

Exploring Differences in Minority Students' Attitudes towards Computing after a One-Day Coding Workshop

Michael J. Lee

New Jersey Institute of Technology
Newark, New Jersey, USA
mjlee@njit.edu

ABSTRACT

As programming continues to be an essential 21st century skill, it is critical to focus on diversity and increasing participation of under-represented groups in computing. To address this need, we must better understand minorities' views and attitudes towards programming, especially in their youth, as literature shows that children form ideas about their interests and careers in middle school or earlier. To explore this, we provided middle school students in the U.S. with a full day (7 hours) of programming activities to learn about their initial attitudes towards computing and how a short intervention might change these attitudes. We ran two separate one-day events, serving a total of 34 minority students (21 males and 13 females; grades 6 and 7) from a low-income, urban area. We found that students' initial attitudes towards computing were high, and that one day of learning programming increased their reported attitudes in computing even more. We also found differences in attitudes by gender and ethnicity. These findings highlight the positive attitudes minority students have towards computing, and the importance of providing resources and support to help maintain their interests in computing while recognizing demographic differences.

CCS CONCEPTS

• **Social and Professional Topics** → **Computing Education.**

KEYWORDS

Computing education; Middle school; Minorities; Attitudes

ACM Reference Format:

Michael J. Lee. 2019. Exploring Differences in Minority Students' Attitudes towards Computing after a One-Day Coding Workshop. In *Innovation and Technology in Computer Science Education (ITiCSE '19)*, July 15–17, 2019, Aberdeen, Scotland, UK. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3304221.3319736>

1 INTRODUCTION

Software is transforming the world economy, creating significant demand for programming skills in science, technology, education, health, and other rapidly growing fields [37]. Meanwhile, programming and computing-related jobs are among the fastest growing

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ITiCSE '19, July 15–17, 2019, Aberdeen, Scotland, UK

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-6301-3/19/07...\$15.00

<https://doi.org/10.1145/3304221.3319736>



Figure 1: Sussex Avenue Renew School students (6th and 7th grade) playing the Gidget game for their morning activity.

career opportunities [10], growing faster than all other occupations in some parts of the world such as in the U.S. [37]. However, many of these positions will go unfilled as there are not enough people with the right training/skills to take these jobs [17, 50]. Moreover, for positions that do get filled, the diversity of both the workforce and undergraduate population continues to be dominated by certain groups (e.g., males), contributing to severe inequities and bias in software design and use [38], including disproportionately low participation by minority groups. Without new learning opportunities that both engage a wider range of the population and sustain their learning and engagement over time, we cannot adequately address the lack of diversity and low supply of technology workers.

Many youth are not choosing to pursue computing-related subject fields, further increasing the lack of representation in computing (this is particularly true for underrepresented groups in the U.S. [37]). Although many computing related enrichment activities are available to high school students, research has shown that youth begin to form ideas about future careers during their formative years in middle school [11, 15, 21].

This project attempts to address inequities in computing by learning more about low-income, minority, middle school students' initial views and attitudes towards computing and how a one day programming workshop (event) might change these attitudes. We surveyed middle school students using validated instruments, before and after they were exposed to programming with a debugging game and block programming environment. Our goal was to provide minority students a positive experience with programming, and also to share our findings about these students' attitudes towards computing so that educators and researchers can better design effective programming interventions to motivate these students who are underrepresented in computing.

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 [11, 15, 21]. Many educational research efforts targeting younger children have focused on how to engage them in 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., [32]), Alice (e.g., [25, 26]), Gidget [31], and others [19]. Overall, studies repeatedly found that these types of environments are effective in the acquisition of basic programming skills and concepts (e.g., [29, 31, 32]) and support computational thinking. For example, Meerbaum-Salant et al. [34] found that Scratch helped middle schoolers learn most of the targeted CS concepts. Lee & Ko [31] 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 and minority middle students and their attitudes towards computing, both before and after working with some of the programming environments mentioned above.

2.2 Females and Minority Students

Females are underrepresented in computing [41]. Research shows that social environment influences many girls to exclude considering computing career choices by the time they enter high school, even before they have been exposed to any computing instruction [36, 43, 47]. During their formative time in middle school, girls are deeply engaged in forming confidence in STEM disciplines [1, 18, 48] but might be more risk averse [9, 22, 46] and have lower self-efficacy and self-confidence in computing than their male peers [2, 4, 8, 33]. For example, females tend to have lower confidence in computer-based problem-solving tasks like programming [2, 6], which may make them less likely to pursue computing activities on their own [33]. They also have a lower likelihood of trying new software features [2, 6] and a greater reluctance to tinker with code or programming tools [3, 6]. We aim to make our events gender-inclusive [7] to appeal to a wide range of people with different preferences and computing skill.

Members of ethnic minority groups are also largely underrepresented in computing. Availability of computing resources such as internet access has been reported to be significantly related to household income and inversely affected by minority status [24]. Even when minorities gain access to educational computing resources, they are often taught rote learning activities rather than cognitively demanding tasks [45] or patronized by their instructors, mentors, and peers [16, 20]. Young minorities that express interest in computing may stand out and be dissuaded by their friends and family [12], so they may avoid or quit pursuing computing. We hypothesize that early computing instruction in a safe and positive environment will counter many of these negative effects.

There have been numerous efforts to increase participation of young underrepresented minorities and females in computing activities. Robinson & Pérez-Quinones, were able to change underrepresented middle school girls' initial negative perceptions of computing by teaching them human-computer interaction (HCI) methods

through paper prototyping over a week-long workshop [40]. Similarly, Doerschuk et al., found that their underrepresented middle school students' self-reports of computer science knowledge increased significantly after a one-day workshops run by undergraduate students who created the programming instructional materials [14]. Denner analyzed data from middle school students on the first day of a computing after-school program and found that the strongest direct predictor of girls' interest in computing classes and careers was the extent to which they see value in computing, and perceived support from peers and adults (teachers and parents) [11]. Our experience report adds to this body of knowledge, using validated survey instruments and similar curricula to learn more about underrepresented students' attitudes towards computing and how we might be able to encourage more participation.

3 METHOD

The purpose of our experience report is to investigate what minority middle school students think of computing, and how programming—in the form of a concrete learning experience using an educational programming game and block programming environment—might change their initial attitudes. Our hypotheses were:

- (1) Minority children will have positive views towards computing.
- (2) Minority children's attitudes towards computing will increase (positively) after a basic educational programming experience.

3.1 Location and Recruitment

We ran two separate one-day, Saturday events during different semesters of the school year. The first took place at Sussex Avenue Renew School, a local K-8 public school in Newark, New Jersey (U.S.). The second took place at the Urban League of Essex County's office (a local non-profit organization), and included students from two other K-8 public schools from Newark. The schools' demographic makeup reflects the community they serve, composed of 52% Hispanics, 47% African Americans, and 1% others; 82% of these students are low-income—eligible for free or reduced-price lunch.

Principals recruited students from their respective schools by recommendations from their 6th and 7th grade teachers, who made announcements in their classes. Venues for both events included a classroom, equipment (e.g., laptops with WiFi access), a middle school teacher (one male technology teacher at the first event, and one female technology teacher at the second event), and two male undergraduate student instructors. The same two instructors ran both 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). All of the middle school students were dropped off and picked up by their respective parents/guardians.

3.2 Procedure

Although traditional survey measures are commonly used to understand views toward computing [29], they often ask about attitudes abstractly, or without exposing participants to actual programming. Our seven-hour day consisted of breakfast, taking a pretest, a programming game activity, lunch, a block programming activity, and then a posttest. Breakfast was 15 minutes, lunch was 45 minutes, and the surveys took 30 minutes each. The game and block programming game were each 2.5 hours. In more detail:

3.2.1 Morning Activity. For their opening activity, students played Gidget [30, 31] (www.helpgidget.org)—a free, online, educational programming game—to give them a positive introduction to coding, as one's first experience with code is important [28]. Jernigan et al. [23] and Lee et al. [30] demonstrated that the game is engaging for a wide range of programming novices and that it is a good introduction to (text-based) programming for after-school programs. We encouraged our middle school students to complete as many levels as they could during the scheduled 2.5 hours, as each subsequent level in Gidget covers progressively difficult introductory programming (CS1) concepts [30].

3.2.2 Afternoon Activity. For our afternoon closing activity, our events used the Scratch "Creative Computing" curriculum [5] 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 [32, 34]. The morning session consisted of the "Unit 0: Getting Started" section of Harvard University's Creative Computing curriculum for 2.5 hours, giving students an introduction to Scratch, its features, and how to use block programming with examples that they could follow to make animations.

3.2.3 Pre and Post Tests. We used two existing, tested, and validated paper-based instruments to collect the data from students: 1) the Computer Attitudes Questionnaire (CAQ) v5.14 [27] (designed for middle school students), and 2) the Computing Attitudes Survey (CAS) v4.0 [42]. The former measures general computing attitudes (e.g., "I enjoy doing things on a computer.") whereas the latter specifically measures computer science programming attitudes (e.g., "After I study a topic in computer science and feel that I understand it, I have difficulty solving problems on the same topic.") and is designed to be an extension of the former.

Our pre and post tests consisted of the first 20 questions from the CAQ (labeled as "Part 1" in the original document [27]), and all 26 of the CAS questions [42]. For the CAS, we replaced all occurrences of "computer science" with "computing" to keep the scope of the questions intentionally broad and to minimize potential confusion (especially during the pretest, since students had not started the programming activities yet). We added two questions to the end of the test, Q46: "I am interested in learning more about computer programming," and Q47: "I can become a computer programmer." We also inserted one attention check question towards the end of the questionnaire, "Please choose 'agree' (number 4) for this answer," to help ensure that students were paying attention to the 48 questions. All answer choices were in the form of a 5-point Likert scale [39] to determine how agreeable students were to the statements (with 1 being least agreeable, and 5 being most agreeable).

There were three additional 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. Then, only for the pretest (since we would not expect responses to change during the course of the day), 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.

4 RESULTS

It took the middle school students approximately 30 minutes to complete each of the pre and post tests. In total, we had 34 students ($n = 34$) participate in our event (11–13 years old). The first event included 12 boys and 6 girls (seven 6th graders and eleven 7th graders; see Figure 1), and the second event included 9 boys and 7 girls (seven 6th graders and nine 7th graders). All of the students were underrepresented minorities in STEM, identifying as either African American (AA students; 8 each at the events) or Hispanic/Latino (HL Students; 10 at the first event, and 8 at the second event), and all were eligible for free/reduced cost lunch.

We report the median of our (ordinal) responses. Throughout this analysis, we use nonparametric Wilcoxon/Kruskal-Wallis rank sums tests (for differences between pre and post tests) and Chi-Squared likelihood ratio tests (for differences within a pretest or posttest) with $\alpha = 0.05$ confidence—as our data was not normally distributed—to compare participants' responses based on their demographic categories. We report our statistically significant results in Tables 1 through 4 with the understanding that our sample size is small and that the resulting statistics may not be widely generalizable. As both of our events used the same schedule and instructors, and all the students came from the same school district with comparable demographic distributions (i.e., grade, gender, ethnicity, and eligibility for free/reduced cost lunch), we report our results aggregating all event participants into one group.

4.1 Computing Attitude (CAQ & CAS) Results

4.1.1 Pretest Scores (Initial Attitudes). We examined the students pretest scores to gauge their initial attitudes towards computing before taking part in our activity. Overall, the students reported positive attitudes towards computing for on the pretest, with medians of 4 ("agree") or 5 ("strongly agree") for all positive valence questions (e.g., Q11: "I believe that it is important for me to learn how to use a computer," and Q31: "I find the challenge of solving computing problems motivating."), medians of 1 ("strongly disagree") through 3 ("neither agree nor disagree") for all negative valence questions (e.g., Q16, "Working with a computer makes me nervous," and Q45: "I worry that mistakes I make when using a computer may damage my computer."), and medians of 2 ("disagree") or 3 ("neither agree nor disagree") for all neutral valence questions (e.g., Q23: "If I want to apply a method used for solving one computing problem to another problem, the problems must involve very similar situations.").

We found some differences in response by ethnicity and gender in the pretest scores. For example, Q7 ("I know that computers give me opportunities to learn many new things.") and Q20 ("I can learn more from books than from a computer.") had significantly different responses by ethnicity, where AA students were more likely to agree with both statements compared to their HL classmates (see Table 1). Additionally, Q30 ("When working on a computing problem, I brainstorm about solution strategies before trying to solve it.") had significantly different responses on the pretest by ethnicity, where AA students were more likely to agree with this statement compared to their HL classmates (see Table 1).

For Q10 ("I believe that the more often teachers use computers, the more I'll enjoy school."), males were significantly more likely to agree with the statement over their female classmates (see Table 2).

Question (Pretest-Top/Posttest-Bottom):	Statistic	p.val
7. I know that computers give me opportunities to learn many new things.	$\chi^2(4, N=34)=9.988$ (AA is Higher)	.041*
20. I can learn more from books than from a computer.	$\chi^2(4, N=34)=10.67$ (AA is Higher)	.031*
30. When working on a computing problem, I brainstorm about solution strategies before trying to solve it.	$\chi^2(3, N=34)=8.831$ (AA is Higher)	.032*
3. I will be able to get a good job if I learn how to use a computer.	$\chi^2(1, N=33)=5.405$ (AA is Higher)	.020*
27. There are times I solve a computing problem in more than one way to help with my understanding.	$\chi^2(3, N=34)=9.371$ (AA is Higher)	.025*
28. I think about the computing technologies or problems I experience in everyday life.	$\chi^2(3, N=34)=7.917$ (AA is Higher)	.048*
45. I worry that mistakes I make when using a computer may damage my computer.	$\chi^2(4, N=34)=9.550$ (HL is Higher)	.049*

AA = African American/Black Students; HL = Hispanic/Latino Students

Table 1: Pretest & Posttest Differences by Ethnicity

Next, Q33 ("I enjoy solving computing problems."), Q40 ("Understanding computing basically means being able to recall something you've read or been shown."), and Q44 ("To learn about computing, I only need to memorize solutions to sample problems."), male students were significantly more likely to agree with these statements than their female classmates (see Table 2).

4.1.2 Posttest Scores (Post-Activity Attitudes). Like the pretest, students reported highly positive attitudes towards computing on the posttest with some minor improvements for some questions. There were no statistically detectable differences by gender within the posttest scores. However, we did find a few questions that had statistically significant difference responses by ethnicity (see Table 1).

For example, in Q3 ("I will be able to get a good job if I learn how to use a computer."), AA students were significantly more likely to agree with the statement over their HL classmates. Additionally, Q27 ("There are times I solve a computing problem in more than one way to help with my understanding.") and Q28 ("I think about the computing technologies or problems I experience in everyday life.") had significantly different responses on the posttest by ethnicity, where AA students were more likely to agree with both statements compared to their HL classmates (see Table 1). Conversely, on Q45 ("I worry that mistakes I make when using a computer may damage my computer."), HL students were more likely to agree with the statement compared to their AA classmates (see Table 1).

4.1.3 Comparison between Pre and Post Test Scores. Table 3 lists the questions where statistically significant differences were detected comparing the posttest and pretest. Responses from seven questions (two of the twenty from the CAQ, and five of twenty-six from the CAS) were significantly different between the tests.

Question (Pretest):	Statistic	p.val
10. I believe that the more often teachers use computers, the more I'll enjoy school.	$\chi^2(3, N=34)=8.657$ (Males Rate Higher)	.034*
33. I enjoy solving computing problems.	$\chi^2(2, N=34)=6.730$ (Males Rate Higher)	.035*
40. Understanding computing basically means being able to recall something you've read or been shown.	$\chi^2(3, N=34)=9.890$ (Males Rate Higher)	.020*
44. To learn about computing, I only need to memorize solutions to sample problems.	$\chi^2(3, N=33)=8.769$ (Males Rate Higher)	.033*

Table 2: Pretest Differences by Gender

For example, Q3 ("I will be able to get a good job if I learn how to use a computer.") and Q4 ("I can concentrate on a computer when I use one.") having statistically higher posttest values than the pretest values in both cases (see Table 3). Negative valence questions, Q22 ("Errors made by computers are random, and when they happen, there's not much I can do to understand why."), and Q41 ("If I get stuck on a computing problem, there is no chance I'll figure it out on my own."), both had statistically significant lower posttest values than the pretest in both cases (demonstrating an improvement in attitudes). Likewise, for positive valence questions, Q25 ("When I solve a computing problem, I break it into smaller parts and solve them one at a time."), Q33 ("I enjoy solving computing problems."), and Q35 ("Learning about computing is just about learning how to think differently."), all had statistically higher posttest values compared to pretest values.

4.1.4 Comparison by Gender and Ethnicity. We also compared the difference between the posttest and pretest scores by gender and ethnicity. Although there were no statistically significant differences detected between the pretest and posttest scores by gender, we did find several differences by ethnicity (see Table 4).

We found that for Q3 ("I will be able to get a good job if I learn how to use a computer."), our AA students' scores from their pretest to their posttest increased significantly more than those of their HL classmates (Table 4). We also found several questions where HL students' scores improved significantly in their posttest over their pretest compared to their AA classmates. These included, Q4 ("I concentrate on a computer when I use one."), and Q33 ("I enjoy solving computing problems."). Q19 ("Computers are difficult to use.") has a negative valence, with the posttest score being significantly lower, meaning that HL students were more likely to report that computers were not as difficult to use in their posttest than their pretest compared to their AA classmates (see Table 4).

For negative valence questions, Q23 ("If I want to apply a method used for solving one computing problem to another problem, the

Question:	Statistic	p.val
3. I will be able to get a good job if I learn how to use a computer. Pre- Median: 4 (range 3-5) / Post- Median: 5 (range 4-5)	W=1336.5, Z=3.189 (Post is Higher)	.001*
4. I can concentrate on a computer when I use one. Pre- Median: 4 (range 3-5) / Post- Median: 5 (range 4-5)	W=1323.5, Z=2.00 (Post is Higher)	.045*
22. Errors made by computers are random, and when they happen, there's not much I can do to understand why. Pre- Median: 4 (range 1-5) / Post- Median: 2 (range 1-5)	W=1023, Z=-1.997 (Post is Lower)	.046*
25. When I solve a computing problem, I break it into smaller parts and solve them one at a time. Pre- Median: 3 (range 1-5) / Post- Median: 3.5 (range 1-5)	W=1336, Z=2.044 (Post is Higher)	.041*
33. I enjoy solving computing problems. Pre- Median: 4 (range 3-5) / Post- Median: 4 (range 1-5)	W=1335, Z=2.092 (Post is Higher)	.036*
35. Learning about computing is just about learning how to think differently. Pre- Median: 4 (range 2-5) / Post- Median: 5 (range 3-5)	W=1280.5, Z=2.106 (Post is Higher)	.035*
41. If I get stuck on a computing problem, there is no chance I'll figure it out on my own. Pre- Median: 3 (range 1-4) / Post- Median: 2 (range 1-5)	W=999, Z=-2.245 (Post is Lower)	.025*
48. I can become a computer programmer. Pre- Median: 3 (range 3-5) / Post- Median: 5 (range 3-5)	W=1347, Z=2.276 (Post is Higher)	.023*

Table 3: Comparison of Pre/Post Test (General)

Question:	Statistic	p.val
3. I will be able to get a good job if I learn how to use a computer. Pre- Median: 4 (range 3-5) / Post- Median: 5 (range 4-5)	W=336,Z=3.116 (AA Higher Post)	.002*
4. I concentrate on a computer when I use one. Pre- Median: 4 (range 3-5) / Post- Median: 5 (range 4-5)	W=398.5,Z=2.204 (HL Higher Post)	.028*
19. Computers are difficult to use. Pre- Median: 2 (range 1-5) / Post- Median: 1 (range 1-5)	W=265.5,Z=-2.28 (HL Higher Pre)	.022*
23. If I want to apply a method used for solving one computing problem to another problem, the problems must involve very similar situations. Pre- Median: 3 (range 1-5) / Post- Median: 3 (range 1-5)	W=211,Z=-2.028 (AA Higher Pre)	.043*
25. When I solve a computing problem, I break it into smaller parts and solve them one at a time. Pre- Median: 3 (range 1-5) / Post- Median: 3.5 (range 1-5)	W=314.5,Z=1.967 (AA Higher Post)	.049*
33. I enjoy solving computing problems. Pre- Median: 4 (range 3-5) / Post- Median: 4 (range 1-5)	W=402.5,Z=2.296 (HL Higher Post)	.022*
35. Learning about computing is just about learning how to think differently. Pre- Median: 4 (range 2-5) / Post- Median: 5 (range 3-5)	W=364,Z=2.030 (HL Higher Post)	.042*
41. If I get stuck on a computing problem, there is no chance I'll figure it out on my own. Pre- Median: 3 (range 1-4) / Post- Median: 2 (range 1-5)	W=208,Z=-2.251 (AA Higher Pre)	.024*
48. I can become a computer programmer. Pre- Median: 3 (range 3-5) / Post- Median: 5 (range 3-5)	W=318,Z=2.161 (AA Higher Post)	.031*

AA = African American/Black Students; HL = Hispanic/Latino Students

Table 4: Comparison of Pre/Post Test by Ethnicity

problems must involve very similar situations."), and Q41 ("If I get stuck on a computing problem, there is no chance I'll figure it out on my own."), AA students reported statistically significant lower agreement in the posttest for all three questions (demonstrating an improvement in attitudes) compared to their HL classmates.

For the positive valence question Q25 ("When I solve a computing problem, I break it into smaller parts and solve them one at a time."), AA students reported statistically significant higher agreement in their posttest than their pretest compared to HL students. Similarly, for Q33 ("I enjoy solving computing problems.") and Q35 ("Learning about computing is just about learning how to think differently."), HL students reported statistically significant higher agreement in their posttest than their pretest compared to their AA classmates.

4.2 Additional Question Results

For Q48 ("I can become a computer programmer"), we found that the posttest scores were significantly different from the pretest (median 3), with the posttest (median 5) scoring much higher (see Table 3). Moreover, we found a statistically significant difference when comparing posttest and pretest scores for this question, where AA students were more likely to agree with the statement compared to their HL classmates (see Table 4).

All (34 of 34) students reported that they wanted to attend college, and would take more classes *at* school to learn more about computers and computer programming. All but one student (33 of 34) reported that they would want to take more classes *outside* of school to learn more about computers and computer programming.

For our pretest-only questions, 25 of 34 (73.5%) students reported that they owned a smartphone. Additionally, 27 of 34 (79.4%) students indicated they had a computer at home. These same students also had access to the Internet at home. Those 7 of 34 (20.6%) who stated not having a computer at home also did not have Internet access at home. 10 of the 34 (29.4%) students reported they had some previous programming experience (all using Scratch).

5 DISCUSSION

5.1 Initial View of Computing & Programming

We confirm our first hypothesis that our minority middle school students had initially positive attitudes towards computing and programming, before participating in our activity (Sections 4.1.1 and 4.2.1). In addition to our survey results, our students were open to talking (positively) about computing and appeared to be genuinely interested in partaking in the day's computing activities along with their classmates. This is similar to DiSalvo et al.'s findings that AA students interested in a computing activity were eager to talk about it with their peers in the context of use, but not outside [13]. Unfortunately, we are unable to determine if our students would be willing to talk to other peers about their computing activities outside, as we only interacted with them during the one day event(s). However, we observed that our students were eager to learn, and admired their peers with more experience/skill in programming. These skilled students also spent time helping their less-experienced peers with different concepts and issues throughout the day.

5.1.1 Comparison by Gender and Ethnicity. We found a total of four significantly different scores in the students' initial (i.e., pretest) computing attitudes by gender. Based on the responses, it appears that our male students had higher computing self-efficacy and wanted to use computers in their classes more often than their female classmates (see Q10 and Q33 in Table 2). However, it also appears that our male students believed that computing mostly involved rote learning (see Q40 and Q44 in Table 2). Overall, these differences found by gender are not entirely unexpected, as previous research has shown that females tend to have lower self-efficacy in computing than males [2, 3, 6].

We also found a total of three significantly different scores in the students' initial computing attitudes by ethnicity. Our AA students were more likely than their HL classmates to agree that computers provide opportunities to learn new things and that they brainstorm solutions before working on computing problems (see Q7 and Q30 in Table 1). However, these AA students were also more likely than their HL counterparts to agree that they could learn more from books than computers (Q20, Table 1). These findings on AA students' views towards success are similar to those described in the narratives of successful AA college students by Urias et al. [44], and highlight the need for educators to rethink how to deliver computing education that will both engage and set them up for success. This also highlights the need to learn more about the various, distinct perceptions that different minority groups may have towards computing, as a one-size-fits all model for teaching and engaging these different groups may not be ideal.

5.1.2 Post Views of Computing. Addressing our second hypothesis, we found an increase in several of the reported attitudes by our students towards computing after participating in our activity.

We found a total of eight questions where responses in the posttest were significantly different from the pretest. The first were related to jobs, where students agreed on their posttest that they could get a good job if they learned how to use a computer, and that they could become a computer programmer themselves (Q3 and Q48, Table 3). Students also reported on their use of computers and solving problems, where they said they can concentrate on

computers when they use one, that they enjoy solving computing problems, and to do so, they can break it into smaller pieces to solve individually (Q4, Q25, and Q33, Table 3).

Additionally, these students better understood how computers worked, stating that the errors made by computers are not random (Q22), learning computing is about thinking differently (Q35), and that they can figure out on their own how to overcome difficult computing problems (Q41). For Q48 ("I can become a computer programmer."), students originally had a median of 3 ("neither agree nor disagree") (Table 3), but this increased significantly to a median of 5 ("highly agree") by the end of our event. This result was especially pronounced for our AA students (Table 4).

These results demonstrate that a short, one-day event teaching programming concepts can have significant impacts on students' understanding of computing, how to solve programming problems, and increase their confidence in becoming programmers themselves. Future work will explore how to sustain these students' interest and attitudes, and what types of activities are effective in doing so.

5.1.3 Comparison by Ethnicity and Gender. We also found several questions where responses in the posttest were significantly different from the pretest based on ethnicity and by gender. Examining the responses, it appears that our AA students became much more confident in their own ability to overcome obstacles in computing after participating in the event compared to their HL classmates.

First, AA students were more likely to agree that learning how to use a computer could get them a good job (Q3). They also became more confident in their computing abilities, understanding that computing problems can be solved in multiple ways (Q23), and that they have the capability to solve these problems on their own (Q44), using strategies such as dividing a problem into smaller parts (Q25).

Our HL students' computing self-efficacy also improved significantly between their pretest and posttest scores, but differently from their AA classmates. They reported that they concentrated on computers when they were using one (Q4), that computers are not difficult to use (Q19), that they enjoy solving computing problems (Q33), and that learning about computing is just a matter of thinking differently (Q35, Table 4).

Next, we found that the gender differences described in 5.1.1 completely disappeared by the end of the event, suggesting that something about our event raised the female students' self-efficacy and understanding of computing. We do not believe that the gender of the other participants (21 of 34; 61.8% were male), instructors (two males), or teachers (one male and one female) largely attributed to this positive change in female students, as there were fewer females overall. Instead, we suspect that the events' programming activities attributed to this effect. According to its developers, Gidget was designed using a gender-inclusive approach and was found to engage high school females in programming summer camps [7, 30]. Researchers examining Scratch also did not find any gender differences in the number of projects created [35], nor with participation patterns or project complexity [49], suggesting that Scratch provides similar opportunities and engagement across genders. We plan to examine these issues in future events to help identify specific factors that may engage (or disengage) females.

These findings demonstrate that there are major differences in the attitude changes between different minority groups and

genders. Future work will further examine these differences so that educators and researchers can better understand how to create learning interventions that support different groups of learners.

5.2 Limitations and Future Work

We had significant results in several of our measures. However, we had two activities throughout the day (playing a debugging game and making animations with a block programming language), making it difficult to isolate the specific intervention or factors that affected students' responses. Future events can offer only one activity throughout the day so that we can better understand the specific intervention's effect.

We reported statistics combining participants from our two events into one group. We believe this is acceptable, as the demographic makeup of both groups were comparable, all students attended public schools in the same school district, and both events followed the same structure/curriculum with the same instructors.

All of our students were recruited by participating schools' principals, which may have introduced a selection bias (e.g., students who signed up for a Saturday programming event are likely to have positive views of computing). Moreover, we had a fairly small number of participants overall and though we found some statistically significant results for some demographic factors, we were unable to detect anything at the intersection of these factors (e.g., the interaction between gender and ethnicity on the responses). These issues may limit the generalizability of our results. In the future, we will work with additional schools in recruiting a larger number of participants and seek to minimize effects of potential selection biases (e.g., recruiting an entire class if possible).

6 CONCLUSION

We worked with 34 minority middle school students across two Saturday events, exposing them to programming through an educational game and a block programming environment. We found that these students' initial views and attitudes towards computing were positive. We also found that the day of activities increased these positive views even further, with a significant number of them stating that they could become a computer programmer themselves.

We also found some differences by gender and ethnicity in students' initial attitudes towards computing, and also after they finished their activity. These findings highlight the need to further examine the differences between minority groups and even across different demographic factors within these groups. Better understanding of these differences can lead to more targeted and effective programming-related interventions to help increase minorities engagement and representation in computing.

ACKNOWLEDGMENTS

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. 1986. Social Foundation of Thought and Action: A Social-cognitive View. *Englewood Cliffs* (1986).
- [2] Laura Beckwith, Margaret Burnett, Susan Wiedenbeck, Curtis Cook, Shraddha Sorte, and Michelle Hastings. 2005. Effectiveness of End-user Debugging Software Features: Are there Gender Issues?. In *SIGCHI Conference on Human Factors in Computing Systems*. ACM, 869–878.
- [3] Laura Beckwith, Cory Kissinger, Margaret Burnett, Susan Wiedenbeck, Joseph Lawrance, Alan Blackwell, and Curtis Cook. 2006. Tinkering and Gender in End-user Programmers' Debugging. In *SIGCHI Conference on Human Factors in Computing Systems*. ACM, 231–240.
- [4] Joel Brandt, Mira Dontcheva, Marcos Weskamp, and Scott R Klemmer. 2010. Example-centric Programming: Integrating Web Search into the Development Environment. In *SIGCHI Conference on Human Factors in Computing Systems*. ACM, 513–522.
- [5] K Brennan, M Chung, and J Hawson. 2011. Scratch Curriculum Guide: A Design-based Introduction to Computational Thinking with Scratch. (2011).
- [6] Margaret Burnett, Scott D Fleming, Shamsi Iqbal, Gina Venolia, Vidya Rajaram, Umer Farooq, Valentina Grigoreanu, and Mary Czerwinski. 2010. Gender Differences and Programming Environments: Across Programming Populations. In *ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 28.
- [7] Margaret M Burnett, Elizabeth F Churchill, and Michael J Lee. 2015. SIG: Gender-Inclusive Software: What We Know About Building It. In *ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 857–860.
- [8] Tor Busch. 1996. Gender, Group Composition, Cooperation, and Self-efficacy in Computer Studies. *Journal of Educational Computing Research* 15, 2 (1996), 125–135.
- [9] James P Byrnes, David C Miller, and William D Schafer. 1999. Gender differences in risk taking: a meta-analysis. *Psychological Bulletin* 125, 3 (1999), 367.
- [10] Anthony P Carnevale, Nicole Smith, and Michelle Melton. 2011. STEM: Science Technology Engineering Mathematics. *Georgetown University Center on Education and the Workforce* (2011).
- [11] Jill Denner. 2011. What Predicts Middle School Girls' Interest in Computing? *International Journal of Gender, Science and Technology* 3, 1 (2011).
- [12] Betsy DiSalvo, Mark Guzdial, Amy Bruckman, and Tom McKlin. 2014. Saving Face while Geeking Out: Video Game Testing as a Justification for Learning Computer Science. *Journal of the Learning Sciences* 23, 3 (2014), 272–315.
- [13] Betsy DiSalvo, Sarita Yardi, Mark Guzdial, Tom McKlin, Charles Meadows, Kenneth Perry, and Amy Bruckman. 2011. African American Men Constructing Computing Identity. In *SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2967–2970.
- [14] Peggy Doerschuk, Jiangjiang Liu, and Judith Mann. 2009. INSPIRED Computing Academies for Middle School Students: Lessons Learned. In *Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations*. ACM, 52–57.
- [15] Jacquelynne S Eccles and Rena D Harold. 1993. Parent-school Involvement during the Early Adolescent Years. *Teachers College Record* (1993).
- [16] Erwin Flaxman. 1992. The Mentoring Relationship in Action. *The Institute for Urban & Minority Education Briefs* 3 (1992).
- [17] Gary Gereffi, Vivek Wadhwa, Ben Rissing, Kiran Kalakuntla, Soomi Cheong, Qi Weng, and Nishanth Lingamneni. 2005. Framing the Engineering Outsourcing Debate: Placing the United States on a Level Playing Field with China and India. *Duke University Report* (2005).
- [18] Judith Gill. 1994. Shedding Some New Light on Old Truths: Student Attitudes to School in Terms of Year Level and Gender. (1994).
- [19] Shuchi Grover, Roy Pea, and Stephen Cooper. 2016. Factors Influencing Computer Science Learning in Middle School. In *ACM Technical Symposium on Computer Science Education (SIGCSE)*. ACM, 552–557.
- [20] Eleanor Guetzloe. 1997. The Power of Positive Relationships: Mentoring Programs in the School and Community. *Preventing School Failure: Alternative Education for Children and Youth* 41, 3 (1997), 100–104.
- [21] Nancy E Hill and Ming-Te Wang. 2015. From Middle School to College: Developing Aspirations, Promoting Engagement, and Indirect Pathways from Parenting to Post High School Enrollment. *Developmental Psychology* 51, 2 (2015), 224.
- [22] Gerald A Hudgens and Linda Torsani Fatkin. 1985. Sex Differences in Risk Taking: Repeated Sessions on a Computer-simulated Task. *The Journal of Psychology* 119, 3 (1985), 197–206.
- [23] William Jernigan, Amber Horvath, Michael Lee, Margaret Burnett, Taylor Cully, Sandeep Kuttal, Anicia Peters, Irwin Kwan, Faezeh Bahmani, Andrew Ko, et al. 2017. General Principles for a Generalized Idea Garden. *Journal of Visual Languages & Computing* 39 (2017), 51–65.
- [24] Yasmin B Kafai, Kylie A Peppler, and Grace M Chiu. 2007. High Tech Programmers in Low-income Communities: Creating a Computer Culture in a Community Technology Center. In *Communities and Technologies*. Springer, 545–563.
- [25] Caitlin Kelleher and Randy Pausch. 2007. Using Storytelling to Motivate Programming. *Commun. ACM* 50, 7 (2007), 58–64.
- [26] Jordana Kerr, Mary Chou, Reilly Ellis, and Caitlin Kelleher. 2013. Setting the Scene: Scaffolding Stories to Benefit Middle School Students Learning to Program. In *Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 95–98.
- [27] G Knezek and R Christensen. 1997. Attitudes Toward Information Technology at Two Parochial Schools in North Texas. Retrieved September 5 (1997), 2005.
- [28] Andrew J Ko. 2009. Attitudes and Self-efficacy in Young Adults' Computing Autobiographies. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 67–74.
- [29] D Midian Kurland and Roy D Pea. 1985. Children's Mental Models of Recursive LOGO Programs. *Journal of Educational Computing Research* 1, 2 (1985), 235–243.
- [30] Michael J Lee, Faezeh Bahmani, Irwin Kwan, Jilian LaFerte, Polina Charters, Amber Horvath, Fanny Luor, Jill Cao, Catherine Law, Michael Beswetherick, et al. 2014. Principles of a Debugging-first Puzzle Game for Computing Education. In *Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 57–64.
- [31] Michael J Lee and Andrew J Ko. 2015. Comparing the Effectiveness of Online Learning Approaches on CS1 Learning Outcomes. In *ACM International Computing Education Research (ICER)*. ACM, 237–246.
- [32] John H Maloney, Kylie Peppler, Yasmin Kafai, Mitchel Resnick, and Natalie Rusk. 2008. *Programming by Choice: Urban Youth Learning Programming with Scratch*. Vol. 40. ACM.
- [33] Jane Margolis and Allan Fisher. 2003. *Unlocking the Clubhouse: Women in Computing*. MIT press.
- [34] Orni Meerbaum-Salant, Michal Armoni, and Mordechai Ben-Ari. 2013. Learning Computer Science Concepts with Scratch. *Computer Science Education* 23, 3 (2013), 239–264.
- [35] Andrés Monroy-Hernández and Mitchel Resnick. 2015. Empowering Kids to Create and Share Programmable Media. *arXiv preprint arXiv:1507.01282* (2015).
- [36] NCWIT. [n. d.]. National Center for Women and Information Technology. ([n. d.]). Retrieved April 21, 2018 from <https://www.ncwit.org/>
- [37] U.S. Bureau of Labor Statistics. 2018. Occupational Outlook Handbook. (2018). Retrieved April 21, 2018 from <https://www.bls.gov/ooh/>
- [38] David A Patterson. 2005. Restoring the Popularity of Computer Science. *Commun. ACM* 48, 9 (2005), 25–28.
- [39] Melanie A Revilla, Willem E Saris, and Jon A Krosnick. 2014. Choosing the Number of Categories in Agree–Disagree Scales. *Sociological Methods & Research* 43, 1 (2014), 73–97.
- [40] Ashley Robinson and Manuel A Pérez-Quinones. 2014. Underrepresented Middle School Girls: On the Path to Computer Science through Paper Prototyping. In *ACM Technical Symposium on Computer Science Education (SIGCSE)*. ACM, 97–102.
- [41] Jennifer A Rode, Eleanor F Toye, and Alan F Blackwell. 2004. The Fuzzy Felt Ethnography—Understanding the Programming Patterns of Domestic Appliances. *Personal and Ubiquitous Computing* 8, 3–4 (2004), 161–176.
- [42] Allison Tew, Brian Dorn, and Oliver Schneider. 2012. Toward a Validated Computing Attitudes Survey. In *ACM International Computing Education Research (ICER)*. ACM, 135–142.
- [43] Lisa Tsui. 2007. Effective Strategies to Increase Diversity in STEM Fields: A Review of the Research Literature. *The Journal of Negro Education* (2007), 555–581.
- [44] Marissa Vasquez Urias, Vannessa Falcon, Frank Harris III, and J Luke Wood. 2016. Narratives of Success: A Retrospective Trajectory Analysis of Men of Color who Successfully Transferred from the Community College. *New Directions for Institutional Research* 2016, 170 (2016), 23–33.
- [45] Mark Warschauer. 2004. *Technology and Social Inclusion: Rethinking the Digital Divide*. MIT press.
- [46] Elke U Weber, Ann-Renee Blais, and Nancy E Betz. 2002. A Domain-specific Risk-attitude Scale: Measuring Risk Perceptions and Risk Behaviors. *Journal of Behavioral Decision Making* 15, 4 (2002), 263–290.
- [47] Sarita Yardi and Amy Bruckman. 2007. What is Computing?: Bridging the Gap Between Teenagers' Perceptions and Graduate Students' Experiences. In *ACM International Computing Education Research (ICER)*. ACM, 39–50.
- [48] L Zimmer and S Bennett. 1987. Gender Differences on the California Statewide Assessment of Attitudes and Achievement in Science. In *American Educational Research Association*.
- [49] Oren Zuckerman, Ina Blau, and Andrés Monroy-Hernández. 2009. Children's Participation Patterns in Online Communities. *Interdisciplinary Journal of E-Learning and Learning Objects* 5, 1 (2009), 263–274.
- [50] Stuart Zweben. 2011. 2009–2010 Taubee Survey Undergraduate CS Degree Production Rises; Doctoral Production Steady. (2011).