

# Using Informed Design in Informal Computer Science Programs to Increase Youths' Interest, Self-efficacy, and Perceptions of Parental Support

JODY CLARKE-MIDURA, CHONGNING SUN, KATARINA PANTIC,  
FREDERICK POOLE, and VICKI ALLAN, Utah State University

---

Our work is situated in research on Computer Science (CS) learning in informal learning environments and literature on the factors that influence girls to enter CS. In this article, we outline design choices around the creation of a summer programming camp for middle school youth. In addition, we describe a near-peer mentoring model we used that was influenced by Bandura's self-efficacy theory. The purpose of this article, apart from promoting transparency of program design, was to evaluate the effectiveness of our camp design in terms of increasing youths' interest, self-efficacy beliefs, and perceptions of parental support. We found significant gains for all three of these concepts. Additionally, we make connections between our design choices (e.g., videos, peer support, mentor support) and the affective gains by thematically analyzing interview data concerning the outcomes found in our camps.

CCS Concepts: • **Social and professional topics** → **Computing education; Computing education programs; Computer science education; Informal education; K-12 education;**

Additional Key Words and Phrases: Informal CS learning environments, near-peer mentors, block-based programming

## ACM Reference format:

Jody Clarke-Midura, Chongning Sun, Katarina Pantic, Frederick Poole, and Vicki Allan. 2019. Using Informed Design in Informal Computer Science Programs to Increase Youths' Interest, Self-efficacy, and Perceptions of Parental Support. *ACM Trans. Comput. Educ.* 19, 4, Article 37 (July 2019), 24 pages.  
<https://doi.org/10.1145/3319445>

---

## 1 INTRODUCTION

Although the overall percentage of K-12 schools that offer at least one computer science (CS) class has increased in the past few years, only 40% of schools reported offering at least one CS course in 2016 [33]. While there is strong parental and teacher support to integrate CS into K-12, there are barriers at the school level. Some of these barriers include a scarcity of instructional time, insufficient funds to pay CS teachers, standardized testing systems that do not assess CS skills, and a lack of need seen by administrators [33]. Perhaps in response to both the plea by parents and educators for CS opportunities at the K-12 level, and the challenges for implementing such

---

This work was supported by an ITEST Grant No. 1614849 from the National Science Foundation.

Authors' addresses: J. Clarke-Midura, C. Sun, K. Pantic, and F. Poole, 2830 Old Main Hill, Logan, UT 84322; Vicki Allan, 4205 Old Main Hill, Logan, UT 84322.

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 ACM 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](mailto:permissions@acm.org)

© 2019 Association for Computing Machinery.

1946-6226/2019/07-ART37 \$15.00

<https://doi.org/10.1145/3319445>

programs in K-12 classrooms, people are turning to programs that offer CS experiences in informal learning environments.

The number of informal CS programs for K-12 is encouraging. While it is important that these programs continue to grow, and expand, it is also helpful for educators, researchers, and other stakeholders to learn from the successes and failures that occur in these informal learning environments. To accomplish this, we recommend that researchers try to make connections between affective outcomes, learning outcomes, and design choices. This requires sharing detailed descriptions of program designs and using rich data and analyses to explain and illustrate what is happening in these programs.

With these issues in mind, we situate the design and outcomes of a summer App Camp we ran in the larger body of research on CS learning in informal learning environments that is being shared and reported within the CS Education community. The purpose of this study is to evaluate the effectiveness of our App Camp design in terms of increasing interest (i.e., career interest and intrinsic interest) in CS, beliefs about ability to program (i.e., self-efficacy), and perceptions of parental support for pursuing CS studies. The research questions that guide this study are: (1) How does our App Camp affect campers' interest in CS? (2) How does our App Camp affect their self-efficacy? and (3) How does our App Camp affect campers' perceptions of parental support systems?

In the following sections, we first provide an overview of the research trends and strategies that are employed in the aforementioned informal CS learning environments. Then, we present an in-depth description of our App Camp. In addition, we draw connections between our App Camp design choices and our hypothesized outcomes. Finally, in the results section, we use qualitative data to illustrate the affective changes we saw in our camps.

## 2 RELATED WORK

### 2.1 Research on CS Camps and Informal Learning Environments

To explore the effectiveness of existing CS informal learning programs, we conducted a literature search in several databases common to the CS community: the ACM Digital Library, cse-research.org, IEEE, ICLS, and Computers & Education (see Appendix C for a full list of articles identified in this search). The following search terms were used: computer science and afterschool, summer camp, summer program, summer course, summer project, outreach program, and computational thinking. The search resulted in 92 articles. After filtering out articles that were not investigating middle/high school students, which is the age group we were focusing on, and those that did not focus on learning CS in an informal environment, 31 articles were identified.

Our review of literature revealed numerous informal learning programs for students between the ages of 11 and 16 years in the form of summer camps [2, 15, 50], afterschool programs [23], workshops [73], and seminars [34]. Most programs were implemented over a period of one to four weeks [51], though some were longer than four weeks [57] and some were less than a week [52] long. The programs were diverse in their approaches to introducing CS to youth who previously had little or no computing experience. For example, many offered block programming languages such as Scratch [4, 29], Alice [1, 75], and App Inventor [15, 61, 62], while others offered scripting languages, such as Python, JavaScript, and HTML [57, 70, 72]. A common goal in these programs was to broaden participation in CS with some programs focusing specifically on girls [37, 43]. These informal learning programs used a variety of strategies and approaches to increase interest in programming [37], often including an attempt to address the four factors researchers claim broaden participation in CS: improving self-efficacy, providing social support, exposure to the field, and offering opportunities to program [74].

Of the 31 articles identified, 24 assessed either participants' career interest or intrinsic interest in CS with only six of the studies reporting significant gains in one or both of the two interest

constructs [15, 17, 22, 23, 70, 73]. Eleven studies [3, 20, 29, 34, 43–45, 50, 51, 57, 61] used qualitative and/or descriptive analysis and seven studies [1, 11, 61, 62, 64, 69, 75] reported non-significant gains in interest. Although success was reported, we argue that there is still a need to further investigate best practices, as well as how they impact participants' interest in, and subsequently, future participation in CS. With this goal in mind, in the next section, we provide an overview of literature on the four factors [74] to influence CS participation and illustrate how they guided the design of our app camps.

## 2.2 Factors of Interest in CS Informal Learning Environments

Of the four factors, opportunities to program is perhaps the easiest to accomplish when designing a program. All programs that provide a space and equipment are able to provide their participants with opportunities to program, which was precisely what we did. However, the ways in which programs promote self-efficacy, provide social support, and exposure to the CS field vary. In this section, we summarize some of the approaches used in other programs to address these three factors and make the connection with our own design choices.

**2.2.1 Self-Efficacy.** Given that many informal learning programs aim to broaden participation in CS, it is not surprising that such programs often assessed self-efficacy gains among their campers [4, 15, 23, 30, 51, 62, 65]. According to social cognitive career theory (SCCT) [46], high self-efficacy beliefs positively influence one's outcome expectations and interest in a particular field or career. It would be beneficial for the field to identify best practices or features that lead to the growth or development of self-efficacy across some of these studies. For example, in a summer app camp using App Inventor to design apps to address community needs, Ni, Schilder, Sherman, and Martin [51] found that students reported significant gains in self-efficacy. Similarly, Aritajati et al. [4] reported the findings of four summer camps for youth where they found significant but small increases in self-efficacy. The authors attributed their gains in self-efficacy to "effective topics, activities, and instructors" (p. 585) without expanding on what made them effective. Some studies did provide in-depth descriptions of their programs but did not make connections between program design and outcomes [22, 61]. In one study that did make a connection, Sabin, Deloge, Smith, and DuBow [62] argued that the significant finding in self-efficacy was the result of having a final project (the app) that participants could share with friends and family. This finding aligned with our own anecdotal experience in working with programming camps. Therefore, one of our design choices was to allow and encourage participants to take their android devices home, so that they could share their apps with friends and family. To further contribute to literature on what leads to self-efficacy gains in programming, in this article, we highlight all the design features that, based on campers' testimonies, led to self-efficacy gains of our participants.

**2.2.2 Social Support.** Previous research attempted to account for the importance of learning programming in a social context [26, 55]. To that end, many online programming communities, such as Scratch, are already designed to provide social support [41] inside the context itself. Additionally, research interventions are frequently designed to include features that provide programming support in face-to-face interventions. These types of support range from providing campers with access to CS experts [62], mentors [30, 47], peer support [55], and/or role models [34]. For example, when situating campers' learning among their peers, by encouraging peer-to-peer interactions and collaborations, Pantic et al. [55] found peer support to be an encouraging characteristic in a Scratch camp. In an earlier study, we found that relatable near-peer mentors influenced a positive change in campers' interest and self-efficacy [15]. For those reasons, our camp was designed to both provide mentor support and encourage peer interactions inside the camp. For more details on the mentoring model, see Section 2.3 (Theoretical Framework).

In addition, external support (e.g., parents, friends, and other family members) are also important. Parental support has been found to be a strong predictor of career interest and aspirations for youth [14, 39, 53, 54, 68]. It has been significantly associated with higher utility value beliefs, an increase in computing interest [21] and general career decidedness and career self-efficacy, which subsequently increased persistence in the area of study [59]. Studies looking into the differences between mother and father support are rare and showed mixed results [14, 31, 63]. While many studies do address in-camp learning support or parental support in general, only a few studies acknowledge and/or assess the impact of informal learning programs on learner perceptions of support outside of the learning environment [14, 62, 69]. This is important, because participation in informal learning programs is often the first CS encounter that many learners experience. Designing curriculum features that provide an opportunity for learners to express or share their interest with those outside of the learning environment may prove valuable in establishing a dialogue around CS and subsequently increasing the participants' awareness of the social support and encouragement available in their immediate social environments. To that end, and as mentioned above, we encouraged campers to take their Android devices home during the camp and show their artifacts to their parents, siblings, and friends. In this article, we describe the types of interactions (aka parental support) that were generated when learners attempted to share their artifacts with parents during camp.

**2.2.3 Exposure to the CS Field.** In addition to designing for self-efficacy and social support, research has also supported the need for exposing learners to the CS field. Exposure to the field is a positive way to address negative stereotypes that are often held about computer scientists. In addition, it also fuels interest in CS careers by illustrating possible futures [74]. To provide learners with these experiences, informal learning programs have achieved success using field trips [75], guest speakers [18, 75], lectures/sessions about CS careers [11] and commercially developed videos (e.g., Code.org) introducing CS careers [18]. For example, Craig and Horton [18] invited female CS professionals from all career stages to give talks on the source of their passion for their jobs and the pathways to their current positions. Additionally, they showed videos on the variety of available CS careers. Similarly, DeWitt et al. [22] introduced campers to CS faculty and students to provide them with an opportunity to “chat with a computer scientist” (p. 162). The guest speakers talked about their research projects and why they chose to study and or teach CS. Khoja, Wainwright, Brosing, and Barlow [43] organized three field trips to local technology companies (e.g., Intel) to expose students to a variety of role models and to let them see CS in action. This research influenced the design of our camp in two ways. First, we incorporated a presentation on CS careers by a Computer Science professor, and we showed a video created by Code.org [16]. Second, we developed our own videos of young adults who come from similar backgrounds as our campers and decided to pursue CS as a career. We describe these videos in more detail in Section 2.5.

In the following section, we discuss the theoretical principles associated with our near-peer mentoring model and how the near-peer mentoring model that we used addresses each of the aforementioned factors that influenced our design.

### 2.3 Theoretical Framework

Another design feature of our camp was an implementation of a near-peer mentoring model. Research in CS reveals a variety of studies that used mentoring as a pedagogical strategy for enhancing CS participation [9, 10, 24, 40, 42, 66]. Most of these studies, however, are not designed around the proximity of age and expertise dyad, which is the basis of our model.

Our mentoring model was influenced by Bandura's self-efficacy theory [5, 6, 8]. This theory posits that perceived self-efficacy is a major determinant of choice of activities, effort expenditure

on a task, and persistence in the task in the face of obstacles. While not the sole determinant of behavior, self-efficacy plays a major role in people's choices of how much effort they will expend and how long they will sustain effort when dealing with stressful situations. Similarly, social cognitive career theory (SCCT) [46], a theory built upon Bandura's social cognitive theory [7] to explain career and academic interest development, states that self-efficacy affects career development via its influence on outcome expectations, interests, choice goals, and choice actions [46]. In other words, self-efficacy was found to positively influence one's interest in a field or career.

Bandura [5, 7, 8] hypothesized that four sources influence self-efficacy: enactive experience (i.e., success and failures), vicarious experience (i.e., observing others perform a task and its outcomes), social persuasion (i.e., encouragement or discouragement), and physiological state (i.e., anxiety or fear). Our App Camp was designed with these principles in mind. As an example, in our model, near-peers, who are slightly more experienced in programming and a few years older than the campers, provide instructional support to youth learning to program. By using near-peer mentors, we argue that such mentors can improve the feeling of success (enactive experience) by providing feedback, function as role models (vicarious experience), and provide encouragement embedded in feedback (social persuasion). Finally, given the near-peers proximity in age and expertise to their mentees, we argue that compared to a traditional instructor (a professor or teacher) near-peers may provide (1) an enhanced learning experience and (2) a less stress-inducing environment for learning to code. In the following section, we will provide more information about our camps, its structure (e.g., the mentoring model we used), and the curriculum.

## 2.4 App Camp Design

During the summer of 2017, we offered five camps to middle and high school students. In the U.S., middle school students can range in age between 10 and 14 years old. High school students typically range between 14 and 18 years old. The first of our five camps specifically focused on mentor training and programming for high school youth, while the other four camps were geared toward teaching middle school youth how to program. Both age groups created apps using MIT's App Inventor. The camps for middle school-aged youth cost \$35 for early bird registration and then \$40 per child the week before the camp. Campers who applied for a "need-based scholarship" did not have to pay for the camp.

The camps used a near-peer mentoring model that consisted of lead mentors, mentors, and campers. Several PhD student researchers and a CS professor were also nearby as needed. In this section, we discuss the format and structure of the App Camp design including the near-peer mentoring model and the curriculum.

**2.4.1 Lead Mentors.** Lead mentors were high school students who received training and served as mentors in our camps the previous summer. We hired five female lead mentors (average age = 16) to help us run mentor training. Their racial/ethnic make-up consisted of three Caucasians, one Latina, and one multiracial. Two of them reported being on free or reduced lunch. In addition to their prior experience with programming and mentoring, lead mentors attended a one-day lead mentor training event.

**2.4.2 Mentors.** In 2017, we hired 25 mentors (6 males and 19 females, average age = 16). Mentor ethnicities included Asian/Pacific Islander (16%), Native American (4%), Latina/o (4%), and Caucasian (76%). Thirty-two percent of the mentors reported being on free or reduced lunch. All mentors participated in a five-day training (30h), where they learned how to program apps using App Inventor and how to be a role model/mentor for the campers. Programming included debugging activities in which they learned to troubleshoot broken programs that simulated problems campers



may encounter in practice. By the end of the week, they were expected to create their own apps without instructions.

Mentors were also exposed to three types of activities: social activities with their lead mentors, modeled behavior from their lead mentors, and mentor training activities. Each day started with social activities (e.g., Scavenger hunt, icebreakers). The goal of the social activities was to emphasize the importance of putting in the effort to get to know your mentees and help them feel more comfortable around the mentors. In the two weeks immediately following mentor training, mentors helped us run programming camps for middle school youth, providing content and emotional support and encouragement.

Mentors applied to the position by filling out an online questionnaire about their experience working with youth and availability. Prior experience programming or using App Inventor was not a requirement. Mentors were randomly assigned to one of the four camps based on their availability. Two of the camps had all female mentors and two had both male and female mentors. Each mentor was assigned to work with four mentees, on average. However, it should be noted that mentors occasionally worked with campers outside of their assigned group. Mentors were paid \$200 after mentoring the camp. Finally, mentors were provided with an android device to borrow during both training and camps. They were encouraged to take the devices home.

**2.4.3 Campers.** The target audience for App Camps are middle-school-aged youth ranging in age from 10 to 13. We selected this age because research shows that middle school is a critical juncture for deciding academic interest and career choice [76, 77]. Campers spent 3h per day over five days (15h total) designing 11 apps and engaging in debugging activities. Our previous experience running camps suggested that 3h was ideal for this age group to maintain their productivity and attention.

As stated above, we lent an android device to each camper and encouraged them to take it home for the duration of the camp. The goal of taking devices home was to prompt conversation with their family and friends while sharing the apps they created during the camp.

## 2.5 App Camp Curriculum

Our App Camp curriculum was designed to provide instructional support on an individual camper basis as well as to encourage collaboration amongst peers and mentors. The curriculum was delivered via the Canvas learning management system. Participants were provided a login that allowed them to access the materials at home and after the camp was over. The curriculum started with simple apps that provided detailed instructions in a step-by-step format accompanied with images, which represented different coding steps. Each app built upon CS concepts taught in earlier apps while also introducing new concepts. As such, instructions for previously addressed concepts were not provided when those concepts were used again. As the scaffolds were removed, the campers were encouraged to ask questions and problem solve amongst themselves and with their mentors. A full list of apps and programming concepts is available in Appendix B. Our whole curriculum, including debugging and non-programming activities is available through our website ([appcamp.usu.edu](http://appcamp.usu.edu)).

Campers also engaged in debugging activities with peers. We hoped that by creating a collaborative atmosphere, campers would enjoy programming more and see programming as collaborative and/or social (as opposed to the existing stereotype that CS is isolating or something you do alone [13, 27, 48]). Finally, camp schedules were designed to allow approximately 30–60min a day (depending on camper programming speed) for “free” programming time. During this time, campers were encouraged to create their own apps, further personalize previous apps, and/or work on design challenges presented on the Canvas page. Not all the campers used this time to create

additional apps, but they all personalized their apps to some extent. Manipulation of media, or personalization, has been found to be important in media creation, as it allows youth to make connections with their prior knowledge and personal interest [56]. Upon making those connections, the activities become more relevant, which is a basis for more long-term interest [35], while students also become more engaged [28] and more successful [58].

To expose campers to the CS field, we provided two types of activities. First, a female CS professor gave a presentation on working in CS. The CS professor talked about salaries, availability of jobs, job flexibility, and underrepresentation of certain populations. Second, we showed multiple videos, including a video made by Code.org [16] and videos we created for the camps. The Code.org video shows several famous people (e.g., Bill Gates, Chris Bosh, Elena Silenok) talking about their first CS experiences. More importantly, it shows what working at some of the tech companies look like- open spaces with games and people collaborating. In addition, we showed three short videos (2–3min) that our research team made. Our videos highlight stories of recent college graduates who grew up in the rural, intermountain west with a background similar to our campers (e.g., growing up on a farm or growing up playing sports). These recent graduates share their experiences about how they decided to major in CS (e.g., after attending a camp or after being interested in playing video games) and what their CS jobs entail (e.g., a job with social relevance or a job with a creative component). Our goal in connection to these videos was to provide perspectives on what working in CS was like and to provide role models that our campers would find more relatable, coming from rural areas in the Intermountain West.

### 3 METHODS

Research in educational settings often employ a mixture of quantitative and qualitative data, because relationships between learning and affective change, interventions, and learning environments can be very complex and dynamic [19]. Given that our goal was to establish a relationship between affective change and camp design, we used a combination of quantitative and qualitative methods to determine both self-reported changes in affect in the aforementioned constructs (e.g., survey data) as well as possible explanations for those gains (e.g., interviews).

#### 3.1 Sample

Our camp participants were recruited from a rural area in the Intermountain West of the United States through a variety of recruitment channels. Specifically, we distributed the information about our camps through flyers, local media (e.g., radio stations and newspapers), Facebook, school listservs, and university mailing systems. While all the participants volunteered to enroll, we did observe a selection-bias in our sample. In the pre-survey, we asked campers why they signed up for the camp. They were provided options such as “I am interested in computing,” “This camp sounded fun,” “My parents signed me up,” “My friends told me about it,” and “Other.” They could select all that apply. Of our sample, 67% of the participants reported that they signed up for the camp, because they were “interested in computing” and 69% selected “this camp sounded fun.” Twenty-eight percent of the participants reported that their parents signed them up.

In respect to their demographic information, our sample consisted of 111 campers (40 males and 71 females) aged 10 to 14 years ( $M = 11.98$ ) spread across four camps. Two of our camps had mixed gender campers (e.g., males and females) and the other two were all-female camps. Table 1 shows how campers were distributed by camp. Please note, we only present data from 105 campers due to lack of parental consent or campers not showing up the last day of camp. Of the 105 campers, 68 were female and 37 were male. Camper ethnicities included Asian/Pacific Islander (1%), Native American (4%), Latino/a (5%), Other (2%), and Caucasian (88%). Seventeen percent of the campers reported being on free and/or reduced lunch.

Table 1. Camper Distribution by Camp

Camp	Camper	N <sub>campers</sub>
1	Mixed-gender	36 (19 Males, 15 Females)
2	Female-only	21 (21 Females)
3	Female-only	20 (20 Females)
4	Mixed-gender	37 (21 Males, 15 Females)

Table 2. Measures' Reliabilities Across Occasions

Measure	Cronbach's Alpha	
	Pre-Camp	Post-Camp
Father Support	0.83	0.87
Mother Support	0.76	0.81
Self-efficacy Now	0.87	0.90
Self-efficacy Future	0.76	0.86
Intrinsic Interest	0.78	0.85
Career Interest	0.76	0.76

### 3.2 Data Collection Procedures

**3.2.1 Surveys.** Students took an affective survey prior to the camp on day 1 (pre) and then again on the last day of camp (post). Surveys were administered via the online survey software, Qualtrics. See Measures (section 3.3) for more details.

**3.2.2 Interviews.** Thirty campers across the four camps were randomly selected to be interviewed on the last day of camp. Interviews were conducted during camp by one of two authors of this article (one professor and one research assistant) in two separate rooms inside the same building. They lasted around 15min, on average. Interview protocols focused on several topics, such as CS background, identity development, working with mentors and similarity to mentors. In this study, we focus on campers' responses on the support they received from their parents during camp, as well as the rationale they provided for any positive changes in their interest in and self-efficacy beliefs about CS (see Appendix A for the interview protocol). All interviews were audio recorded and transcribed verbatim.

### 3.3 Measures

The affective survey used in this study was taken from several established STEM self-report scales [12, 25, 49, 71], but adapted for CS education. The survey consisted of 32 items and instructed campers to rate the items on a scale of 1 to 8 from strongly disagree to strongly agree. Twenty-two of those items, measuring interest, self-efficacy, and parental support (see Appendix D) were used and reported on in this article. Composite scores were derived by averaging responses for each construct (three to six items). We ran analyses to confirm unidimensionality of the constructs and conducted Cronbach's alpha, a measure of internal consistency (see Table 2). For this article, we report findings from the parental support items (mother and father support), interest items (career and intrinsic interest), and self-efficacy items (self-efficacy now and self-efficacy future).

## 4 FINDINGS

Our data did not meet assumptions of normality, which is quite common when dealing with Likert-scale data [38]. Thus, to answer our three research questions, we conducted Wilcoxon signed-rank



Table 3. Wilcoxon Signed-rank Test for INTEREST in CS ( $N = 105$ )

	Pre-Camp			Post-Camp			Z	A
	M	Med	SD	M	Med	SD		
Career Interest	5.54	5.66	1.63	6.10	6.33	1.65	4.70*	0.39
Intrinsic Interest	6.79	7.00	1.08	7.09	7.5	1.08	3.41*	0.40

Notes: \* $p < 0.05$ . A = Vargha and Delaney's A. An A value of 0.5 signifies no difference between the groups. The further the score deviates from 0.5 the stronger the effect.

test analyses in R [67] using the *Coin Package* [36]. We used the nonparametric test, Vargha and Delaney's A, to calculate the effect size. In the analysis of RQ3, numbers were not normalized for the three single-parent households. Therefore, it is possible that those students over- or underestimated parental support. However, given that we looked at father and mother as separate constructs, and that only three of our participants reported coming from a single-parent unit, we do not believe the results have been affected.

To explain the change in constructs, we conducted a thematic analysis [32] of 30 post-camp interviews with campers across the four camps, which resulted in 232 codes across all three research questions, with three to four subcategories corresponding each research question. Unit of analysis was each utterance explaining the reason behind an increase in interest, an increase in self-efficacy and or the type of support they received. Initially, one coder coded the whole data set. In the next phase, two coders discussed all the subcategories developed, their exclusiveness and the codes they incorporated until they reached consensus on all the names and codes. The following section describes our findings for each individual research question.

#### 4.1 How Does Our App Camp Affect Campers' Interest in CS?

Our first research question asked about campers' interest, we measured both intrinsic interest in programming and career interest in CS. Table 3 above shows that both career interest ( $Z = 4.70$ ,  $p < 0.001$ ) and intrinsic interest ( $Z = 3.41$ ,  $p = 0.001$ ) reported significantly higher scores on the post-test after our one-week camp. As our data did not meet assumptions of normality, we used the nonparametric test Vargha and Delaney's A to calculate the effect size. The magnitudes of these changes were small for both career interest ( $A = 0.39$ ) and intrinsic interest ( $A = 0.40$ ).

To understand how and why these changes occurred, we analyzed our interview data. In the interviews, all campers mentioned at least one reason why the camp contributed to their interest in programming. Twenty students (67%) mentioned two or more reasons. We organized these statements into four overarching subcategories: *The Fun Factor* (50%), *Social Factors* (33%), *Personal Growth* (53%), and *The Role of Videos* (60%), each of which explained the reasons behind a positive interest change. The percentage mentioned in relation to subcategories refers to the number of interviews each subcategory appeared in. All subcategories are explained more thoroughly in the following subsections.

**4.1.1 The Fun Factor.** One of the reasons youth emphasized as important for their interest increase was the element of fun in programming. Whether they had prior experience that made them think programming was "*confusing*" or they considered themselves someone who was not into technology, their participation in the camp made them realize programming was "*fun*" to do:

"... at first, I thought that it might be [laughing] really boring, but now that I've done it, it is really fun to me. I just enjoy making [apps]. I just enjoy doing it."

One of the reasons the campers found programming fun was the freedom they had to personalize their apps:

*“It’s fun to figure out what’s wrong with it and fix it, and go through the process... My favorite was probably the robot one, because it took a lot of work, and it was fun just to figure out how to make it play different sounds... when you personalize it, you’re actually having fun with it and getting it done at the same time...”*

Encouraging our campers to personalize their apps was a conscious design choice that we implemented at the mentor and curriculum development level. During mentor training, we stressed the importance of not only reminding campers to personalize their apps but also to provide suggestions and inspiration for personalization opportunities. In the curriculum, we prompted campers to personalize their apps. In addition, we removed detailed instructions for some apps hoping that students would take more ownership over the design.

**4.1.2 The Social Factor.** In addition to programming being fun, some campers mentioned that certain social factors influenced their positive change in interest. Some of these factors included the ability to show the apps they made to their parents, giving and receiving help, and/or the presence of the mentors.

Many campers mentioned that bringing the device home to show off their apps sparked their interest *“because it was very fun ... [to] take the phone home to play on your apps and show them to your family...”*. The social factor also meant providing and receiving help:

*“I like the people that help me with it. I like how it’s easier than I expected... having people around me to help me and having to help people...”*

Finally, the following excerpt illustrated the role of the mentors in interest development for one camper:

*“... I learned a little bit more ‘cause Harry was like, ‘Well, what do you think you have to do before you can do this?’ He didn’t just show me, and so I had to think myself. Then I feel like it was easier”.*

We designed our camp around working with others, working with mentors, and sharing apps with family and friends. In particular, the mentors were trained to ask questions rather than provide ready-made answers, which made the learning experience more enjoyable and engaging for the campers.

**4.1.3 Personal Growth.** Personal growth involves gaining a deeper understanding of what computer programming is. Campers mentioned personal growth as important for their interest change. The level of growth itself depended on the child. Some novice campers became more interested in programming simply because they finally learned what programming actually entailed. As illustrated by the following excerpt, learning more about the scope of the field sparked youth’s curiosity to learn even more:

*“At first, I was like, ‘Hmm,’ cuz I didn’t know anything about it. Then, the more you know, the more you want to know, as well.”*

Learning more about programming included expanding their understanding of what programming is (e.g., *“Before I thought it was just like typing”*) or what you can do with it: *“[This camp] it helped me learn what it’s [CS] really about and what it is. Before I just thought, ‘Oh, you’re on a computer doing science stuff,’ but now I know exactly what it is.”*

Designing apps helped some campers realize how dynamic programming can be. This newfound realization led to an increase in interest for some campers. These campers perceived our App Camp

as more interesting than their previous programming experience(s), because they got to create more:

*“In our computer programming at school, all we learned about was how to move stuff around, it was boring. Now I figured out that I can do more stuff with computer programming besides just moving stuff around. I can program buttons and stuff like that now. I can add stuff to lists, I can take ‘em away, create variables, procedures, that sort of stuff.”*

Being able to program functional apps contributed to camper interest, self-efficacy, and ability to program more complex apps.

**4.1.4 The Role of Videos.** The videos we showed during the camp played a role in increasing CS career interest for more than half of the campers interviewed. Seeing three video stories of successful CS professionals from Intermountain West Region who were either “*athlete[s] like [them]*,” “*grew up on a farm like [them]*,” or whose career interests differed from their parents’ career choices, helped campers envision CS as a potential career in their future. Additionally, it helped the campers envision that such careers are achievable “*‘cause it kind of just shows you where you start from and where you can get to if you just keep going at it,*” all you need to do is “*keep trying.*”

Seeing glimpses of the protagonists’ professional life, such as where they work, and what kind of jobs they do with CS contributed to increasing interest in CS as a career for some campers. The following excerpt shows how the videos provided exposure to the profession:

*“I liked it when they showed us the workspace and when they talked to us about what they got to do and stuff like that... Because it’s cool to be able to see where you would work if you got into this career, and how you would do it, and what you’d be doing.”*

The videos were designed to promote an interest in CS careers by showing campers what working in CS entails. In addition, by using role models who had similar backgrounds, we hoped that campers would be more likely to envision themselves as someone who could work in CS.

## 4.2 How Does Our App Camp Affect Campers’ Self-efficacy in CS?

For our second research question, we looked at self-efficacy in two ways: self-efficacy now and the ability to learn CS in the future (self-efficacy future). Campers reported significant gains for both self-efficacy now ( $Z = 7.65, p < 0.001$ ) and self-efficacy future ( $Z = 4.99, p < 0.001$ ) with a medium effect size for self-efficacy now ( $A = 0.28$ ) and a small effect size for self-efficacy future ( $A = 0.35$ ) (see Table 4).

To explore potential factors for these changes in self-efficacy, we looked at the interview data. All participants discussed at least one reason for the positive change in their self-efficacy and three participants (10%), discussed two. We organized these utterances into two subcategories explaining why campers’ believed their self-efficacy increased: *I can do this!* (83%) and *I’ve developed independence* (33%). The percentage refers to the number of students who mentioned such reasons in relation to an increase in their self-efficacy. The following subsections explain these two subcategories in detail.

**4.2.1 I can Do This!** In their discussion of self-efficacy, some students mentioned that the camp helped them realize they do have the ability to program. These explanations were mostly given from the perspective of those campers who did not have previous programming experience or started the camp lacking confidence in their programming, because they did not think they were “*that good*” or because it was their “*first time doing something this big.*” Upon attending the camp, they realized that programming is “*easier than [they] thought,*” that they did indeed learn a lot

Table 4. Wilcoxon Signed-Rank Test for Self-Efficacy in CS ( $N = 105$ )

	Pre-Camp			Post-Camp			Z	A
	M	Med	SD	M	Med	SD		
Self-Efficacy Now	5.03	5.00	1.68	6.29	6.66	1.46	7.65*	0.28
Self-Efficacy Future	6.02	6.00	1.23	6.55	6.75	1.40	4.99*	0.35

Notes: \* $p < 0.05$ . A = Vargha and Delaney's A. An A value of 0.5 signifies no difference between the groups. The further the score deviates from 0.5 the stronger the effect.

about a new skill (e.g., programming) and that they can program. This change in self-efficacy is illustrated in one of the excerpts from the camp:

*"I feel like I could be a better programmer now and I could get more into it. I never thought before this camp that I would be a programmer... I feel like that I could do it, and I could do harder stuff, and I can just see being a programmer if I really wanted to."*

A few campers mentioned that learning to program using App Inventor was "a lot easier" than some of their previous programming experiences. While there is not enough research on the trajectory from block-based programming to languages such as Python or Java, after running our camps for three years, we see App Inventor as a good entry into programming. Campers learn some difficult concepts like lists, databases, and coordinate systems that give them confidence in their ability to program.

**4.2.2 I have Developed Independence.** In addition to lowering the perception of difficulty, some kids developed a belief in their own ability by becoming more independent at programming throughout the camp. As an example, one girl mentioned that as she began to personalize more and use the instructions less, she began to feel more confident about her programming:

*"I used to just follow the instructions and not go off 'em, but now I like to go off 'em and change a lot of the little things."*

Again, we see that by promoting personalization of the apps and by removing instructions as the students gained programming ability, students not only became more interested in programming, but they also developed more confidence in their ability to program. A few campers challenged themselves not to ask for help and managed to fix their problem independently:

*"... instead of asking for help, [I would] like re-read and re-learn and try and see what I did wrong."*

We purposefully designed the curriculum so that campers could work at that their own pace. In addition, detailed instructions for concepts within App Inventor were only provided once. Our goal was to get campers to think critically about the concepts and program without instructions. Although we did provide campers with the support of mentors, some campers gained confidence by figuring it out themselves.

### 4.3 How Does Our App Camp Affect Campers' Perceptions of Parental Support Systems?

In our third research question, we were interested in how our camps affect perception of parental support. Here it is important to acknowledge that we do not believe that parental strategies or behaviors will change as a result of a one-week camp. However, we do argue that participants' *perceptions* of parental support may have been influenced by conversations about the camp and

Table 5. Wilcoxon Signed-Rank Test for Parental Support ( $N = 105$ )

	Pre-Camp			Post-Camp			Z	A
	M	Med	SD	M	Med	SD		
Mother Support	6.00	6	1.26	6.35	6.75	1.48	4.63*	0.41
Father Support	6.08	6.25	1.47	6.43	6.75	1.52	4.06*	0.42

Notes: \* $p < .05$ . A = Vargha and Delaney's A. An A value of 0.5 signifies no difference between the groups. The further the score deviates from 0.5 the stronger the effect.

the apps campers made. To observe the effect of our camp on how participants perceived parental support, we compared camper pre- and post-camp perceptions of their mothers' and fathers' interest in and overall support for their CS studies. Table 5 below shows that campers reported higher levels of perceived mother ( $Z = 4.63$ ,  $p < 0.001$ ) and father ( $Z = 4.06$ ,  $p < 0.001$ ) support after attending our one-week App Camp with a small effect size for both mother ( $A = 0.41$ ) and father ( $A = 0.42$ ) support.

To explore potential reasons for these changes, we examined camper responses to interview questions that asked about the interactions between campers and their parents during the camp. In terms of interactions with their fathers, nine campers (30%) reported not having direct interactions during the camp due to their father's work. We organized the types of parental support that they reported receiving into three subcategories: Career Discussions (33% with mother, 30% with father, and 17% reporting having those discussions with both parents), Quality Time (60% with mother, 43% with father, and 20% reported spending quality time with both parents), and Approval/Praise (97% from mother, 47% from father, and 43% reporting receiving praise from both parents). In the following sections, we describe interactions with mothers and fathers, respectively, in detail.

#### 4.4 Mother Support

Campers reported receiving mother support more frequently than father support. After analyzing the data, we learned that each individual camper shared the apps that they had created with their mothers, which resulted in them receiving at least one type of three reported types of support, with 21 (70%) of campers reporting such interactions resulting in Quality Time or Career Discussions with their mothers, in addition to Praise.

**4.4.1 Praise.** The type of support that the campers received most frequently from their mothers was praise. Upon showing the mothers their apps, the mothers frequently characterized them as "cool," "great," "creative," "awesome," "interesting," and/or "nice." Alternatively, they told their children that they were proud of them or that they did a good job. This type of support was valuable to the campers as it made the youth feel good, proud of their work, supported and/or served as a confirmation of the quality of the apps that they made. Such importance is reflected in the words of one of the campers:

*"It made me feel like I did something. [Laughter] Instead of just like okay. I did this. Now what? [...] It made me feel like I did something I could show to people."*

Occasionally, the mothers would emphasize the fact that they themselves did not know how to program. This type of feedback made the youth feel even more proud of their work, or in the words of one camper, made them feel that they are "a bit more advanced than a lot of other people," which made them feel special especially if those people were older than themselves.

**4.4.2 Quality Time.** Some campers also reported spending Quality Time with their mothers in relation to their app creations. According to these testimonies, after picking their children up from

camp, mothers would either examine the app(s) that the youth made or have conversations with their children about their camp and/or programming experience. App examination ranged from passive observations of their child's demonstrations of apps to active interactions with and testing of the apps. This quality time with their mothers around their apps was important to the youth, because it made them feel good about their programming experience and the skill that they were developing:

*“It made me feel good that I knew something and that she didn't know that, and that I could teach her how, 'cause it just made me feel special.”*

This quality time was made possible by bringing their android device home. It is also important to note that these moments of parental interactions seem to also benefit the campers' interest and self-efficacy.

**4.4.3 Career/skill Discussion.** Finally, some campers we interviewed mentioned having some type of discussion with their mother about the benefits of developing programming skills or even considering programming as a career. Mothers would confirm computer programming as a “good skill to have” or would suggest other ways to develop CS skills:

*“We talked about what college I would go to, to go into computer science. (The camper names some universities). [...] It was cool to find out how many classes and stuff you could take, and how you can get better at doing everything.”*

For this camper, bringing the android device home and showing her mother the apps she made became a mediator for conversation about careers in CS. It is this type of interaction that can have a powerful effect on youth and career choices.

## 4.5 Father Support

As mentioned before, not all the campers received fathers' support during our camp. These answers were coded as Absence. Those that did receive father support, received the same type of support that they did from their mothers (aka *Praise*, *Quality Time*, and discussions or conversations about campers' programming skills and future career [*Skill/Career Discussion*]), but in a slightly different shape and form.

**4.5.1 Absence.** Father support was not as prominent as mother support mostly due to the fact that some campers reported their father as being absent. Some reasons youth provided for the absence of their male parent were business trips, engagement with after work projects, or simply the fact that they “work most of the time” and come home late.

**4.5.2 Quality Time.** Among the campers who did talk to their fathers about the apps that they made, quality father-child time was the most frequent way fathers provided support for their children. The fathers either engaged in a conversation about the programming experience or they engaged with the apps themselves. Youth reported showing the apps they made to their fathers and fathers testing them out. All the conversations the youth described were about them explaining to their fathers how the apps worked, how you code in App Inventor, or how their own apps were personalized in comparison to other campers. This type of interaction made the youth feel good and supported, as it is illustrated by the following excerpt:

*“It makes me feel good that he wants to know more about it and that he thinks that it's something that's really cool.”*



Again, bringing the android device home and sharing their apps was a mediator for fathers to discuss programming and encourage their child's CS interest.

**4.5.3 Praise.** Praise appeared in some interactions with campers' fathers as well. More precisely, fathers expressed liking the apps that their children made and referred to them as "cool," "great," or "inspiring." One camper said:

*"My dad gets really excited when I make a new app, and he's always all like, that's so cool that you can do that and I can't do that."*

For this camper, their father was impressed, because he was not familiar with CS or programming. This type of praise has the potential to benefit campers' self-efficacy and foster interest in CS.

**4.5.4 Career/skill Discussion.** Finally, some campers also reported that their involvement with the camp brought some type of discussion with their father about the benefits of developing programming skills or how such skills could fit their future career. Campers reported telling their father that they were interested in CS as a career, which led to conversations about the importance of planning one's career early, going to a good college, as well as whether or not CS is a good career:

*"He said it would be a really fun job, and that I could probably go pretty far in it. [It made me feel] more confident about my skills... Because to have my father tell me I was good at it."*

This quote shows how important it is bridge informal learning experiences to the home and involve fathers (and mothers) in some way. Without the ability to take the android device home and share their apps, this conversation may not have happened.

It is important to note that many of these forms of support and encouragement may be common practice in the homes of our campers regardless of their attendance in our camp. Thus, we do not claim that our camp is changing parental practices, rather we argue that by allowing our campers to take their android devices home and by encouraging our campers to personalize and take ownership over their apps, we may be creating a proxy for discussions specifically about CS and the opportunities afforded by learning CS skills. These discussions and interactions around CS may heighten a child's awareness or recognition of parental interest in or support of the pursuit of CS knowledge and skills. The significance of these findings will be expanded on in the following section.

## 5 DISCUSSION

In this article, we set out to evaluate the effectiveness of our camp design in terms of increasing youths' interest, self-efficacy beliefs, and perceptions of parental support. We found significant gains for all three of these concepts. An additional goal was to make connections between our design choices (e.g., videos, peer support, mentor support) and the affective gains.

The camp was designed using principles found to broaden participation in CS, such as an opportunity to program, exposure to the field, social support, and designs that support self-efficacy [74]. Additionally, we allowed and encouraged youth to share artifacts with friends and family, which has been found to increase self-efficacy [62]. We also encouraged peer interactions [55], organized mentor support [42], and provided career-related videos [18].

Overall, we found statistically significant results for all three constructs. In terms of interest, participants reported a significant increase in both career interest and intrinsic interest after attending the camp. Interview data suggests that an increase in interest may have been the result of

several factors. Some campers perceived the camps as “fun”, directly connecting personalization as something that contributed to the fun factor. This is aligned to research that has attempted to integrate creativity into CS courses as a way to promote interest for women [60]. Other campers believed that it was the social factor that contributed to their interest gains, such as peer interactions and mentor support. These findings are also aligned to the literature [15, 42, 55]. Some campers reported gaining a deeper understanding of what programming entailed and what you can do with it. In addition, campers realized that programming with App Inventor was easier than they thought it would be. Having the app that they created on the phone was a source of confidence.

Some of the campers’ interest was sparked by the videos we showed in the camp. While many studies used videos as a tool to expose youth to the CS profession [18, 43], our study was slightly different. We designed videos with people who work in the CS field that we thought our campers would relate to—people with similar backgrounds and experiences. Our findings illustrate how important relatable role models are for youth when being introduced to a new field such as CS.

Finally, we found that by taking home the android devices and sharing their apps, several campers had conversations about working in CS with their parents. This is important, because past research has indicated that parental support can play a role in youth’s career interest [39, 59].

In addition to gains in interest, we also found gains in self-efficacy. This is similar to other studies that found self-efficacy gains among the campers in their informal learning programs [4, 23, 30, 51, 62, 65]. Few studies, however, linked their findings to concrete design features of their camp. Our study provides such links through the analysis of youth’s interviews. We found that youth who had no prior experience with programming gained more confidence in their ability to program during the camp. We believe this is because we used App Inventor, which is a block-programming language. It has a low entry threshold for novice programmers. In addition, our curriculum was designed to provide scaffolding that was slowly faded out. The apps progressed in difficulty and the concepts built off each other. The curriculum allowed campers to work at their own pace and to also solicit help from their near-peer mentors and fellow campers when needed.

Youth who had previous experience with programming dedicated more time to personalizing their apps, and attempted to code independently, both of which gave them higher self-efficacy beliefs. In prior research, we found that due to the proximity in age and expertise, as well as their approachability and role modeling, near-peer mentors provide a model with which young learners could relate, which affected both their interest and self-efficacy [15]. We believe the gains in self-efficacy is due to a combination of the curriculum, near-peer mentors, and the block programming environment used to develop the apps.

We also found increases in student perceptions of parental support. As many studies found parental support to be important for youths’ career choices [39, 53, 68], we created opportunities for youth to discuss their artifacts with their parents. In an earlier study [14], we found that both mother and father support led to an increase in CS interest. In this study, we add to that literature by providing descriptions of the type of support youth received from their parents. We learned that youth more frequently interacted with their mothers around their apps. Mothers usually provided praise for their artifacts and programming, tested their apps, wanted to learn more about programming, or occasionally engaged in conversations about CS careers. Among the youth who shared their apps with their fathers, similar trends were reported. Fathers, however, seemed to be more willing to interact with the apps than provide praise. We do not claim that our one-week camp has changed parental behaviors or the amount of support that they provide to their children. We do argue that involvement in our camp, and the sharing of apps created in our camp, do function as a potential mediator for conversations about CS. These conversations can be vital for

recruiting youth into CS. These findings suggest that future camps should explore ways to involve parents in their child's CS learning experience.

## 6 LIMITATIONS AND FUTURE RESEARCH

Our study has a few notable limitations. First, our campers voluntarily attended our camp. As evident by the reasons they reported for signing up for the camps, as well as their high pre-test scores, the majority of them had prior interest in CS. Future research will need to determine the effectiveness of practices established in this study for those who may not be as interested in CS as much as our campers were. Second, we only looked at the effect of our camp on affect gains directly after the camp. Future research will need to examine long-term effects on constructs such as campers' CS interest in general and interest in CS as a career. Future research will also need to investigate the effect of the camp on campers' CS knowledge and skills, which were beyond the scope of our study. In addition, we explored the associations between campers' affect gains and our camp design using qualitative methods. Future research can test and quantify the significance and strengths of the factors we identified to the affect gains.

## 7 CONCLUSION

In this article, we intended to provide researchers and practitioners with a model and a set of principles for both designing and running similar CS programs. We provided a detailed overview of the camp design used in this study and made connections between our program designs and our desired outcomes. We found positive gains in self-efficacy, interest, and perceptions of parental support, and we illustrated them with potential explanations from campers' interviews. We also illustrated how our camp designs may have led to the gains found.

## APPENDICES

### A INTERVIEW PROTOCOLS

#### INTEREST IN COMPUTER PROGRAMMING

- Has your interest toward programming changed during this camp? How so? Why?
- Are there any experiences that stand out to support this change?

#### PERCEIVED SELF-EFFICACY IN COMPUTER PROGRAMMING

- Has your confidence in your ability to program changed during this camp? How so? Why?
- Are there any experiences that stand out to support this change?

#### PERCEIVED SUPPORT

- Have you shared any apps you created this week with your mother? Did this lead to any conversations about programming or computer science? What was it about? How did that make you feel?
- Have you shared any apps you created this week with your father? Did this lead to any conversations about programming or computer science? What was it about? How did that make you feel?

## B CAMP ORGANIZATION WITH APPS AND CONCEPTS BEING TAUGHT

Day # and Project Name	About the app(s)	Concepts/Skills targeted
DAY 1: Talk to Me and Paint Pot	Talk to Me app is their first exposure to programming. While the programming is simple, the students get to explore a rich collection of concepts.	App Inventor layout, QR code, when event, procedure call, button, text, textbox, accelerometer sensor, text to speech, strings, good variable naming
	In PaintPot students learn the concept of creating a graphical image. Line drawing, pixels, and color representation are all important.	Camera, colors, drawing on canvas, coordinate system, images, paint colors, variables, slider, horizontal arrangements, after picture, algorithms, parameters
DAY 2: Channel Surfing, I'm Not a Robot, and Counting	Channel surfing allows the students to use audio files	Conditional logic-if.then.else blocks
	I am not a robot introduces modularity.	Table Arrangement, Invisible components, procedures, parameters.
	Counting introduces variables and increments	Clock components, textToSpeech, enabling/disabling a timer.
DAY 3: Riddle me This, and Color me a Rainbow	Riddle Me This is a first exposure to lists.	Lists, indexing, when.click blocks
	Color Me A Rainbow exposes students to colors as a mixture of red/green/blue and uses a preview canvas to see the colors before applying them to the drawing.	Multiple canvases, colors, horizontal arrangement, sliders. More work with procedures. Use of coordinate system on canvas.
DAY 4: Positive Self-Talk and Excuse Generator	In Positive Self-talks, students practice their list skills with parallel lists.	Editing of sound files, parallel lists, indexing.
	In Excuse Generator app, lists are revisited as students create a list of excuses that are selected at random. Random numbers are used. User input is utilized.	Lists, tinyDB, adding/deleting from lists, compound conditions
DAY 5: Favorite Sport & Selfie Slideshow	Favorite sport is a whack-a-mole variant.	Sprites, increments, coordinate systems
	Selfie slide show combines lists, camera, drawing on the pictures, and shapes	Adding, deleting, and replacing in a list.

**C LIST OF REVIEWED STUDIES AND EFFECT SIZES OF CONSTRUCTS OF INTEREST**

ID	Study	Self-efficacy	Effect Size	
			Interest	Social Support
1	Adams (2007)		n/a	
2	Al-Bow et al. (2009)	n/a		
3	Al-Duwis et al. (2013)		n/a	
4	Aritajati et al. (2015)	Cohen's d = 0.27		
5	Burge et al. (2013)		Cramer's V = 0.26	
6	Clarke-Midura et al. (2018)	Cohen's d = 0.91	Cohen's d = 0.27	
7	Craig and Horton (2009)		n/a	
8	DeWitt et al. (2017)	n/a	n/a	
9	DiSalvo et al. (2013)	Vargha & Delaney's A = 0.50–0.69*	Vargha & Delaney's A = 0.48–0.64	
10	Franklin et al. (2013)			
11	Franklin, Aldana, and Hough (2011)		Phi = 0.41	
12	Gannod et al. (2013)	Cohen's d = 0.20		
13	Graham and Latulipe (2003)			
14	Hulsey et al. (2014)	n/a		
15	Khoja et al. (2012)			
16	Lau et al. (2009)		n/a	
17	Layer et al. (2014)	n/a	n/a	
18	Ni, Harunani, and Martin (2017)			
19	Ni, Schilder et al. (2016)	Cohen's d = 1.13		
20	Outlay (2016)			
21	Pollock et al. (2004)			
22	Roy (2012)	n/a	n/a	
23	Sabin et al. (2017)	Cohen's d = 1.71	Cohen's d = 0.09	Cohen's d = 0.39
24	Starett et al. (2015)	n/a	n/a	
25	Sullivan et al. (2015)	Cohen's d = 1.59		
26	Urness and Manley (2013)	n/a	n/a	n/a
27	Vachovsky et al. (2016)	n/a	n/a	
28	van Delden and Yang (2016)		n/a	
29	Wagner et al. (2013)			
30	Wang et al. (2012)		Cohen's d = 0.57**	
31	Webb and Rosson (2011)		Cohen's d = 0.31	

*Note:* Studies that have blank space for effect size either presented qualitative findings or did not examine the constructs of interest. Studies that have N/A for effect size presented quantitative evidence for the construct(s) of interest. However, their effect size information was unobtainable: They neither reported the effect size information nor provided requisite information (such as standard deviations and significance test value) for the research team to calculate the effect sizes.

\*It is the range of effect sizes for item-level changes, that is, authors of the study computed effect sizes for each item that measures the same variable.

\*\*This study used the letter *g* to denote effect sizes and stated that they were computed using Cohen's measures. We could not identify Cohen's *g* as a measure of effect size in the literature. Additionally, from the statistical techniques the article used to test changes, we hence believe that the effect sizes reported in the study are factually Cohen's *d*.

## D MEASURES

### *Intrinsic Interest*

I enjoy computer programming.

I am interested in learning more about computer programming.

I think computer programming is interesting.

I think computer programming is boring.

### *Career Interest*

I can't imagine myself working in a computer science career.

I would like to get a degree in computer science.

I am interested in working in a job that involves computer science.

### *Self-Efficacy Now*

I can program computers well.

I am a good computer programmer.

I am confident in my ability to program computers.

### *Self-efficacy Future*

If I took a class on computer programming, I could do well.

If I wanted to, I could be a computer programmer in the future.

I think I could do more challenging computer programming.

I can learn to do computer programming.

### *Father Support*

My father has encouraged me to learn computer programming.

My father has shown no interest in whether I learn computer programming.

My father thinks I could be a good computer programmer.

My father thinks I'll need to learn computer programming for the future.

### *Mother Support*

My mother thinks I could be a good computer programmer.

My mother has shown no interest in whether I learn computer programming.

My mother has encouraged me to learn computer programming.

My mother thinks I'll need to learn computer programming for the future.

## ACKNOWLEDGMENTS

Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation or Utah State University.

## REFERENCES

- [1] Joel C. Adams. 2007. Alice, middle schoolers, and the imaginary worlds camps. *ACM SIGCSE Bull.* 39, 1 (2007), 307–311.
- [2] Mohammed Al-Bow, Debra Austin, Jeffrey Edgington, Rafael Fajardo, Joshua Fishburn, Carlos Lara, Scott Leutenegger, and Susan Meyer. 2009. Using game creation for teaching computer programming to high school students and teachers. *ACM SIGCSE Bull.* 41, 3 (2009), 104–108.



- [3] M. Al-Duwis, H. S. Al-Khalifa, M. S. Al-Razgan, N. Al-Rajebah, and A. Al-Subaihini. 2013. Increasing high school girls awareness of computer science through summer camp. In *Proceedings of the IEEE Global Engineering Education Conference (EDUCON'13)*. 231–237. DOI : <https://doi.org/10.1109/EduCon.2013.6530111>
- [4] Chulakorn Aritajati, Mary Beth Rosson, Joslenne Pena, Dana Cinque, and Ana Segura. 2015. A socio-cognitive analysis of summer camp outcomes and experiences. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. 581–586.
- [5] Albert Bandura. 1977. Self-efficacy: Toward a unifying theory of behavioral change. *Psychol. Rev.* 84, 2 (1977), 191.
- [6] Albert Bandura. 1982. Self-efficacy mechanism in human agency. *Am. Psychol* 37, 2 (1982), 122–147. DOI : <https://doi.org/10.1037/0003-066X.37.2.122>
- [7] Albert Bandura. 1986. *Social Foundations of Thought and Action: A Social Cognitive Theory*. Prentice-Hall, Inc, Englewood Cliffs, NJ.
- [8] Albert Bandura. 1997. *Self-efficacy: The Exercise of Control*. W H Freeman/Times Books/ Henry Holt & Co, New York, NY.
- [9] Kristy Elizabeth Boyer, E. Nathan Thomas, Audrey S. Rorrer, Deonte Cooper, and Mladen A. Vouk. 2010. Increasing technical excellence, leadership and commitment of computing students through identity-based mentoring. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE'10)*. 167–171. DOI : <https://doi.org/10.1145/1734263.1734320>
- [10] Jennifer J. Burg, Victor Paúl Pauca, William H. Turkett, Errin W. Fulp, Samuel S. Cho, Peter Santago, Daniel A. Cañas, and H. Donald Gage. 2015. Engaging non-traditional students in computer science through socially inspired learning and sustained mentoring. In *Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE'15)*. DOI : <https://doi.org/10.1145/2676723.2677266>
- [11] Janet E. Burge, Gerald C. Gannod, Maureen Doyle, and Karen C. Davis. 2013. Girls on the go: A CS summer camp to attract and inspire female high school students. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*. 615–620.
- [12] Cheryl Carrico and Chosang Tendhar. 2012. The use of the social cognitive career theory to predict engineering students' motivation in the produced program. In *Proceedings of the 119th ASEE Annual Conference and Exposition*. 25.1354.1–25.1354.13.
- [13] Sapna Cheryan, Victoria C. Plaut, Caitlin Handron, and Lauren Hudson. 2013. The stereotypical computer scientist: Gendered media representations as a barrier to inclusion for women. *Sex Roles* 69, 1–2 (July 2013), 58–71. DOI : <https://doi.org/10.1007/s11199-013-0296-x>
- [14] Jody Clarke-Midura, Frederick J. Poole, Katarina Pantic, Chongning Sun, and Vicki Allan. 2018. How mother and father support affect youths' interest in computer science. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. 215–222.
- [15] Jody Clarke-Midura, Frederick Poole, Katarina Pantic, Megan Hamilton, Chongning Sun, and Vicki Allan. 2018. How near peer mentoring affects middle school mentees. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 664–669.
- [16] Code.org. 2013a. *What Most Schools Don't Teach—Short Film*. Retrieved from <https://www.youtube.com/watch?v=mPiGHyIXKtw>.
- [17] Annemieke Craig, Jo Coldwell-Neilson, and Jenine Beekhuizen. 2013. Are IT interventions for girls a special case? In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*. 451–456.
- [18] Michelle Craig and Diane Horton. 2009. Gr8 designs for Gr8 girls: A middle-school program and its evaluation. In *Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE'09)*. DOI : <https://doi.org/10.1145/1508865.1508949>
- [19] John W. Creswell and Vicki L. Plano Clark. 2010. *Designing and Conducting Mixed Methods Research* (2nd ed.). SAGE Publications, Inc., Los Angeles.
- [20] Sebastian van Delden and Kuo-Pao Yang. 2014. Robotics summer camps as a recruiting tool: A case study. *J. Comput. Sci. Coll.* 29, 5 (2014), 14–22.
- [21] Jill Denner. 2011. What predicts middle school girls' interest in computing? *Int. J. Gen. Sci. Technol* 3, 1 (2011). Retrieved from <http://genderandset.open.ac.uk/index.php/genderandset/article/view/106>.
- [22] Anita DeWitt, Julia Fay, Madeleine Goldman, Eleanor Nicolson, Linda Oyolu, Lukas Resch, Jovan Martinez Saldaña, Soulideth Soumalath, Tyler Williams, and Kathryn Yetter. 2017. Arts coding for social good: A pilot project for middle-school outreach. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. 159–164.
- [23] Betsy DiSalvo, Mark Guzdial, Charles Meadows, Ken Perry, Tom McKlin, and Amy Bruckman. 2013. Workifying games: Successfully engaging african american gamers with computer science. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*. 317–322.
- [24] Daryl D'Souza, Margaret Hamilton, James Harland, Peter Muir, Charles Thevathayan, and Cecily Walker. 2008. Transforming learning of programming: A mentoring project. In *Proceedings of the 10th Conference on Australasian Computing Education—Volume 78 (ACE'08)*. 75–84. Retrieved from <http://dl.acm.org/citation.cfm?id=1379249.1379256>.

- [25] Elizabeth Fennema and Julia A. Sherman. 1976. Fennema-sherman mathematics attitudes scales: Instruments designed to measure attitudes toward the learning of mathematics by females and males. *J. Res. Math. Educ.* 7, 5 (1976), 324–326.
- [26] Deborah Fields, Veena Vasudevan, and Yasmin B. Kafai. 2015. The programmers’ collective: Fostering participatory culture by making music videos in a high school scratch coding workshop. *Interact. Learn. Environ.* 23, 5 (Sept. 2015), 613–633. DOI : <https://doi.org/10.1080/10494820.2015.1065892>
- [27] Allan Fisher, Jane Margolis, and Faye Miller. 1997. Undergraduate women in computer science: Experience, motivation and culture. In *Proceedings of the 28th SIGCSE Technical Symposium on Computer Science Education (SIGCSE’97)*. 106–110. DOI : <https://doi.org/10.1145/268084.268127>
- [28] Andrea Forte and Mark Guzdial. 2004. Computers for communication, not calculation: Media as a motivation and context for learning. In *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS’04)*, 40096.1. Retrieved from <http://dl.acm.org/citation.cfm?id=962752.962945>.
- [29] Diana Franklin, Phillip Conrad, Gerardo Aldana, and Sarah Hough. 2011. Animal tlatoque: Attracting middle school students to computing through culturally-relevant themes. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. 453–458.
- [30] G. C. Gannod, J. E. Burge, V. McLe, M. Doyle, and K. C. Davis. 2014. Increasing awareness of computer science in high school girls. In *Proceedings of the IEEE Frontiers in Education Conference (FIE’14)*. 1–8. DOI : <https://doi.org/10.1109/FIE.2014.7044456>
- [31] Janet L. Gates. 2002. Women’s career influences in traditional and nontraditional fields. *Poster Resented at the 9th Biennial Meeting of the Society for Research in Adolescence*. New Orleans, LA.
- [32] Corrine Glesne. 2015. *Becoming Qualitative Researchers: An Introduction*. Pearson.
- [33] Google Inc. and Gallup Inc. 2016. *Trends in the State of Computer Science in U.S. K-12 Schools*. Retrieved from <http://goo.gl/j291E0>.
- [34] Sandy Graham and Celine Latulipe. 2003. CS girls rock: sparking interest in computer science and debunking the stereotypes. In *ACM SIGCSE Bulletin*. 322–326.
- [35] Suzanne Hidi and K. Ann Renninger. 2006. The four-phase model of interest development. *Educ. Psychol* 41, 2 (2006), 111–127. DOI : [https://doi.org/10.1207/s15326985ep4102\\_4](https://doi.org/10.1207/s15326985ep4102_4)
- [36] Torsten Hothorn, Kurt Hornik, Mark A. Van De Wiel, and Achim Zeileis. 2008. Implementing a class of permutation tests: the coin package. *J. Stat. Softw.* 28, 8 (2008), 1–23.
- [37] Caitlin Hulsey, Toni B. Pence, and Larry F. Hodges. 2014. Camp cybergirls: Using a virtual world to introduce computing concepts to middle school girls. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. 331–336.
- [38] Susan Jamieson. 2004. Likert scales: How to (ab) use them. *Med. Educ.* 38, 12 (2004), 1217–1218.
- [39] Kathleen M. Jodl, Alice Michael, Oksana Malanchuk, Jacquelynne S. Eccles, and Arnold Sameroff. 2001. Parents’ roles in shaping early adolescents’ occupational aspirations. *Child Dev.* 72, 4 (2001), 1247–1266. DOI : <https://doi.org/10.1111/1467-8624.00345>
- [40] Dalorion Johnson, J. Michael Wyss, Jeff Gray, Shaundra Daily, Alan Shih, and Gypsy Abbot. 2012. Broadening participation in computing: The multi-tiered approach. In *Proceedings of the 50th Annual Southeast Regional Conference*. 371–372.
- [41] Yasmin B. Kafai and Quinn Burke. 2014. *Connected Code: Why Children Need to Learn Programming*. MIT Press.
- [42] Yasmin Kafai, Jean Griffin, Quinn Burke, Michelle Slattery, Deborah Fields, Rita Powell, Michele Grab, Susan Davidson, and Joseph Sun. 2013. A cascading mentoring pedagogy in a CS service learning course to broaden participation and perceptions. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*. 101–106.
- [43] Shereen Khoja, Camille Wainwright, Juliet Brosing, and Jeffrey Barlow. 2012. Changing girls’ attitudes towards computer science. *J. Comput. Sci. Coll.* 28, 1 (2012), 210–216.
- [44] Winnie WY Lau, Grace Ngai, Stephen CF Chan, and Joey CY Cheung. 2009. Learning programming through fashion and design: A pilot summer course in wearable computing for middle school students. In *ACM SIGCSE Bulletin*. 504–508.
- [45] Ryan Layer, Mark Sherriff, and Luther Tychonievich. 2012. “Inform, experience, implement”—teaching an intensive high school summer course. In *Proceedings of the Frontiers in Education Conference*. 1–6.
- [46] Robert W. Lent, Steven D. Brown, and Gail Hackett. 1994. Toward a unifying social cognitive theory of career and academic interest, choice, and performance. *J. Vocat. Behav.* 45, 1 (1994), 79–122.
- [47] John H. Maloney, Kylie Pepler, Yasmin Kafai, Mitchel Resnick, and Natalie Rusk. 2008. Programming by choice: Urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education (SIGCSE’08)*. 367–371. DOI : <https://doi.org/10.1145/1352135.1352260>
- [48] Jane Margolis and Allan Fisher. 2003. *Unlocking the Clubhouse: Women in Computing*. MIT Press.

- [49] Christine McLachlan, Annemieke Craig, and Jo Coldwell. 2010. Student perceptions of ICT: A gendered analysis. In *Proceedings of the 12th Australasian Conference on Computing Education*. 103, 127–136.
- [50] Lijun Ni, Farzeen Harunani, and Fred Martin. 2017. Empowering Middle school students to create data-enabled social apps. *J. Comput. Sci. Coll.* 32, 6 (2017), 88–100.
- [51] Lijun Ni, Mark Sherman, Diane Schilder, and Fred Martin. 2016. Computing with a community focus: An app inventor summer camp for middle school students. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 690–690.
- [52] Christina N. Outlay. 2016. Targeting underrepresented minority and low-income girls for computing camps: Early results and lessons learned. *J. Comput. Sci. Coll.* 31, 5 (2016), 85–94.
- [53] Sylvia Palmer and Larry Cochran. 1988. Parents as agents of career development. *J. Couns. Psychol.* 35, 1 (1988), 71–76. DOI : <https://doi.org/10.1037/0022-0167.35.1.71>
- [54] Ramona Paloš and Loredana Drobot. 2010. The impact of family influence on the career choice of adolescents. *Procedia Soc. Behav. Sci.* 2, 2 (2010), 3407–3411. DOI : <https://doi.org/10.1016/j.sbspro.2010.03.524>
- [55] Katarina Pantic, Deborah A. Fields, and Lisa Quirke. 2016. Studying situated learning in a constructionist programming camp: A multimethod microgenetic analysis of one girl’s learning pathway. In *Proceedings of the The 15th International Conference on Interaction Design and Children (IDC’16)*, 428–439. DOI : <https://doi.org/10.1145/2930674.2930725>
- [56] Kylie A. Pepler and Yasmin B. Kafai. 2007. From supergoo to scratch: Exploring creative digital media production in informal learning. *Learn. Media Technol.* 32, 2 (2007), 149–166. DOI : <https://doi.org/10.1080/17439880701343337>
- [57] Lori Pollock, Kathleen McCoy, Sandra Carberry, Namratha Hundigopal, and Xiaoxin You. 2004. Increasing high school girls’ self confidence and awareness of CS through a positive summer experience. In *ACM SIGCSE Bulletin*. 185–189.
- [58] Leo Porter, Mark Guzdial, Charlie McDowell, and Beth Simon. 2013. Success in introductory programming: What works? *Commun. ACM* 56, 8 (2013), 34–36.
- [59] Simon Lloyd D. Restubog, Afryll R. Florentino, and Patrick Raymond James M. Garcia. 2010. The mediating roles of career self-efficacy and career decidedness in the relationship between contextual support and persistence. *J. Vocat. Behav.* 77, 2 (2010), 186–195. DOI : <https://doi.org/10.1016/j.jvb.2010.06.005>
- [60] Lauren Rich, Heather Perry, and Mark Guzdial. 2004. A CS1 course designed to address interests of women. In *ACM SIGCSE Bulletin*. 190–194.
- [61] Krishnendu Roy. 2012. App inventor for android: report from a summer camp. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*. 283–288.
- [62] Mihaela Sabin, Rosabel Deloge, Adrienne Smith, and Wendy DuBow. 2017. Summer learning experience for girls in grades 7–9 boosts confidence and interest in computing careers. *J. Comput. Sci. Coll.* 32, 6 (2017), 79–87.
- [63] Jacqueline C. Simpson. 2003. Mom matters: Maternal influence on the choice of academic major. *Sex Roles* 48, 9–10 (2003), 447–460.
- [64] Courtney Starrett, Marguerite Doman, Chlotia Garrison, and Merry Sleigh. 2015. Computational bead design: A pilot summer camp in computer aided design and 3D printing for middle school girls. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. 587–590.
- [65] Kevin Sullivan, Jake Rowan Byrne, Nina Bresnihan, Katriona O’Sullivan, and Brendan Tangney. 2015. CodePlus—designing an after school computing programme for girls. In *Proceedings of the IEEE Frontiers in Education Conference (FIE’15)*, 1–5.
- [66] Rahman Tashakkori, James T. Wilkes, and Edward G. Pekarek. 2005. A systemic mentoring model in computer science. In *Proceedings of the 43rd Annual Southeast Regional Conference*. 1, 371–375.
- [67] R. Core Team. 2014. *R: A Language and Environment For Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 2013.
- [68] Sherri Turner and Richard T. Lapan. 2002. Career self-efficacy and perceptions of parent support in adolescent career development. *Career Dev. Q* 51, 1 (2002), 44–55.
- [69] Timothy Urness and Eric D. Manley. 2013. Generating interest in computer science through middle-school android summer camps. *J. Comput. Sci. Coll.* 28, 5 (2013), 211–217.
- [70] Marie E. Vachovsky, Grace Wu, Sorathan Chaturapruek, Olga Russakovsky, Richard Sommer, and Li Fei-Fei. 2016. Toward more gender diversity in CS through an artificial intelligence summer program for high school girls. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 303–308.
- [71] Ioanna Vekiri and Anna Chronaki. 2008. Gender issues in technology use: Perceived social support, computer self-efficacy and value beliefs, and computer use beyond school. *Comput. Educ.* 51, 3 (2008), 1392–1404.
- [72] Amber Wagner, Jeff Gray, Jonathan Corley, and David Wolber. 2013. Using app inventor in a K-12 summer camp. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*. 621–626.
- [73] Chia Q. Wang, Carmen Tang, Liyang Zhang, and Diana Cukierman. 2012. Try/CATCH—a CS outreach event organized by female university students for female high school students: A positive experience for all the parts involved. In *Proceedings of the 17th Western Canadian Conference on Computing Education*. 12–18.

- [74] Jennifer Wang, Hai Hong, Jason Ravitz, and Marielena Ivory. 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*. 117–122.
- [75] Heidi C. Webb and Mary Beth Rosson. 2011. Exploring careers while learning Alice 3D: A summer camp for middle school girls. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. 377–382.
- [76] Lecia J. Baker, Eric Snow, Kathy Garvin-Doxas, and Tim Weston. 2006. Recruiting middle school girls into IT: Data on girls' perceptions and experiences from a mixed-demographic group. In *Women and Information Technology: Research on Underrepresentation*, J. McGrath Cohoon and William Aspray (Eds.). The MIT Press, Cambridge, MA, 115–135.
- [77] Philip M. Sadler, Gerhard Sonnert, Zahra Hazari, and Robert Tai. 2012. Stability and volatility of STEM career interest in high school: A gender study. *Sci. Educ.* 96, 3 (2012), 411–427.

Received July 2018; revised March 2019; accepted March 2019