students were underrepresented minorities in STEM, identifying as either African American/Black (13 students) or Hispanic/Latino (15 students), and all were eligible for free/reduced cost lunch.

Figure 1: Participants and near-peer mentor instructors interacting in our camp.

All of our 11 instructors were first year college students (5 females and 6 males; aged 18-19 years old) from a local university, majoring in various fields including computer science, biology, informatics, and history. They also identified with a wide range of ethnicities, including South Asian (3 students), White/Caucasian (2 students), Pacific-Islander (2 students), Hispanic/Latino (3 students), and Middle-Eastern (1 student). Without any prompting, each group of instructors selected a lead instructor for the morning session and afternoon session (both were female students), respectively.

When applicable, we use nonparametric Chi-Squared likelihood ratio tests with $\alpha = 0.01$ confidence throughout our analyses—as our data was nominal and not normally distributed—to compare participants' responses. We report our statistically significant results with the understanding that our sample size is small and that the resulting findings may not be widely generalizable. For all questions, we ran comparisons between demographics (grade level, ethnicity, or gender) and questionnaire (pre-test vs. post-test). Unless indicated otherwise, there were no significant differences between responses by demographics within or between the pre-test or post-test for the results described below.

4.1 Questionnaire Results

All students indicated that they wanted to attend college in both the pre- and post- test questionnaires. Similarly, all students indicated that they would be willing to take more computing-related courses *at* school if available. As there was no difference or change in responses for either of these questions, there were no significant differences between responses by demographics (grade level, ethnicity, or gender) within or between the pre-test or post-test questions.

Two students (both African American/Black, female students) indicated in their pre-test that they would *not* want to take computing-related courses *outside* of school. However, their responses changed for the post-test, indicating they *would* take computing-related courses *outside* of school.

4.1.1 Mentorship & Role-Model Results

For our question, *Do you know of anyone famous related to computing? If so, who?,* in the pre-test, 21 students (75%) said 'no' or left it blank. The remaining 7 students (25%) put either *Steve Jobs* (4 students), *Bill Gates* (2 students), or both (1 student)—both White/Caucasian males. This did not change significantly for the post-test, with 11 students (39.3%) writing *Elon Musk* (2 students), *Steve Jobs* (3 students), *Bill Gates* (2 students), or any combination of these names (4 students)—all famous White/Caucasian males.

Next, for our question, *Do you have a role-model or role-models related to computing? If so, who?,* in the pre-test, 26 of 28 students (92.9%) said 'no' or left it blank. For the two students (one Hispanic/Latino male and one African American/Black male) who indicated that they had a role-model, they mentioned a family member (a cousin or an older brother who was a game developer). There was a statistically significant change in response between the pre- and post- test, with 23 students (82.14%) indicating in the latter test that they had a role-model ($\chi^2(1, N = 56) = 44.083, p < .01$). Students indicated a guest speaker (11 students), a near-peer instructor (9 students), or any combination of these categories (3 students; e.g., one guest speaker and one near-peer mentor, or two near-peer mentors), as their role-model(s).

The results for our next question was similar, *Do you know anyone personally who has a computing related job? If so, who?,* in the pre-test, 26 of 28 students (92.9%) said 'no' or left it blank. The same two students from the previous question mentioned a family member with a computing related job. There was a statistically significant change in response between the pre- and post- test, with all 28 students in the latter test indicating that they knew someone with a computing related job ($\chi^2(1, N = 56) = 62.937, p < .01$). Here, students indicated a family member (2 students), a guest speaker (25 students), or a near-peer instructor (1 student), as someone they knew personally with a computing related job.

For the final question, *Have you ever had a mentor for anything related to computing? If so, who?*, in the pre-test, 24 of 28 students (85.7%) said 'no' or left it blank. The same two students from the previous question mentioned a family member with a computing related job as someone who mentored them, and two additional students (two Hispanic/Latina females) mentioned one of their middle school teachers as a mentor. There was a statistically significant change in response between the pre- and post- test, with *all* students in the latter test indicating that they had a computing mentor ($\chi^2(1, N = 56) = 53.519, p < .01$). Here, students indicated a teacher (1 student), a guest speaker

(1 student), a near-peer instructor (23 students), or any combination of these categories (3 students), as someone that mentored them in computing.

4.1.2 Pre-Test Only Results

For our pre-test only questions, 21 of 28 (75%) students reported that they owned a smartphone. Additionally, 22 of 28 (78.6%) students indicated that they had a computer at home. These same students also had access to the Internet at home. Those 6 of 28 (21.4%) who stated not having a computer at home also did not have Internet access at home. 7 of the 28 (25%) students reported they had some previous programming experience (all using Scratch).

5 Discussion & Conclusion

In this study, we found that nearly all of our minority middle school students initially reported that they did not have any role-models or mentors in computing prior to participating in our coding camp. Moreover, prior to participating in the camp, the majority of these students (26 of 28) did not know anyone who had a computing related job, and most (21 of 28 students) could not readily identify any famous people related to computing (and for those that did, these celebrities were all White/Caucasian males, which may not be helpful for minority students seeing themselves in these leadership positions [2]). These are serious issues, as youth are at a major disadvantage in pursuing education/careers in computing not knowing anyone with computing related jobs, and not having role-models or mentors that they can look up to, relate to, or talk to.

However, we also found that a 9-Saturday coding camp—emphasizing instruction by near-peer mentors and including guest speakers from the local metropolitan area who the students could better relate and identify with [2]—significantly increased *all students* reporting of having role-models and mentors by the conclusion of the camp. Helping youth, especially minority youth, identify these types of individuals early in their academic experience is essential in keeping them engaged with topics such as computer science [6, 8, 10] and may ultimately contribute to increasing diversity and representation in computing careers [24]. These findings have important implications, demonstrating that relatively short interventions such as after-school programs can provide lasting positive outcomes on underrepresented minority youth.

Some limitations of this work include the representativeness and generalizability of our sample, our instructor demographics, and the quantitative nature of our data. We believe that our sample is representative of the community we aim to serve, as our students were all underrepresented minorities in computing. Moreover, we are confident that our results can generalize to other similar contexts focusing on increasing engagement and providing mentorship for minority youth since we had such strong effects. Next, our instructors represented a wide range of ethnicities and academic interests, and were close in age with the middle school students. We found that our younger students overwhelmingly reported that they viewed these college students as computing mentors, suggesting a good fit between mentors and mentees. Finally, we only collected quantitative survey data from the participants, and only from the perspective of the younger students. Collecting qualitative data, through interviews or focus groups, could give us a richer picture of both the middle school and college students' views and feelings about the nature of their mentor-mentee relationships, and what it means to them. We plan to collect this type of data, from both middle school and college students' perspectives, in our future camps.

6 Acknowledgements

We thank the Newark Public Schools and the Urban League of Essex County for their contributions and participation. This work was supported in part by the National Science Foundation (NSF) under grants DRL-1837489 and IIS-1657160. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the views of the NSF or other parties.

References

[1] Albert Bandura. Self-efficacy: toward a unifying theory of behavioral change. *Psychological review*, 84(2):191, 1977.

[2] Albert Bandura and Richard H Walters. *Social learning theory*, volume 1. Prentice-hall Englewood Cliffs, NJ, 1977.

[3] Karen Brennan, Michelle Chung, and Jeff Hawson. Scratch curriculum guide: A design-based introduction to computational thinking with scratch, 2011.

[4] Anthony P Carnevale, Nicole Smith, and Michelle Melton. Stem: Science technology engineering mathematics. *Georgetown University Center on Education and the Workforce*, 2011.

[5] Jody Clarke-Midura, Frederick Poole, Katarina Pantic, Megan Hamilton, Chongning Sun, and Vicki Allan. How near peer mentoring affects middle school mentees. In *ACM SIGCSE*, pages 664–669, 2018.

[6] Jill Denner. What predicts middle school girls' interest in computing? *International Journal of Gender, Science and Technology*, 3(1), 2011.

[7] Daryl D'Souza, Margaret Hamilton, James Harland, Peter Muir, Charles Thevathayan, and Cecily Walker. Transforming learning of programming: a mentoring project. In *Australasian Computing Education-Vol.78*, pages 75–84, 2008.

[8] Jacquelynne S Eccles and Rena D Harold. Parent-school involvement during the early adolescent years. *Teachers College Record*, 1993.

[9] Shuchi Grover, Roy Pea, and Stephen Cooper. Factors influencing computer science learning in middle school. In *ACM SIGCSE*, pages 552–557, 2016.

[10] Nancy E Hill and Ming-Te Wang. From middle school to college: Developing aspirations, promoting engagement, and indirect pathways from parenting to post high school enrollment. *Developmental Psychology*, 51(2):224, 2015.

[11] William Jernigan, Amber Horvath, Michael Lee, Margaret Burnett, et al. General principles for a generalized idea garden. *Journal of Visual Languages & Computing*, 39:51–65, 2017.

[12] Yasmin Kafai, Kylie Peppler, and Grace Chiu. High tech programmers in low-income communities: Creating a computer culture in a community technology center. In *Communities and Technologies*, pages 545–563. Springer, 2007.

[13] Caitlin Kelleher and Randy Pausch. Using storytelling to motivate programming. *Communications of the ACM*, 50(7):58–64, 2007.

[14] Jordana Kerr, Mary Chou, Reilly Ellis, and Caitlin Kelleher. Setting the scene: Scaffolding stories to benefit middle school students learning to program. In *IEEE VL/HCC*, pages 95–98, 2013.

[15] Andrew J Ko. Attitudes and self-efficacy in young adults' computing autobiographies. In *IEEE VL/HCC*, pages 67–74, 2009.

[16] D Midian Kurland and Roy D Pea. Children's mental models of recursive logo programs. *Journal of Educational Computing Research*, 1(2):235–243, 1985.

[17] Michael J Lee. How can a social debugging game effectively teach computer programming concepts? In *ACM ICER*, pages 181–182, 2013.

[18] Michael J Lee. *Teaching and engaging with debugging puzzles*. PhD thesis, 2015.

[19] Michael J Lee. Exploring differences in minority students' attitudes towards computing after a one-day coding workshop. In *ACM ITiCSE*, pages 409–415, 2019.

[20] Michael J Lee and Andrew J Ko. Comparing the effectiveness of online learning approaches on cs1 learning outcomes. In *ACM ICER*, pages 237–246, 2015.

[21] John H Maloney, Kylie Peppler, Yasmin Kafai, Mitchel Resnick, and Natalie Rusk. *Programming by Choice: Urban Youth Learning Programming with Scratch*, volume 40. ACM, 2008.

[22] Orni Meerbaum-Salant, Michal Armoni, and Mordechai Ben-Ari. Learning computer science concepts with scratch. *Computer Science Education*, 23(3):239–264, 2013.

[23] U.S. Bureau of Labor Statistics. Occupational outlook handbook, 2018.

[24] David A Patterson. Restoring the popularity of computer science. *Communications of the ACM*, 48(9):25–28, 2005.