Investigating the Role of Being a Mentor as a Way of Increasing Interest in CS

Jody Clarke-Midura
Utah State University
2380 Old Main Hill
Logan, UT
jody.clarke@usu.edu

Vicki Allan
Utah State University
429 Old Main Hill
Logan, UT
vicki.allan@usu.edu

Kevin Close
Utah State University
2380 Old Main Hill
Logan, UT
kevin.close@usu.edu

ABSTRACT

In order to affect career decisions, it is important to reach youth at early ages. While some have focused on using mentors in order to successfully teach mentees, few have focused on the benefits to the mentors themselves. To our knowledge, no other research has been conducted on the effect that serving as a near-peer mentor has on increasing the *mentors'* interest, self-efficacy, value-beliefs, and skills in computer science. Our paid mentorships provided youth two weeks of participation in computer science camps using App Inventor. The mentors in our pilot study increased self-efficacy and interest after the mentoring activity, on average. This all-female mentoring experience provided opportunities to transcend barriers such as negative stereotyping and lack of role models. We feel that being able to reach high school girls at a critical stage makes this pedagogical approach ideal. The positive results, even with the short duration of the intervention, are encouraging.

Keywords

Near Peer Mentoring; App Inventor; Girls; Self-Efficacy

1. INTRODUCTION

Despite effort to broaden participation, the numbers of women and minorities earning bachelor degrees in computer science (CS) is still low [39]. The low numbers of women in CS contradict the trends of higher numbers of women obtaining college degrees and increases in the number of women in other technical fields [10]. Both recruitment and retention of women are of interest, as women tend to leave the major at higher rates than do males. This lack of gender diversity produces unfavorable development solutions for industry and highlights a need for research on how we can increase participation in CS, particularly for women and minorities [27].

Research indicates the importance of reaching youth at early ages, in time to make important decisions about acquiring math and computer skills before college [13][26]. Yet, computer science is not prevalent in K-12 education [5]. To our knowledge, no state requires computer science for graduation, and only half of the states allow computer science to substitute for math requirements [9]. Thus, it is unclear how to reach students when access to computer programming instruction is often non-existent, even in high school [27].

In order to engage middle and high school girls in computing, some researchers have focused on mentors as a way to transcend some of these barriers [2][10]. For example, the National Center for Women & Information Technology (NCWIT) AspireIT program focuses on summer camps and after school programs for middle school girls using near-peer (high school aged) mentors. While these camps serve mentees, in this study, we focused on the mentors. Specifically, we explored the effect of acting as a near-peer mentor on high school girls' self-efficacy, interest, and skill in computer science. Though recent literature highlights the importance of social encouragement and identity-forming activities in girls' computer science lessons, to our knowledge, no other research has been conducted on the effect of acting as a near-peer mentor as a pedagogical approach to increase the *mentors'* interest, self-efficacy, and programming skills in computer science [14][34]. Considering the effectiveness of pair programming among girls in middle school and undergraduate courses and the theoretical underpinnings contributing to the success of pair programming (e.g. increased metacognition, reciprocal teaching, etc.), there is reason to believe that playing a role of near-peer mentor could lead to increased learning and interest in computer science [37]. In this paper, we present results of our pilot study where we studied the effect of being a mentor on high school girl's self-efficacy, interest, and skill in computer science. In the following sections, we briefly provide background and context for our study, our camp, the research design followed by findings and conclusions.

2. PREVIOUS RESEARCH

While there is much discussion about the role of mentors in technology fields, the research focuses on the advantage to the protégées, not on the benefits to the mentors themselves. Goals for mentoring include: providing role models, providing advice, giving support, offering validation, alleviating fears, and addressing isolation [19]. Historically, none of the goals consider the advantages to the mentors. Mentoring is key in recruitment and retention of the protégées in computing [2][10]. Role models, relatives, and the peer group may influence choices [11], with peer support as an especially important factor [20][25][30].

SIGCSE '16, March 02-05, 2016, Memphis, TN, USA

DOI: http://dx.doi.org/10.1145/2839509.2844581

There is a need for research on the mentors themselves and how the role of acting as a mentor may affect knowledge of programming and interest in computer science courses or interest in computer science as a career. The idea of near-peer mentoring has proven successful in increasing the mentors' learning and interest in academics and career in the medical field [31], geoscience [12], as well as in undergraduate science education [17]. Feldon et al [17] found that graduate student teaching experiences improve research skills. Furthermore, Karcher [21] found that high school students mentoring near-peers exhibited increased feelings of school connectedness. Yet, there is a lack of research on the mentors in computer science education. In this research, we build on prior work and study the effect of mentoring on the mentor's computer skills, self-efficacy, and interest in computer science.

When providing guidance, mentors must exercise many essential skills across a diverse set of problems, such as decoding error messages, determining what features to include, and organizing the components. In contrast, learning on one's own provides fewer opportunities for social interaction and justification of decisions, because an individual typically only works on one project at a time. As a mentor of multiple campers, participants in our study engaged with multiple, diverse sets of issues specific to each project's demands, providing a form of varied practice [32]. A further anticipated benefit to mentors is that they need to express their reasoning orally when they provide guidance to campers. Research evidence suggests that when students engage in such self-explanation, they develop more effective problem solving techniques [8][32]. As they explain how to think about a problem, they reinforce their own learning [8]. While the mentors were immersed in coding during the week of mentoring, it was different than doing their own coding therefore we were interested to see if their computational skills improved from mentoring. This benefit to learning occurs as a joint benefit of both practice and self-explanation that occurs during mentoring. Increased skills and motivation has been shown to link to social learning, peer tutoring, and group problem solving. Research has shown that one's perception of their skills is more important to their success than the actual skills they possess [3]. Our goal was to evaluate both the acquisition of skills and perception of skill mastery. The research questions that guided our study were:

- *Does acting as a mentor affect participants' self-efficacy and confidence in their ability to program?*
- *Does acting as a mentor affect participants' interest in computer science careers/academics?*
- *Does acting as a mentor affect participants' programming skill?*

3. METHODS

The purpose of our study was to explore the effect of acting as a near peer mentor on high school girls' programming skills, self-efficacy of their skills, and interest in a technical career. In this section we describe the App Camps and research design.

3.1 App Inventor camp

During the summer of 2015, we held four App Camps; one was focused on preparing mentors to use App Inventor. The other three were camps for middle school students. All of the camps lasted five days, four and a half hours per day. The curriculum was the same in all camps and mentors programmed the same apps as the campers. The curriculum focused on ten apps including: Talk to Me, Paint, Women Speak, Personal Message, Timer, Mole Mash, Chore List, and Walking Tour. The apps focused on important concepts in computing such as Boolean logic, lists, variables, procedures, parameters, conditional statements, looping, incrementing, random selection, use of media files, and event-driven programming. Each camper worked at their own computer and had their own device for testing the apps. Campers and mentors were allowed to take the devices home to show their friends and family what they created.

In addition to the ten cell phone apps that each camper created, we played games to learn terminology. Campers had a chance to step away from their computers, get exercise, and socialize while pantomiming and blurting out computer terms. We also offered career coaching, in the form of a visit from the dean of engineering for each camp, where she led a discussion of job prospects for computer scientists. We also showed a video made by Code.org to show how exciting jobs in computer science could be. We emphasized the ability of programmers to develop apps that help people. We also utilized games to create a relaxed attitude among the mentors and campers.

3.2 Mentor training

In addition to the week-long pre camp training, the mentors were involved in an hour and a half of mentor training each day after the camp. This time was used to discuss learning styles, allow mentors to create blog entries discussing specific successes and challenges, and work on programming challenges. The non-technical part of mentor training gave the mentors ownership of the camp as they could suggest improvements to the organization. It was also a time for reflection as we talked about what went well and why.

The duration of the intervention for each mentor was short – one week for training and one week for mentoring. While we expect that the improvements would be more dramatic in a longer intervention, it is noteworthy that others have found significant changes are possible in a short period of time [18]. This is important as, during the summer, it is challenging to find mentors with long periods of availability or the resources to conduct long interventions.

3.3 Sample

Our sample consisted of twenty-six high school female mentors from four different high schools in northern Utah. The racial makeup of our sample was African-American (8%), Asian/Pacific Islander (15%), Latina (4%), and Caucasian (73%). The average age of the mentors was 16.17 years. The mentors were required to be entering ninth through twelfth grades, but because of the large number of applications, we gave first preference to older mentors who would have fewer opportunities to participate in such a program.

Twenty-five of the mentors have computers at home and access to the Internet. Twenty-one of the mentors have taken a Computer Education and Technology (CET) class in their school. The CET course is a required course intended for students to learn concepts associated with key application software, basic computing fundamentals, and appropriate behavior while using technology as a tool in the classroom and in life. One mentor had experience with the Scratch programming language through a local workshop.

3.4 Procedures

Mentors completed an application and indicated which week they could mentor. Based on their availability, we randomly divided the mentors into three groups: A, B, and C. All three combined groups attended the week-long App Training camp on a college campus. Prior to the camp, all participants took pre surveys in which they were asked about their computer self-efficacy and interest (described in the next section). At the end of the camp, all participants took identical post surveys on their self-efficacy and interest and a programming skills test. The following week, Group A mentored an all girl camp. They received mentor training immediately following each day of camp for four days. At the end of the first camp, Group A took identical delayed post surveys with the skills test. Two weeks later, group B took the surveys and then mentored a mixed gender camp. They received mentor training following each camp session for four days. At the end of the camp, group B took the delayed post surveys. The following week group C mentored a group of all girl campers. They received mentor training after each camp session. After the camp, they took the delayed post surveys.

3.5 Measures

3.5.1 Affective measures

In order to assess whether acting as a mentor had an effect on their self-efficacy and interest in computer science, we piloted an affective survey that was developed from three instruments: The Social Cognitive Career survey [7], Motivational Beliefs and Assessment survey [33], and Student Perceptions of ICT survey [24]. Our instrument contained 41 items and asked students to rate the items on a scale of 1-6 from strongly disagree to strongly agree. One item was eliminated because it was worded awkwardly and did not fit in the initial factors. To aid the interpretation, we conducted a commonly used exploratory factor analysis technique [1], orthogonal varimax rotation, in SPSS. Varimax clarifies each factor and set of variables by enhancing the load of variables on a single factor and reducing their load on the remaining set [29]. Three factors explained 53.234% of the variance: self-efficacy, interest, and value-beliefs. Self-efficacy contains items like, “I can succeed in a computer science class” and “I am confident in my ability to program.” Interest contains items like, “I am interested in studying computer science in the future” and “Working in the computer science is an interesting occupation.” Value-belief contains items related to personal and social beliefs: “It is important for me to improve my computer programming skills” and “It is useful to have programming skills” as well as supports: “My parents think that being good at computer programming is useful for my future.”

Composite scores were created for each of the three factors, based on the mean of the items, which had their primary loadings on each factor. Higher scores indicated greater self-efficacy, interest, and value-beliefs. Internal consistency for each of the scales was examined using Cronbach’s alpha, a widely used measure of scale reliability [15]. The alphas were high: interest $\alpha = 0.9566$ (19 items), self-efficacy $\alpha = 0.8769$ (11 items), and value-belief $\alpha = 0.7933$ (10 items). Descriptive statistics are presented in table 1.

Overall, these analyses indicated that three distinct factors were underlying mentors responses and that these factors were internally consistent.

Table 1. Descriptive Statistics of Affective Factors

Graphics	Items	Mean (SD)	Alpha
Interest	19	4.61 (.87)	.96
Self-efficacy	11	5.21 (.56)	.88
Value-belief	10	5.13 (.48)	.80

3.5.2 Programming skills

In order to assess whether acting as a mentor affected mentor’s programming skills, we administered a 25 item, multiple-choice test modified from the Evaluation for App Inventor test developed by Erickson [16], before they mentored (SKILL 1) and then again after they mentored (SKILL 2).

The questions assessed a variety of skills: basic understanding of App Inventor terminology, basic programming skills, and detailed understanding of programming logic. Sample skill questions were “Our variables typically had a name beginning with "global". What does that imply?” “The Loop property of the player tells the player to do what?” and “In the following piece of code, what is the value of global total when global meal is 20 and global tipPicker is 15?”

4. RESULTS

Our first two research questions focused on whether high school students’ self-efficacy and interest changed after mentoring. As mentioned above, we surveyed mentors on 41 items that fit into 3 factors: self-efficacy, interest, and value-belief. We conducted a one way repeated measures ANOVA to compare the effect of time on each of these three factors before App Training (T1), Before Mentoring (T2), and after mentoring (T3). We then conducted post hoc comparisons between the conditions (time1-time2, time2-time3 and time1-time3) to look for significant differences. Descriptive statistics for each of the three factors are presented in Tables 2, 3, and 4.

Table 2. Descriptive Statistics of Interest (N=25)

	Mean	Std Dev	Std Error	Median	Range
Pre-App Camp (T1)	4.59	0.67	0.13	4.47	2.21
Pre-Mentoring (T2)	4.51	1.01	0.20	4.82	3.68
Post-Mentoring (T3)	4.71	0.95	0.19	4.84	2.89

Table 3. Descriptive Statistics of Self-Efficacy (N=25)

	Mean	Std Dev	Std Error	Median	Range
Pre-App Camp (T1)	5.01	1	0.12	5	2.64
Pre-Mentoring (T2)	5.22	0.52	0.10	5.27	1.82
Post Mentoring (T3)	5.38	0.52	0.10	5.55	1.91

Table 4. Descriptive Statistics of Value-beliefs (N=25)

	Mean	Std Dev	Std Error	Median	Range
Pre-App Camp (T1)	5.14	0.47	0.09	5.1	1.70
Pre-Mentoring (T2)	5.07	0.43	0.09	5.1	1.80
Post-Mentoring (T3)	5.15	0.53	0.11	5.2	2.00

As you can see in Table 3, the mentors in our sample reported an increase in self-efficacy at each time interval, on average. As shown in Table 2 and Table 4, there was a slight decrease, on average, for interest and value-beliefs from T1 to T2. However, this decrease was not significant. Given the typical decrease in interest experienced by many new programmers, we appear to be experiencing an initial loss in interest that we hypothesize mentoring is able to overcome.

4.1 Interest

We measured mentors' interest in Computer Science three times. As mentioned above, the mentors in our sample reported a slight decrease in interest from Pre-App Camp (T1) to Pre-Mentoring (T2). Not surprisingly, when we conducted a one-way repeated measures ANOVA, we did not find a significant effect of time on interest, *Wilks' Lambda = .860, F(2, 23) = 1.866, p=.177.* However, we did find a significant difference in interest when we made post hoc comparisons between the conditions.

The data did not meet all assumptions for normality, so we conducted Wilcoxon Signed-ranks tests to make comparisons. The tests indicated that mentors' interest was higher after mentoring (time 3, Median = 4.84) than before mentoring (time 2, Median = 4.82), *Z = 1.941, p< .05, r =.3882*. On average, the girls in our sample, who acted as mentors for their middle school peers at a camp on how to program apps using App inventor, saw a statistically significant increase in their interest in computer science from before they mentored to after they mentored.

Interviews and surveys hinted at reasons for the increase in interest. Mentors remarked, "It was really interesting to see all the things I could do and how all the things fit together." "I was surprised that programmers work with a bunch of people. The idea of getting a team together to work on things, that was new to me." "I enjoyed working with the campers and helping them solve their problems. It made me feel very smart." "I am definitely more likely to follow a career in computing, because now I understand what it would be like more." "Before I thought only super smart people were in computing degrees. I really enjoy working with computers, after actually being able to see myself progress and be able to understand things I never thought I would!"

4.2 Self-Efficacy

We conducted a repeated measures ANOVA to look for an effect of time on self-efficacy. Mauchly's test indicated that the assumption of sphericity had been violated, $\chi^2(2)$ = 14.252, p= .001. Therefore, degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity (ε = .684). The results show that there was a significant effect of time on mentors self-efficacy, *F(1.368, 32.835) = 7.677, p= .005*. These results suggest that mentors self-efficacy increased over time after mentoring their younger peers.

There was a significant difference in self-efficacy between all occasions. A Wilcoxon Signed-ranks test indicated that mentor self-efficacy was higher after the initial App Inventor training (time 2, Median = 5.27) than before App Training (time 1, Median = 5), *Z = 2.062, p< .0392, r =.4124.* Mentor self-efficacy was higher after mentoring (time 3, Median = 5.54) than before mentoring (time 2, Median = 5.27), *Z = 2.538, p< .0112, r =.5076*. Finally, mentor self-efficacy was higher after mentoring (time 3, Median = 5.54) than prior to their App Camp training (time 1, Median = 5), *Z = 2.951, p< .0032, r =.5902*. The girls in our sample saw an increase in their self-efficacy after being a near-peer mentor. Their self-efficacy increased over time.

4.3 Value-beliefs

We conducted a one-way ANOVA with repeated measures followed by post hoc comparisons of mentors' value beliefs on the three occasions. We did not find a significant effect of time on value beliefs or significant differences between the three occasions. Thus, even though there was a slight decrease in value beliefs from Pre-App Camp (T1) to Pre-Mentoring (T2) and then an increase after mentoring, none of these differences were statistically significant. These results suggest a need for further research into the development and growth of value beliefs.

4.4 Programming Skills

Our third research question explores whether acting as a mentor affects participants' programming skills. The mean, standard deviation, and standard error of SKILL 1 and SKILL 2 are presented in Table 5. Our data met assumptions of normality. In order to see if there was a difference between mentors programming skills before and after mentoring, we conducted a paired t-test between the mean of SKILL 1 and SKILL 2. While participants scored higher on SKILL 2 on average (mean = 17.95833) than SKILL 1 (mean = 17.33333); the difference was not significant, $t(23) = 1.4288$, $p = > 0$.

Table 5. Descriptive Statistics of Programming Skill (n=24)

	Mean	**Std Deviation**	**Std Error**
SKILL 1 (Pre-Mentoring)	17.33	3.73	.75
SKILL 2 (Post-Mentoring)	17.96	4.46	.91

5. DISCUSSION

We set out to explore whether the role of being a near-peer mentor to middle school peers on App Inventor has any positive effect on high school girls' self-efficacy, interest, and skill in computer programming. In our pilot, we found a relationship between the role of being a near-peer mentor and participants' self-efficacy and interest, on average. While we can't make any causal claims, the results of our pilot are exciting and have broader implications for new research in the field.

5.1 Changes in affect

The girls in our sample, on average, saw a significant increase in their self-efficacy over time. There was also a significant increase in their interest in computer science from before they mentored to after they mentored, on average. The significant increase in self-efficacy over all three occasions is promising because it suggests that short interventions of one week have the potential to have positive effects. This also aligns with research that has shown that mentoring improves academic self-esteem [21]. Similarly, Scott et al [28] found that their hands-on, female-centered information technology camp also increased the students' sense of self as a technosocial manipulator. Finally, other social pedagogies for programming have also shown increased self-efficacy and interest among girls, particularly pair programming [23][37].

Retention of women is an important issue in computer science. Despite efforts to broaden participation, retention rates of women remain low [2][4][10]. It was not surprising that, on average, girls' interest went down after the first App training. However, after mentoring, the girls in our sample reported a statistically significant increase in interest. These findings suggest that placing girls in the role of mentor has the potential to increase retention in CS. Further research is needed to understand how mentoring increases interest in CS and how we can use mentoring as a pedagogical approach to broaden participation in CS.

We did not see a similar change in value-belief that we saw in self-efficacy or interest. We believe there is a reason for this. First, we didn't emphasize the career aspects or value of programming for a wide variety of careers. We have already started conversations with some industry partners and plan to develop videos of women talking about their CS careers. Our future camps will not only place more emphasis on the variety of options a CS career offers, but we also hope to develop career pathways for the mentors through partnerships with industry and the potential of internships. We hope that this will lead to increased value-belief.

5.2 Changes in skill

Assessing programming skills remains a challenge [35]. We struggled to find a good measure that we could use to show change in skill after mentoring. The multiple-choice questions we administered, while useful, were insufficient to understand the learning that can happen in such a short intervention. However, for the purpose of our pilot research, our findings confirm what we observed in each of the camps: the mentors were learning programming skills and we need to find a way to measure this. Researchers are currently looking for ways to assess programming and computational thinking [36][38], and we hope to utilize some of these resources in our future studies.

There is a critical need to find innovative pedagogies that broaden participation in computer science. Our findings are promising because they suggest that the role of acting as a mentor has the potential to increase girls' self-efficacy and interest in computer science and programming. The camaraderie that developed with an all-female group of mentors was key in helping them find role models and friends in the group and allowing them to crack the stereotype of computer programmers. The negative stereotype of computer scientists was further weakened by the admiration of the campers. Not only were the mentors esteemed as role models in fashion, humor, and accomplishments, they were appreciated in their role as technology experts.

6. FUTURE WORK

Our pilot results were promising and encourage us to continue pursuing research on how mentoring affects the mentor's self-efficacy, interest, value beliefs and skills in CS. In future camps, we will place more of an emphasis on the social encouragement that occurs [34]. Cell phone apps are compelling as they utilize pictures, music, text, vibration, and motion. In addition to sparking creativity, the use of media allows for a discussion of file types, pixels, screen coordinate, color, and file manipulation. Apps have the added advantage of being shareable as students take the phones home to show family, friends, and neighbors. This utilization of audience is important in changing attitudes of members of the mentor's support groups. [22]. While campers did take the devices home to show off what they created, mentors were less likely to do the same. As part of the mentor training, we will encourage them to share their apps.. We intend to use observation and interviews to further understand the mechanisms that lead to change in mentors' affect and skill development. We also plan to study the effects of mentoring for a longer period (with two separate camps) to see if the repetition further improves skill development.

We saw a change in interest after mentoring, but that is largely unexplored. What about mentoring increases interest? Certainly the skill acquisition and self-efficacy makes increased interest possible. Other reasons could include: knowledge of job opportunities, knowledge of personal skills, understanding what computer science really is, understanding of the work environment, breaking down of stereotypes, and the presence of role models (in other mentors and faculty). We intend to study the factors that influence change in mentors' attitudes and skills.

7. CONCLUSION

Finding ways to increase girls' participation in computer science is a national priority. A recent study on what factors influence women to pursue CS careers found social encouragement, academic exposure, career perceptions, and self-perception to be the most critical [34]. In this paper, we share a promising pedagogical approach: placing high school girls in the role of mentor to their middle school peers, that has the potential to increase participation in CS while encompassing these four factors.

By offering paid mentorships, we were able to attract young women to a computing experience who are too old for the typical camp experience, yet closer to a career decision. Using near-peer mentors was key to making the camp successful for the campers, but it is equally important to note that by providing a mentoring experience, we were able to attract young women to a computing experience who have few opportunities for such experiences. We feel that being able to reach high school girls at a critical stage make this pedagogical approach ideal.

8. REFERENCES

[1] Abdi, H. 2003. Factor Rotations in Factor Analyses. In M. Lewis-Beck, A. Bryman, & T. Futing (Eds.), *Encyclopedia of Social Sciences Research Methods*. Thousand Oaks, CA: Sage.

[2] Ashcraft, C., Eger, E., & Friend, M. 2012. Girls in IT: the facts. *National Center for Women & IT. Boulder, CO*.

[3] Bandura, A. 1997. Self-efficacy: The exercise of self-control. Gordonsville, VA: WH Freeman & Co.

[4] Bunderson, E. D., & Christensen, M. E. 1995. An analysis of retention problems for female students in university computer science programs. Journal of Research on Computing in Education, 28(1), 1-18.

[5] Barr, V., & Stephenson, C. 2011. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. ACM Inroads, 2(1), 48-54.

[6] Brunner, C., Bennett, D., & Honey, M. 2000. Girl games and technological desire. From Barbie to Mortal Kombat: Gender and Computer Games, ed.

[7] Carrico, C., & Tendhar, C. 2012. The Use of the Social Cognitive Career Theory to Predict Engineering Students' Motivation in the PRODUCED Program. In *American Society for Engineering Education*. American Society for Engineering Education.

[8] Chi, M. T., Leeuw, N., Chiu, M. H., & LaVancher, C. 1994. Eliciting self-explanations improves understanding. Cognitive science, 18(3), 439-477.

[9] Code.org Website

[10] Cohoon, J. M. 2011. Promising Practices: Encouragement Works in Academic Settings (Case Study 1). Boulder, CO: National Center for Women & Information Technology.

[11] Cozza, M. 2011. Bridging Gender Gaps, Networking in Computer Science.Gender, Technology and Development, 15(2), 319-337.

[12] DeFelice, A., Adams, J. D., Branco, B., & Pieroni, P. 2014. Engaging underrepresented high school students in an urban environmental and geoscience place-based curriculum. Journal of Geoscience Education, *62*(1), 49-60.

[13] Denner, J. 2011. What Predicts Middle School Girls' Interest in Computing? International Journal of Gender, Science and Technology, *3*(1).

[14] Denner, J., Werner, L., Bean, S., & Campe, S. 2005. The Girls Creating Games Program. Frontiers: A Journal Of Women Studies, 26(1), 90-98.

[15] DeVellis, R. 2003. Scale Development Theory and Applications. Thousand Oaks, CA: SAGE Publications

[16] Erickson, B. 2015. (retrieved http://home.cc.gatech.edu/ice-gt/311).

[17] Feldon, D. F., Peugh, J., Timmerman, B. E., Maher, M. A., Hurst, M., Strickland, D., & Stiegelmeyer, C. 2011. Graduate students' teaching experiences improve their methodological research skills. Science, 333(6045), 1037-1039.

[18] Hulleman, C. S., & Harackiewicz, J. M. (2009). Promoting interest and performance in high school science classes. *Science*, *326*(5958), 1410-1412.

[19] Hupert, N. 2002. Telementoring: Using Online communication for a student mentoring project, http://cct.edc.org/publications/telementoring-using-online-communication-student-mentoring-project, accessed 1/22/2015.

[20] Jenson, J., De Castell, S., & Bryson, M. 2003. "Girl talk": gender, equity, and identity discourses in a school-based computer culture. In Women's Studies International Forum (Vol. 26, No. 6, pp. 561-573). Pergamon.

[21] Karcher, M. 2009. Increases in academic connectedness and self-esteem among high school students who serve as cross-age peer mentors. Professional School Counseling, 12(4), 292-299.

[22] Marcu, G., Kaufman, S. J., Lee, J. K., Black, R. W., Dourish, P., Hayes, G. R., & Richardson, D. J. 2010, Design and evaluation of a computer science and engineering course for middle school girls. In Proceedings of the 41st ACM technical symposium on Computer science education (pp. 234-238). ACM.

[23] McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. 2003, May. The impact of pair programming on student performance, perception and persistence. In Proceedings of the 25th international conference on Software engineering (pp. 602-607). IEEE Computer Society.

[24] McLachlan, C., Craig, A., & Coldwell, J. 2010, January. Student perceptions of ICT: a gendered analysis. In Proceedings of the Twelfth Australasian Conference on Computing Education-Volume 103 (pp. 127-136). Australian Computer Society, Inc.

[25] Moorman, P., & Johnson, E. 2003, June. Still a stranger here: Attitudes among secondary school students towards computer science. In ACM SIGCSE Bulletin (Vol. 35, No. 3, pp. 193-197). ACM.

[26] Powell, R. M. 2008, March. Improving the persistence of first-year undergraduate women in computer science. In ACM SIGCSE Bulletin (Vol. 40, No. 1, pp. 518-522). ACM.

[27] Scott, K. 2009. The new digital divide: Where are our girls?. LeadCast Blog, http://www.niusileadscape.org/bl/?p=404.

[28] Scott, K., Aist, G. & Zhang, X. 2014. Designing a Culturally Responsive Computing Curriculum For Girls. International Journal of Gender, Science and Technology,6(2), 264-276.

[29] Stevens, J. 2002. *Applied Multivariate Statistics for the Social Sciences* (4th ed.). Mahwah, NJ: Lawrence Erlbaum Associates.

[30] Teague, J. 2002. Women in computing: What brings them to it, what keeps them in it? ACM SIGCSE Bulletin, 34(2), 147-158.

[31] Tenenbaum, L. S., Anderson, M. K., Jett, M., & Yourick, D. L. 2014. An Innovative Near-Peer Mentoring Model for Undergraduate and Secondary Students: STEM Focus. Innovative Higher Education, 39(5), 375-385.

[32] VanLehn, K. 1996. Cognitive skill acquisition. Annual review of psychology, 47(1), 513-539.

[33] Vekiri, I., & Chronaki, A. 2008. Gender issues in technology use: Perceived social support, computer self-efficacy and value-beliefs, and computer use beyond school. Computers & Education, 51(3), 1392-1404.

[34] Wang, J., Hong, H., Ravitz, J., & Ivory, M. 2015. Gender Differences in Factors Influencing Pursuit of Computer Science and Related Fields. In Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (pp. 117-122). ACM.

[35] Weintrop, D., & Wilensky, U. 2015. Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. In Proceedings of the eleventh annual International Conference on International Computing Education Research (pp. 101-110). ACM.

[36] Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. 2012. The fairy performance assessment: Measuring computational thinking in middle school. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 215-220). ACM.

[37] Werner L., Denner J. 2009 Pair programming in middle school: What does it look like? Journal of Research on Technology in Education (ISTE), 42(1), 29-49.

[38] Werner, L., McDowell, C., & Denner, J. 2013. Middle school students using Alice: what can we learn from logging data?. In Proceeding of the 44th ACM technical symposium on Computer science education (pp. 507-512). ACM.

[39] Zweben, S., & Bizot, B. 2014. 2013 Taulbee Survey. COMPUTING, 26(5).