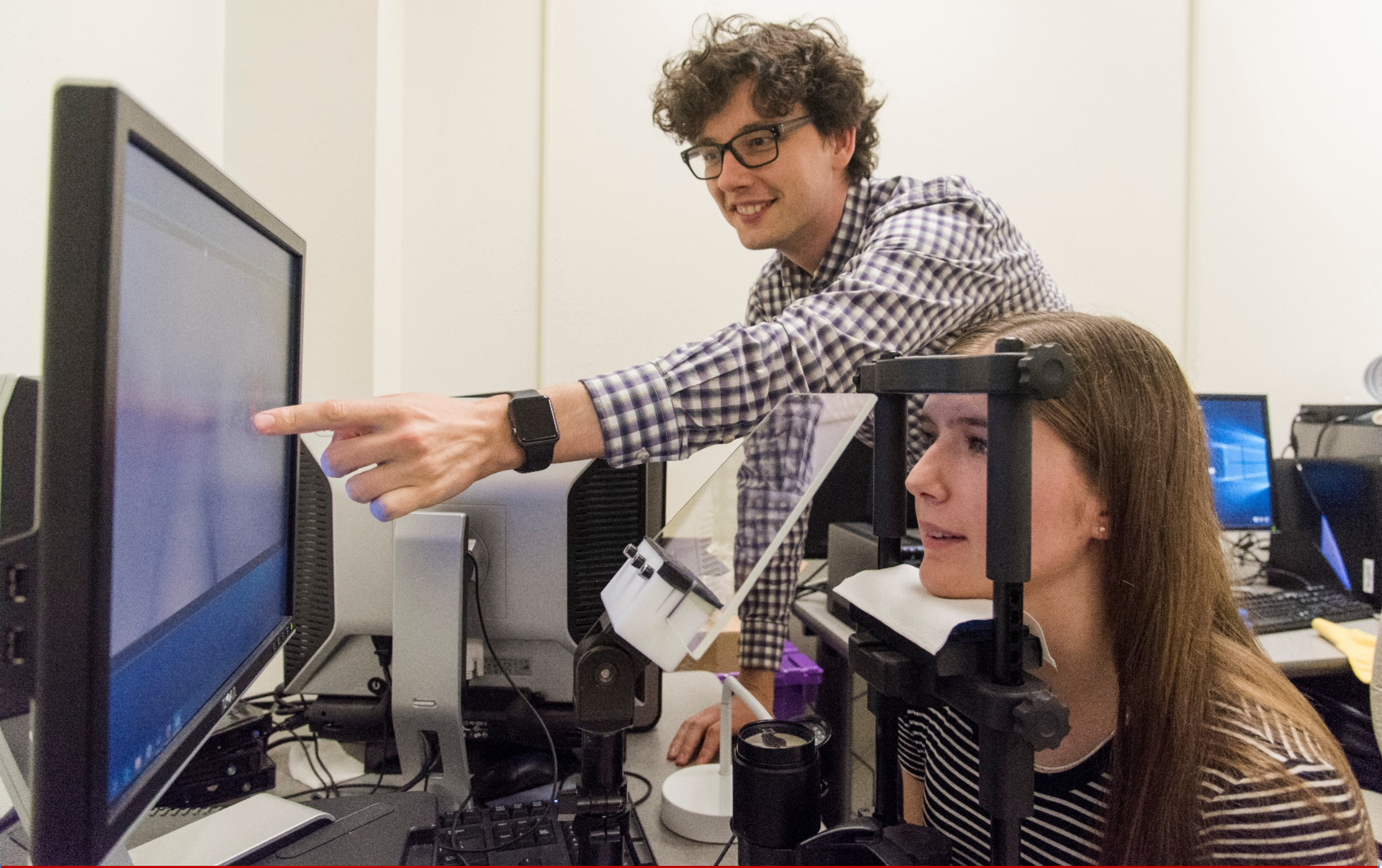
TEXAS STATE

LBJ INSTITUTE FOR
STEM EDUCATION AND RESEARCH

Redesigning Computer Science: An Early Intervention to Combat Undergraduate Student Misconceptions

By Laura Rodríguez Amaya, Ph.D., Mina Guirguis, Ph.D., and Jocabed Marquez, Ph.D.



Research Report

for NASA Educator Professional
Development Collaborative

Volume 1, Issue 2
June, 2019

Redesigning Computer Science: An Early Intervention to Combat Undergraduate Student Misconceptions

Research regarding student perceptions of Computer Science as a field of study, and their motivation to pursue such studies as a career opportunity, reveal misconceptions and lack of motivation. These misconceptions are thought to impact students' understanding of the discipline and lead to the decline in enrollment in Computer Science degrees. This study explores the impact of redesign in introductory computer science courses on student understanding of computer science. An introductory Computer Science course was redesigned by adopting course modules containing paradigms shifts that gave students a holistic view of the field of computer science. Student responses from a pre and post survey after the intervention were analyzed to assess if there were any changes in students' understanding of computer science after the implementation of the new modules in the computer science foundation class. Findings show a positive impact on students' understanding of computer science.

Executive Summary

Some students report regarding the study of computer science as narrowly equivalent to "programming" while others indicate the lack of connection to other domains or real problems (Allan & Kolesar, 1997). The goal of this study is to explore the impact of redesign in introductory computer science courses on student understanding of computer science.

- A total of 47 students participated in the CSPS pre-survey and 26 students in the post-survey. Pre-survey responses show few significant differences in students' understanding of computer science among Computer Science majors and other majors. These results appear to indicate that students that took the foundational computer science course regardless of their major shared similar perceptions on the relevance of computer science to daily life at the beginning and end of the course.

- According to the pre-survey, 62% of 47 respondents either "strongly agreed" or "agreed" when asked the question if they had a *clear understanding of what computers science is.* However, when asked to define computer science, their responses did not support their perception of having a clear understanding.

- In comparison with the pre-survey data, the post-survey data shows the emergence of more accurate definitions of computer science that included themes such as real-world applications. This result appears to indicate a positive change in students' understanding of *what computer science is*.

- Overall, data appears to indicate that the modules had a positive effect in students having a better understanding of computer science. A lower percentage of students reported in the post-survey that there were no differences between computer science and programming.

Introduction

The President's Council of Advisors on Science and Technology's (PCAST) report predicts that the U.S. workforce's supply will be one million short of the demand for graduates in science, technology, engineering and mathematics (STEM), but less than half of those who enter U.S. colleges to pursue majors in STEM persist to graduation (Executive Office of the President, 2012). However, to remain or become globally competitive, countries recognize that they need a well-prepared and diverse STEM professional pipeline. To this end, many countries have developed strategic national science, technology, engineering, and mathematics (STEM) policy frameworks. In 2017, President Obama announced the *Computer Science for All* initiative to give students across the country the chance to learn computer science. This program addressed the concern about high attrition among computer science college students and the need for a workforce with strong computational thinking skills in today's digital economy (Ferrini-Mundy, Kurose, & Garg, 2016). The world's economy is rapidly shifting, and governments, educators and business leaders are increasingly recognizing that computer science is a "new basic" skill necessary for economic opportunity and social mobility (Ortiz & Guirguis, 2016).

Research regarding student perceptions of Computer Science as a field of study and their motivation to pursue such studies as a career opportunity reveal misconceptions and lack of motivation (Beaubouef & McDowell, 2008; Owens & Matthews, 2008; Rafieymehr, 2008; Sloan & Troy, 2008; Van Sickle, 2008; Ortiz & Guirguis, 2016). These misconceptions are thought to impact students' understanding of the discipline and lead to the decline in enrollment in Computer Science degrees. Some students report regarding the study of computer science as narrowly equivalent to "programming" while others indicate the lack of connection to other domains or real problems (Allan & Kolesar, 1997). Moreover, many are not consistently provided the opportunity to realize the true impact of the field within their entry-level courses. Traditionally, students do not get a more holistic and broader understanding of computer science until their upper level courses. This again is thought to lead to student attrition. It is hypothesized that this lack of clarity of the field at an early point in students’ academic career, coupled by perceiving the curriculum as lacking relevance, has impacted the retention rates for computer science majors in their first two years of their programs (Ortiz & Guirguis, 2016).

The goal of this study is to explore the impact of redesign in introductory computer science courses on student understanding of computer science. An introductory Computer Science course was redesigned by adopting course modules containing paradigms focused on providing the students with the computer science big picture, familiarizing students with computational thinking, data-driven analysis, and multidisciplinary approaches. The research questions guiding this study is:

1). Do students that take a fundamental course in computer science have a clear understanding of:
   a. what computer science is?
   b. the difference between computer science and programming?

2). Do the Fundamentals of Computer Science redesign course impact students’ understanding of computer science?

Background

Research in academia and industry continues to identify a decline in enrollment in the field of computer science (Maddrey, 2011). One major component of this decline in enrollment is a shortage of female students (Kisselburgh, Berkelaar, & Buzzanell, 2009; Norris, Barry, Fenwick, Reid & Rountree, 2008; Powell, 2008; Moorman & Johnson, 2003; Rosser, 2005). Some of the reasons for the decline in enrollment presented in the research include misconceptions about the field (Beaubouef & McDowell, 2008; Owens & Matthews, 2008; Rafieymehr, 2008, Sands et al., 2008; Sloan & Troy, 2008; Van Sickle, 2008), negative cultural stereotypes (Adya, 2008; Barker 2009; Edmondson, 2008; Sands et al. 2008; Van Sickle, 2008) and lack of computer experience prior to the first year in college (Barker et al., 2009; Powell, 2008; Van Sickle, 2008).

Another common misconception of computing is that it is unrelated to interesting problems in other domains. Current educational practices tend to support this misconception by focusing solely on programming languages and software development in introductory classes (Hart, Ridley, Taher, Sas, Dix, 2008; Hazzan, Gal-Ezer & Blum, 2008). Allan and Kolesar (1997) note that students in introductory computer science courses indicate a singular focus on successful completion of the current assignment with no thought toward the larger picture of how skills learned will apply in various domains or even to other problems within computer science. A variety of research, discussed next, calls for curriculum changes to combat this aspect of the discipline's image. Tang, Pan, and Newmeyer (2008) note that high school girls show considerably more interest and self-efficacy in careers and subjects involving working with and helping people. Rao (2006) noted an increase in performance among female computing students when assigned tasks in a domain of interest. Wilson (2006) also noted that female students were more likely to be interested in computer use when the problems addressed helped serve society as opposed to computer use simply for the sake of using a computer or to discover how it functions.

Klawe and Shneiderman (2005) discuss the importance of a shift in the overall curriculum of computer science to address the use of computers to solve societal problems. Baker, Krause, Yasar, Roberts, and Robinson-Kurpius (2007) state that a perceived lack of societal relevance keeps many students from entering science and engineering majors. In addition, Rao (2006) emphasized a need for computer science educators to shift techniques to focus on application in areas of interest rather than theory. Current standards of education involve creation of operating systems and analysis of algorithms in a purely computer science oriented context, devoid of real world application. In addition to helping computer science remain relevant, a shift toward teaching computer science as a mechanism for real world problem-solving may help draw and maintain interest in the subject.

Methodology

The study took place at a four-year institution. The Computer Science Perception Survey (CSPS) was adapted from the Computer Science Attitude Survey (Weibe, Williams, Yang, & Miller, 2003) and administered in the Fundamentals of Computer Science participating classes. Administration of the pre-CSPS happened at the beginning of the semester before the intervention. The intervention consisted of redesigning the Fundamentals of Computer

Science course by integrating newly developed modules. Previous work report that students who leave computer science often lack a good understanding of what computer science is (Biggers, Brauer, & Yilmaz, 2008). Students' exposure to the field early in their academic careers through the introductory courses focuses on programming and syntax. By mainly focusing on programming and projects that do not inform students of the big picture of computer science and how this content is relevant to their lives the misunderstanding that computer science is only programming is perpetuated.

The redesign courses incorporated new modules developed to address the misconception that computer science is just programming. One of the modules developed introduces computer science as a field by showing how computer science can be studied without even using a computer. Other modules focus on how computer science content is relevant to the real world; how modern social infrastructure is now more connected than ever with the use of information technology; and how computer science is relevant to other fields such as biology, social networking and big data analytics. Certain activities, following the presentation of the modules, are conducted in class in supporting the idea of computer science as a problem-solving field including a class discussion. Class discussions are geared towards the role of computer as a discipline for real world problem-solving. After the presentation of the modules students in these participating classes were invited to complete a post-survey.

Sample Population

In this study, the population of interest is undergraduate students taking the Fundamentals of Computer Science class. The classrooms selected were those classes whose faculty volunteered to incorporate the new modules. Students were invited to participate by completing a pre and post online survey after the modules were introduced. Of the students that participated in the pre-CSPS 36% were Computer Science majors and 66% percent were in their first two years of their academic program. Unexpectedly, the gender distribution of student participants shows a bigger than expected women participation at a 36%. Refer to Table 1 and 2 for more detailed information. As expected, post survey demographics show similar results.

Table 1: Demographics: Major and Classification

	Major			Classification						
	Other Major	CS	Total	Freshman	Sophomore	Junior	Senior	Other	Total	Missing
Frequency	30	17	47	9	20	9	5	1	44.0	3
Percent	63.8	36.2	100	19.1	42.6	19.1	10.6	2.1	93.6	6.4
Valid Percent	63.8	36.2	100	20.5	45.5	20.5	11.4	2.3	100.0	
Cumulative Percent	63.8	100		20.5	65.9	86.4	97.7	100.0		

Table 2: Demographics: Gender

	Gender					
	Male	Female	Gender Non-conforming	Total	Non-answer	Total
Frequency	27	16	1	44	3	47
Percent	57.4	34.0	2.1	93.6	6.4	100.0
Valid Percent	61.4	36.4	2.3	100.0		
Cumulative Percent	61.4	97.7	100.0			

Pilot Study

The pilot phase of this study served to assess the effectiveness of the CSPS to measure key perception constructs of believes and attitudes of students towards computer science.

The Computer Science Perception Survey (CSPS), a 35-item instrument was adapted from the Computer Science Attitude Survey (Weibe et al., 2003) and piloted among four instructors teaching the Fundamentals of Computer Science course during two academic semester sessions. The 35 questions were on a 5 point Likert scale from 1-Strongly Agree to 5-Completely Disagree and administered as a pre and post survey. A total of 50 students participated in the pilot CSPS survey.

Data Analysis

To analyze the pilot phase data of the CSPS a Confirmatory Factor Analysis (CFA) was run to confirm the constructs being studied. In CFA, the factors influence the observed variables to account for their variation and covariation. Covariation between two observed variables occurs when both variables are influenced by the same factor.

Conducting CFA on the Statistical Package for the Social Science (SPSS) most commonly uses Pearson-product-moment correlations as a method for fitting models to the bivariate associations among a set of variables. Using the product-moment correlations or covariances follows from the fact that the common factor model specifies a linear relationship between the factor and the observed variables. When a researcher conducts CFA, attempts are made to understand why the variables are correlated, and the degree or level of accuracy the variables and factors provide relative to theory. Factor analytic theory posits that variables correlate because they are determined in part from common but unobserved influences (Price, 2016).

First, data from the pilot study was analyzed to determine its suitability for factor analysis. The .830 value obtained for the KMO test showed that the data was suitable for this analysis. A minimum value of 0.6 is recommended for factor analysis (Tabachnick & Fidell, 2012). The Kaiser's measure of sampling adequacy is a ratio of the sum of squared correlations to the sum of squared correlations plus the sum of squared partial correlations (Tabachnick & Fidell, 2012). Second, a factor analysis in SPSS was constrained to five factors, which is considered appropriate for only 35 items with an average of 3-5 items per factor (Price, 2016). Below in Table 3 is the Total Variance Explained when the data was constrained to five factors. This model accounted for 47% of the variance in the 35 survey items.

Table 3: Total Variance Explained

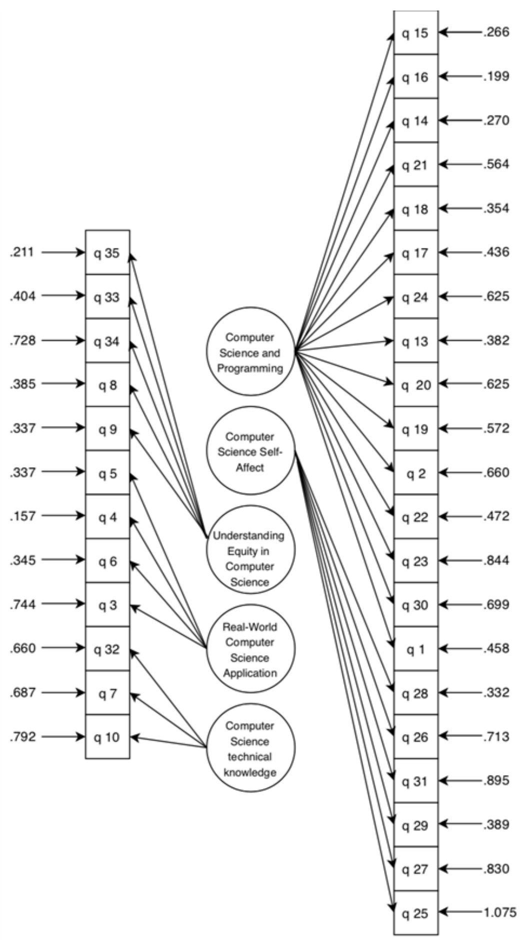
Total Variance Explained							
Factor	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings[a]
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total
1	9.309	26.596	26.596	8.940	25.542	25.542	8.749
2	3.042	8.690	35.286	2.508	7.166	32.708	2.591
3	2.914	8.326	43.613	2.365	6.756	39.465	2.627
4	2.106	6.017	49.629	1.575	4.499	43.964	3.293
5	1.617	4.620	54.250	1.015	2.900	46.864	2.167

The third step taken with the pilot data was to explore the five factors through a CFA. Since SPSS does not offer this option, MPlus was used to run the CFA. Figure 1 shows the CFA model.

The findings of the pilot study informed the focus of this exploratory study. This study focuses on Factor 1 construct of "Computer Science and Programming".

The confirmatory factor model in Figure 1 shows the five latent constructs revealed through this model. Table 4 shows the revealed constructs and the survey items that align with the construct. Refer to Appendix A for a copy of the CSPS pilot survey.

Figure 1. Five-factor model with corresponding survey items and their respective correlations.



Exploratory Study

After data from the pilot study was analyzed, the second phase of the study was to redesign the survey based on the findings. First, some questions needed better structure and language clarification. Absolutes such as (never, always,) are usually not recommended for Likert scale items, since both words on the item and answer choice may contain the exact same wording type (i.e. strongly disagree, never, strongly agree, always). This absolute wording was removed. Questions 9, 11 and 12 did not analytically load on any of the factors. They were problematic and removed. Refer to Appendix A and B for a copy of the pilot and final CSPS instruments used for this study. Two open-ended questions were added to inform the quantitative results. It was also deemed important to add 11 demographic questions to the final instrument.

Data Collection and Analysis

The final CSPS resulted in a 33-item instrument. Administration of the pre and post survey took place in one academic semester in two course sessions of the Fundamentals of Computer Science class. All instructors teaching the introductory course received an invitation to teach the new modules. Only students in the course sessions teaching the re-design courses were invited to participate in the online pre and post CSPS surveys.

This exploratory study followed a mix-methods approach. The sampling method was the same as in the pilot study. Given the low number of pre and post surveys that could be paired for reliable quantitative analysis, data analysis focused on deeper analysis of the qualitative data. Administration of surveys followed the same protocols than in the pilot study. The pre-CSPS administration occurred at the beginning of the semester prior to the introduction of the new modules. Students took the post-CSPS after the presentation of the new modules. Student responses from the pre and the post surveys were analyzed to assess if there were any changes in students' understanding of computer science after the implementation of the new modules in the computer science foundation class. The analysis of participant qualitative responses employed an inductive approach (Patton, 2015) and utilized Saldaña's (2016) first and second coding method. The first phase of data analysis consisted of coding the data to highlight key responses by the participants. This approach to coding was employed to enhance the voices of the students and offer a deeper understanding about changes in students' perceptions of computer science (Saldaña, 2016). Specific words/phrases used by survey respondents were coded and tallied manually. The second level coding stage utilized a thematic coding process that combined codes to identify patterns and create emerging categories and themes. Basic descriptive analysis of the quantitative data allowed for a means to identify any gaps in congruence among students' quantitative and qualitative responses.

FINDINGS

This study was conducted throughout a period of two years. The Computer Science and Perceptions Survey tool seeks to measure the impact of classroom interventions that address misconceptions of computer science. The new modules in the redesign course exposed students to the multiple applications of computer science to their daily lives. The aim of these models is to highlight the difference between Computer Science and programming, to show the relevance of Computer Science in recent advances in various fields, and to inspire students to appreciate Computer Science and the role of algorithms in our daily lives. The modules cover various topics about the role of Computer Science in cyber warfare, understanding biology, electronic voting, etc. Guiding this exploratory study are the research questions:

1). Do students that take a fundamental course in computer science have a clear understanding of:

   a. what computer science is?

   b. the difference between computer science and programming?

2). Do the Fundamentals of Computer Science redesign course impact students' understanding of computer science?

The CSPS pre and post survey data for the construct of *Computer Science and Programming* was analyzed to explore if there were any changes in students' understanding of computer science after the implementation of the new modules in the Fundamentals of Computer Science course. This section will explore the results of the selected survey questions for this study that attempts to answer the research questions. The selected survey items are Q1, Q2, and Q5:

Table 5: Study Selected Survey Items

<table>
<tr><td colspan="5">Q1. To what extent do you agree or disagree with the following statements about Computer Science: [Five point Likert scale from Strongly agree to Strongly disagree]</td></tr>
<tr><td>1. Computer Science can help us understand Biology</td><td>2. The field of Computer Science plays an important role in the safety of our nation</td><td>3. Computer Science can save lives</td><td>4. Computer Science empowers our citizens</td><td>5. I have a clear understanding of what computer science is.</td></tr>
<tr><td colspan="5">Q2. Define Computer Science</td></tr>
<tr><td colspan="5">Q5. According to your understanding, are there any differences between Computer Science and Programming?</td></tr>
</table>

Student Understanding of Computer Science

A total of 47 students participated in the CSPS pre-survey and 26 students in the post-survey. Pre-survey responses show few significant differences in students' understanding of computer science among Computer Science majors and other majors. Table 5 show the set of Likert scale questions used to assess students' understanding of computer science. When comparing Q1_1 through Q1_5 pre-survey responses of computer science majors to other majors, most results yield percentage differences of less than 10% with the exception of one question. Question 1_3 asked students to rate on a 5 point Likert scale of "strongly agree" through "strongly disagree" the statement *Computer science can save lives.* In this question, 83% of 30 non-computer science majors stated either "strongly agree" or "agree" compared with 100% of 17 computer science majors. Refer to Table 6 for more detailed information.

Table 6: Q1 Pre-CSPS Frequency Analysis

	Pre-CSPS				
	Biology	**Safety of Nation**	**Safe Lives**	**Empowers Citizens**	**Clear Understanding**
Strongly Agree	14.9%	61.7%	38.3%	29.8%	29.8%
Agree	21.3%	34.0%	51.1%	57.4%	31.9%
Neutral	42.6%	4.3%	6.4%	10.6%	29.8%
Disagree	14.9%	0.0%	0%	0%	6.4%
Strongly Disagree	6.4%	0.0%	4.3%	2.1%	2.1%
Total	100%	100%	100%	100%	100%

Table 7: Q1 Post-CSPD Frequency Analysis

	Post-CSPS				
	Biology	**Safety of Nation**	**Safe Lives**	**Empowers Citizens**	**Clear Understanding**
Strongly Agree	26.9%	84.6%	80.8%	53.8%	42.3%
Agree	42.3%	15.4%	19.2%	38.5%	46.2%
Neutral	15.4%	0.0%	0.0%	7.7%	11.5%
Disagree	15.4%	0.0%	0.0%	0.0%	0.0%
Strongly Disagree	0.0%	0.0%	0.0%	0.0%	0.0%
Total	100%	100%	100%	100%	100%

When looking at the post-survey results the only question that yield results with a percentage difference of 10% points or more was question Q1_1 *computers science can help us understand biology*. Of the 18 non-computer science majors, 72% “strongly agree” or “agree” with the statement in comparison to 62% of the computer science majors. In addition, this question was the only one in the post-survey that yield results in the “strongly disagree” or “disagree” point scales. All other questions yield responses in the “strongly agree”, “agree” or “neutral” point scales. These results might indicate a higher awareness level of students in other majors (not computer science) of the relevance of computer science to other disciplines as their own.

The above results appear to indicate that students that took the foundational computer science course regardless of their major shared similar perceptions on the relevance of computer science to daily life at the beginning and end of the course. Students’ self-responses in the pre-survey reflect a good understanding of computer science relevance to their daily life (Q1_2 through Q1_4), with at least 87% of students agreeing to the statements that computer science is important to national security, the empowerment of citizens and saving lives.

When looking at the post-survey responses on these same questions, there were still considerable gains on students’ awareness on how computer science connects with their lives as shown in Table 6. Making connections of computer science with other disciplines such biology, however, challenged students with only 36% of students participating in the pre-survey being able to make this connection (Q1_1). After the modules, however, the post-survey data shows twice as many students making this connection (refer to Table 6).

Defining Computer Science

According to the pre-survey, 62% of 47 respondents either “strongly agreed” or “agreed” when asked the question if they had a *clear understanding of what computers science is* (Q1_5 in Table 6). However when asked to define computer science, their responses did not support their perception of having a clear understanding. Students equated programming with computer science stating that computer science is “programming computers”, “the study of computers and how they work with algorithms”, and “computer [science] is the understanding of how computers work”. Out of the 29 students reporting a strong understanding of what computer science is, only 21% made reference to either a tool to solve problems or relevant to daily life.

Data analysis from the pre-survey qualitative question *Define Computer Science* yield three recurring themes: 1) Programming; 2) Computer Systems; 3) and Human Life Applications. Some of the definitions presented for computer science included code wording for more than one theme, and therefore, included in multiple themes. Below is the description and data analysis results for each of the themes identified.

Programming

Out of 47 students 34% used the terms, "coding, programming", "creating codes", "knowledge of networking and programming", "program functions", "code and computing", "knowledge of how to program", and "computer instructions," to define computer science. Here are some examples of students' responses in their own words:

- *Coding and programming*
- *Programming computer.*
- *The profession of understanding software design and program functions. (The brains behind the machine)*

Computer Systems

Another common definition found in 57% of responses included references to the inner working of computers. Some of the terms used throughout these responses included "understanding how computers work", "study of computers", "how computers can be used", "science behind computers", "science of computer hardware", and "uses and reasoning behind a computer". Here are some examples of students' responses in their own words:

- *The science and mechanics behind computers*
- *The science of computer hardware, software and its applications.*
- *The study of the inner workings of computers and their programs.*

Human Life Applications

Coding for this theme yielded 32% of the responses out of the 47. In this theme, references to one's life or human needs were prevalent. Here are some examples of students' responses in their own voice:

- *The science behind computer based methods and ideals that impact human life.*

- *Computer science is the use of code and computing to better understand logic and to improve and ease life for humanity.*
- *Computer science is manipulating software to inspire, create and develop one's life.*

Post-Survey Qualitative Response Coding

As stated in the methods section, after the introduction of the new designed modules, students in the selected Fundamentals of Computer Science classes took a post-survey. According to the post-survey, 88% of respondents, 23 out of 26 students, either "strongly agreed" or "agree" when asked the question if they had a clear understanding of what computers science is. Compared to 62% of respondents in the pre-survey. Participants were then asked to define computer science in an open-ended question.

In comparison with the pre-survey data, the post-survey data shows the emergence of two new themes when asked to define computer science: real world applications and algorithms. Responses were coded in the *Real-World Applications* theme when respondent associated computer science with tangible applications to their lives or the lives of others. The *Algorithms* theme was more straightforward with the inclusion of responses that contained the word algorithm. Although, *algorithm* appeared in 11% of the pre-survey responses, in the post-survey it appeared in 27% of the responses so it was deemed important to include it as a new theme. Responses coded as *how computers work* and *problem solving* were also present in the post-survey but not include in this discussion since student comments were very similar to the pre-survey responses. The post-survey themes selected for further discussion are those that represent a new theme in the post-survey responses or reflect a significant change (more than 10% difference) from the pre-survey results.

Programming

The data shows that only one student responded to the question of *Define Computer Science* with "ability to program" or any other response equating computer science with programming. This appears to indicate that after integrating the new module in the class curriculum students increased their understanding on the distinction between computer science and programming. As part of this early intervention model, students engaged in a discussion on what computer science is and it is not. Emphasizing the differences between computer science and programming.

Real-World Applications

Coded in this theme are 23% of 26 students' responses associated with real world applications, relevance, solving today's world issues/problems, and the applicability to everyday life. Here are some examples of students' responses in their own words:

- *It is not only useful in STEM fields but in every job and will continue to be incorporated farther into this technological society.*
- *The study of software and hardware solutions to process, transmit and store data in addition to controlling technology in all aspects of everyday work and home life.*
- *The theoretical and practical uses of computers and the application of these to our world.*

Algorithm

The pre-survey and the post-survey student responses in this theme are very similar. The responses that include the word algorithm associate the word with "*the study of..*" *as in* "the study of computers and algorithms". The main difference between the pre and post surveys in this theme is the frequency of the word algorithm included in student definitions of computer science. *Algorithm* appeared in 11% of the pre-survey responses compared to 27% of the post-survey responses. Below are some examples.

- *A field of study that analyzes and creates systems of logic and algorithms, in the physical and digital world.*
- *Study of algorithms and the fundamental workings of how a computer operates.*
- *The study of computers and their uses, as well as algorithms.*

Reflected in the data is a shift on student understanding of what *computer science is* represented in the significant decrease of students equating computer science to programming and the new theme associating computer science with real-world applications. Maybe the most significant change is that only one student equated computer science to programming in the post-survey. In addition, students' definitions moved from abstract responses on the application of computer science to human life to an increase awareness of its role in other disciplines and aspects of life. These findings inform and support the quantitative results in Question 1_1 through

Q1-5. In these results there was an overall increase in students' understanding on the relevance of computer science to people's daily lives.

Student Understanding of the Differences Between Computer Science and Programming

It has been reported in previous studies that students who leave computer science often lack a clear idea of what computer science is (Biggers et al.,2008; Ortiz & Guirguis, 2016). Furthermore, students report that they regard the discipline of computer science as narrowly equivalent to "programming" (Ortiz & Guirguis, 2016). This study contributes to this research agenda by further exploring how students see the difference between computer science and programming, if any, before and after the intervention. It is important to have a clear understanding of the misconceptions students hold about computer science to effectively address them.

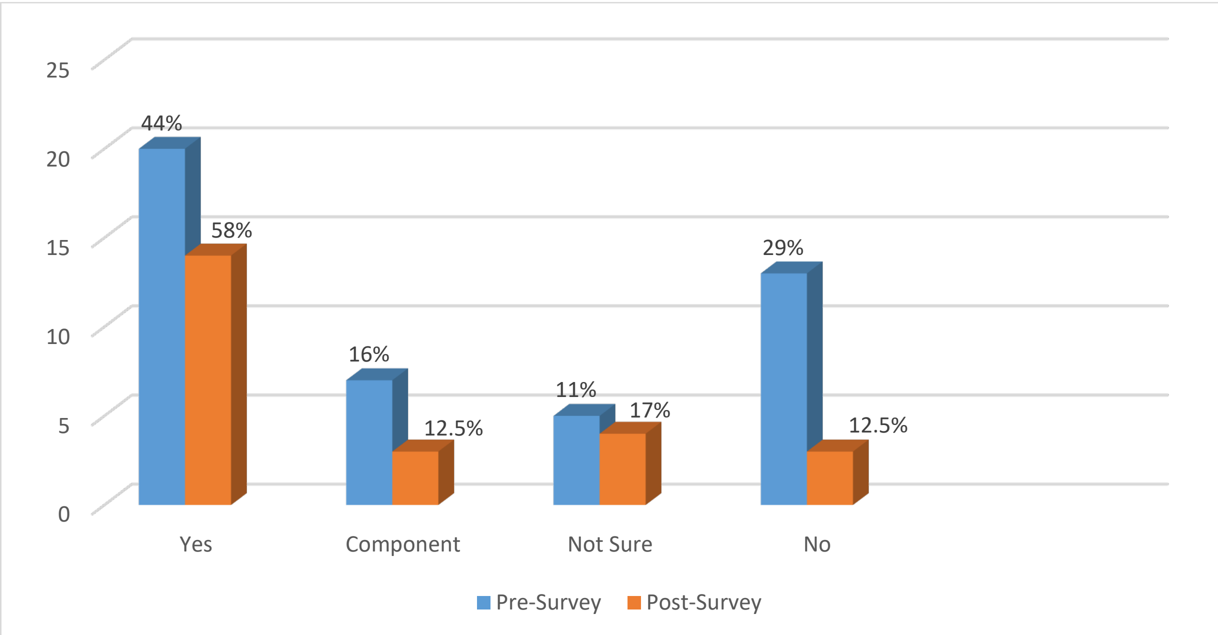
When comparing students' pre-survey and post-survey responses to Question 5: *According to your understanding, are there any differences between Computer Science and Programming?* the results show promising outcomes. Students responses were coded in four different themes: 1)Yes; 2) Component; 3) Not Sure; and 4) No. The *Yes* and *No* themes are straight forward. Answers that included programming as a "key" or "component" or any word that denoted *part of* were coded Component. Responses that included ambivalent language such as "not sure" and "don't know" were coded in the Not Sure theme. Out of 45 students that responded to Question 5 in the pre-survey, 40% reported no difference or unsure of any difference between computer science and programming. Refer to Figure 2. Below are some of the responses on students' own voice:

- *As of right now, I am not sure.*
- *Programming is major key to computer science as far as telling things what to do and how to do them . I don't really see any differences yet.*
- *I don't know.*

After students covered the new modules designed to address students' misconceptions of computer science, data shows a positive change in student understanding on the differences between computer science and programming. Out of the 24 students that answer Question 5, 30%

compared to 40% in the pre-survey stated "no" or were unsure of the differences. Furthermore, students that stated that there were differences between computer science and programming when up 14%.

Figure 2. Student Responses on Q5: According to your understanding, are there any differences between Computer Science and Programming?



It is important to mentioned that in both the pre and post surveys, a significant number of students (16% and 12% respectively) reported programming as a component or part of computer science with responses such as *"Programming is an element of Computer Science. So in a sense Computer Science is more inclusive of the entire field instead of just how to write and debug programs"*.

Limitations

Exploratory research explicit purpose is to discover phenomena, variables, theories, or combination to further a research agenda (DePoy & Gitlin, 2016). Given the exploratory nature of this study, caution should be taken in the interpretation of these study results, and limitations considered. One important limitation is that exploratory research results cannot be generalized to the larger population of interest, unless the quantitative data meets the conditions for such interpretation.

Our study sample population consisted of undergraduate students taking the Fundamentals of Computer Science course at a four-year institution. Given the low number of pre and post surveys that could be paired, the quantitative data analysis deemed appropriate for the data consisted of descriptive statistics. Notwithstanding these limitations, the results presented provide an important insight into the effectiveness of early interventions in addressing undergraduate students' misconceptions of computer science contributing to this important research agenda.

CONCLUSION

Reports on students regarding computer science as equivalent to programming and lack of connection to other domains date longer than 20 years (Allan & Kolesar, 1997) and have been predominant in the last 10 years (Beaubouef & McDowell, 2008; Owens & Matthews, 2008; Rafieymehr, 2008, Sands et al., 2008; Sloan & Troy, 2008; Van Sickle, 2008; Ortiz & Guirguis, 2016). Many Computer Science departments in institutions of higher education are trying new ways of combating misconceptions about the discipline that hinders the recruitment and retention of a diverse student population. Through the redesign of a foundation computer science course, an attempt was made to address students' misconceptions about the discipline. By challenging the traditional expository instructional methodology currently prevalent in these courses, students are provided with hands-on learning experiences that provide computer science "big picture".

The results of the study are promising. Overall, data appears to indicate that the modules had a positive effect in students having a better understanding of computer science. A lower percentage of students reported in the post-survey that there were no differences between computer science and programming. In addition, in the post-survey students' definitions of computer science are more specific and descriptive. Through these responses a new theme emerged indicating students' understanding of the relevance of computer science to other domains and the world around them.

Notwithstanding these promising results, challenges remain on identifying the most effective ways for students to learn early in their academic career what Computer

Science *is*. Especially when many foundational courses curriculum tend to focus on programming and writing applications. Future plans are to expand the sample size of the study to conduct more rigorous statistical analyzes. In addition, including a control group will assist in better assessing the effect of the modules.

One particular aspect worth pursuing is studying the impact of guest lecturers in introductory courses. Anecdotal evidence show that students would typically pay more attention to guest speakers (e.g., faculty pursuing research) who can give students an in-depth insight on a special topic – in particular topics that are usually outside the regular curriculum of the class. Identifying early opportunities to provide this type of learning experiences for undergraduate students in a systematic way might be worthwhile exploring.

REFERENCE

Adya, M. P. (2008). Women at work: Differences in IT career experiences and perceptions between South Asian and American women. *Human Resource Management*, 47(3), 601-635.

Allan, V. H., & Kolesar, M. V. (1997). Teaching Computer Science: A Problem Solving Approach that Works. *ACM SIGCUE Outlook*, *25*(1-2), 2-10.

Baker, D. R., Krause, S., Yaşar, Ş, Roberts, C. & Robinson-Kurpius, S. (2007). An Intervention to Address Gender Issues in a Course on Design, Engineering, and Technology for Science Educators. *Journal of Engineering Education,96*(3), 213-226.

Barker, V. (2009). Older Adolescents Motivations for Social Network Site Use: The Influence of Gender, Group Identity, and Collective Self-Esteem. *CyberPsychology & Behavior,* 12(2), 209-213.

Beaubouef, T., Lucas, R., & Howatt, J. (2001). The UNLOCK system: Enhancing Problem Solving Skills in CS-1 Students. *ACM SIGCSE Bulletin (inroads),* 33(2), 43-46.

Beaubouef, T., & McDowell, P. (2008). Computer Science: Student Myths and Misconceptions. *JCSC*, 23(6), 43-48.

Biggers, M., Brauer, A., and Yilmaz T. (2008). Student Perceptions of Computer Science: A Retention Study Comparing Graduating Seniors with CS Leavers. ACM SIGCSE Bulletin, 40(1), 402-406.

Brown, T. A. (2006). Confirmatory factor analysis for applied research. New York, NY : The Guilford Press.

Cohoon, J. M., Wu, Z., & Chao, J. (2009). *Sexism: Toxic to Women's Persistance in CSE Doctoral Programs.* Paper presented at the 40th SIGCSE Technical Symposium on Computer Science Education, Chattanooga, Tennessee.

DePoy, E., & Gitlin, L. N. (2016). *Introduction to Research understanding and applying multiple strategies*. St. Louis, MO: Elsevier.

Edmondson, A. C. (2008). The competitive imperative of learning. *Harvard Business Review*.

Executive Office of the President; President's Council of Advisors on Science and Technology. (2012). *Engage to Excel: Producing One Million Additional College Graduates with Degrees in Science, Technology, Engineering, and Mathematics.* Washington, DC.

Ferrini-Mundy, J., Kurose, J., & Garg, K. (2016, May 16). Computer Science is for All Students! Retrieved from https://obamawhitehouse.archives.gov/blog/2016/05/16/computer-science-all-students

Hart, J., Ridley, C., Taher, F., Sas, C., & Dix, A. (2008). Exploring the facebook experience. *Proceedings of the 5th Nordic Conference on Human-computer Interaction Building Bridges - NordiCHI 08*.

Hazzan, O., Gal-Ezer, J., & Blum, L. (2008). A model for high school computer science education. *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education - SIGCSE 08*.

Kisselburgh, L. G., Berkelaar, B. L., & Buzzanell, P. M. (2009). Discourse, Gender, and the Meaning of Work Rearticulating Science, Technology, and Engineering Careers Through Communicative Lenses. *Annals of the International Communication Association*, 33(1), 259-299.

Klawe, M., & Shneiderman, B. (2005). Crisis and opportunity in computer science. *Communications of the ACM,48*(11), 27.

Kline, R. B. (1998). Principles and practice of structural equation modeling. New York, NY: Guilford Press.

Maddrey, Elizabeth (2011). *The effect of problem solving instruction on the programming self-efficacy and achievement of introductory compter science students* (dissertation). Graduate School of Computer Information Sciences Noval Southeastern University.

Mertler, C., & Vannatta, R. (2013). *Advanced and Multivariate Statistical Methods, 5th ed*. Pyrczak Publishers.

Moorman, P., & Johnson, E. (2003). Still a Stranger Here: Attitudes Among Secondary School Students Towards Computer Science. *ACM SIGCSE Bulletin,35*(3), 193-197.

Norris, C., Barry, F., Fenwick Jr., J. B., Reid, K., & Rountree, J. (2008). ClockIt: Collecting quantitative data on how beginning software developers really work. *ACM SIGCSE Bulletin,40*(3), 37.

Olsen, A. L. (2005). Using Pseudocode to Teach Problem Solving. *JCSC*, 21(2), 231-236.

Ortiz, A., & Guirguis, M. (2016). Measuring the Impact of Unique Entry-Level Instructional Course Modules Designed to Inspire Computer Science Interest. *2016 ASEE Annual Conference & Exposition Proceedings*.

Patton, M. Q. (2015). *Qualitative research & evaluation methods: Integrating theory and practice*. Los Angeles: SAGE.

Paxton, J., & Mumey, B. (2001). Teaching advanced problem solving: Implications for the CS curriculum. *JCSC, 16*(2), 51-56.

Pollock, L., McCoy, K., Carberry, S., Hundigopal, N., &You, X. (2004). Increasing high school girls' self-confidence and awareness of CS through a positive summer experience. Proceedings of the ACM SIGCSE Technical Symposium on Computer Science Education '04, 185-189.

Powell, A., Bagilhole, & Dainty, A. (2009). How women engineers do and undo gender: consequences for gender equality. *Gender, Work and Organization*, 16(4), 411-428.

Price, L. R. (2016). Psychometric methods: Theory into practice. New York, NY: Guildford.

Pulimood, S. M., & Wolz, U. (2008). Problem solving in community. *ACM SIGCSE Bulletin, 40*(1), 210-214.

Rafieymehr, A. (2008). Kids in Computing (K.I.C.). *ACM SIGCSE Bulletin,40*(2), 107-111.

Ragonis, N., & Hazzan, O. (2008). Disciplinary-Pedagogical Teacher Preparation for Pre-service Computer Science Teachers: Rational and Implementation. *Lecture Notes in Computer Science Informatics Education - Supporting Computational Thinking,*253-264.

Rao Korukonda, A. (2006). Differences that do matter: a dialectic analysis of individual characteristics and personality dimensions contributing to computer anxiety. *Computers in Human Behavior* 23 1921-1942.

Rodríguez Amaya, Laura, Tania Betancourt, Kristina Collins, Orlando Hinojosa, and Carlos Corona. (2018). Undergraduate Research Experiences - Mentoring, Awareness, and Perceptions: A Case Study at a Hispanic Serving Institution. *International Journal of STEM Education* 5:9.

Rosser, S. V. (2005). Through the Lenses of Feminist Theory: Focus on Women and Information Technology. *Frontiers: A Journal of Women Studies,26*(1), 1-23.

Saldaña, J. (2016). *The coding manual for qualitative researchers*. Los Angeles, CA: Sage.

Sloan, R. H., & Troy, P. (2008). Cs 0.5: A Better Approach to Introductory Computer Science for Majors. *ACM SIGCSE Bulletin,40*(1).

Tabachnick, B. G. & Fidell, L. S. (2012). Using Multivariate Statistics, 6th ed. Pearson: Allyn & Bacon Publishers.

Tang, M., Pan, W., & Newmeyer, M. (2008). Factors Influencing High School Students Career Aspirations. *Professional School Counseling,11*(5), 285-295.

Weibe, E. N., Williams, L., Yang, K., & Miller, C. (2003). *Computer Science Attitude Survey*(Rep. No. NCSU CSC TR-2003-1). Raleigh, NC: Department of Computer Science NC State University.

Wilson, B. C. (2006). Gender differences in types of assignments preferred: implications for computer science instruction. *Journal Educational Computing Research*, 34(3) 245-255.