LOW OVERHEAD METHODS FOR IMPROVING EDUCATION CAPACITY AND OUTCOMES IN COMPUTER SCIENCE

by

RICHARD SCOTT BELL

B.S., Missouri University of Science and Technology, 1994
M.S., Missouri University of Science and Technology, 2000

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2014

ABSTRACT

Computer science departments face numerous challenges. Enrollment over the past 15 years reached an all-time high, endured a rapid decline and is now experiencing a just as rapid rebound. Meanwhile, demand for graduates continues to grow at an incredible rate. This is especially true in specialized sub-fields such as cybersecurity, where employers are constantly working to keep up with changing technology and new threats emerging on a daily basis. My research consists of two main objectives. The first is gauging the ability of pre-service teachers from non-STEM areas of study to introduce and utilize computing concepts in a classroom setting. The second goal is to develop an assessment tool that enables improvements in quality of education for students within cybersecurity courses.

Currently, few K-12 school districts in the United States offer stand-alone courses in computer science. My work shows that pre-service teachers in non-STEM areas are capable of effectively introducing basic concepts to students using modern software development tools while exploring content within their own areas of expertise. Survey results indicate that student interest and self-efficacy increased when they were taught by these pre-service teachers. I also found that with only 2 hours of experience, pre-service teachers enrolled in an education technology course showed dramatic increases in interest and confidence related to using this technology. These two findings demonstrate that there are potential ways to increase interest in computing among a broad student population at the K-12 level without changing core curriculum requirements.

Even when students choose to enter computer science departments, a large number do not remain within the program. The second portion of my research focuses on developing an assessment tool for measuring student interest and self-efficacy in cybersecurity courses. Using information gleaned from a series of interviews with cybersecurity students, I developed, and performed the initial testing of, a survey instrument which measures these 2

values. Initial results show that the survey responses were very different between a group of introductory programming students and those enrolled in a cybersecurity course and that general trends in both self-efficacy and interest among theses differing student populations can be observed

LOW OVERHEAD METHODS FOR IMPROVING EDUCATION CAPACITY AND OUTCOMES IN COMPUTER SCIENCE

by

RICHARD SCOTT BELL

B.S., Missouri University of Science and Technology, 1994
M.S., Missouri University of Science and Technology, 2000

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2014

Approved by:

Major Professor
Eugene Vasserman, PhD.

ABSTRACT

Computer science departments face numerous challenges. Enrollment over the past 15 years reached an all-time high, endured a rapid decline and is now experiencing a just as rapid rebound. Meanwhile, demand for graduates continues to grow at an incredible rate. This is especially true in specialized sub-fields such as cybersecurity, where employers are constantly working to keep up with changing technology and new threats emerging on a daily basis. My research consists of two main objectives. The first is gauging the ability of pre-service teachers from non-STEM areas of study to introduce and utilize computing concepts in a classroom setting. The second goal is to develop an assessment tool that enables improvements in quality of education for students within cybersecurity courses.

Currently, few K-12 school districts in the United States offer stand-alone courses in computer science. My work shows that pre-service teachers in non-STEM areas are capable of effectively introducing basic concepts to students using modern software development tools while exploring content within their own areas of expertise. Survey results indicate that student interest and self-efficacy increased when they were taught by these pre-service teachers. I also found that with only 2 hours of experience, pre-service teachers enrolled in an education technology course showed dramatic increases in interest and confidence related to using this technology. These two findings demonstrate that there are potential ways to increase interest in computing among a broad student population at the K-12 level without changing core curriculum requirements.

Even when students choose to enter computer science departments, a large number do not remain within the program. The second portion of my research focuses on developing an assessment tool for measuring student interest and self-efficacy in cybersecurity courses. Using information gleaned from a series of interviews with cybersecurity students, I developed, and performed the initial testing of, a survey instrument which measures these 2

values. Initial results show that the survey responses were very different between a group of introductory programming students and those enrolled in a cybersecurity course and that general trends in both self-efficacy and interest among theses differing student populations can be observed

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acknowledgments

First and foremost, a huge thank you to Tonya for being there every step of the way and for putting up with the crazy hours. I could not have done this without you in my corner.

I also want to give credit where it is due. To my parents, Greg and Karen Bell for the endless support and encouragement they have provided throughout my life.

Thank you to my adviser, Dr. Eugene Vasserman, for keeping me on task, and for letting me take this work and run with it. Thank you as well to Dr. Eleanor Sayre for providing her expertise and advice to get me back on track before I wandered too far off.

I would also like to thank the members of my committee for their insight and opinions when I needed guidance. Especially Dr. Simon Ou for letting me interview students from his course as part of my research. Thank you to Dr. Mitch Neilsen and Dr. Gurdip Singh for giving me the opportunity to attend Kansas State. It has been one of the best experiences of my life.

To my fellow graduate students, those who have already moved on as well as those still in the trenches: thank you for the shared support and friendships that will carry on beyond this place and point in our lives. Without you all it would have been a much less enjoyable and much more difficult journey.

Last, but not least, thank you to all of the faculty within the Department of Computing and Information Sciences at Kansas State University for making the castle a welcoming environment for all of us to learn.

Preface

The research described herein was conducted under the supervision of Dr. Eugene Vasserman in the Department of Computing and information Sciences at Kansas State University, between September 2011 and June 2014. This research is covered under IRB protocols 6793, 5867, and 6750.

To the best of my knowledge, this work is original, except where acknowledgments and references are made to previous work. Neither this, or any similar work has been or is being submitted for any other degree, diploma or other qualification at any other university.

Parts of this work have been presented in the following publications:

- Bell, S., Sayre, E., and Vasserman, E. *A Longitudinal Study of Students in an Introductory Cybersecurity Course* In *Proceedings of the 2014 ASEE Annual Conference*, Indianapolis, IN, June 2014.
- Bell, S., Frey, T., and Vasserman, E. *Spreading the Word: Introducing Pre-Service Teachers to Programming in the K-12 Classroom* In *Proceedings of the 45th Technical Symposium on Computer Science Education*, SIGCSE 2014.

Chapter 1

Introduction

Computer science (CS) is a relatively new field of study compared to other fields such as mathematics, biology, chemistry and physics. As such, the field is lagging behind these other areas in efforts to improve the quality of instruction and engagement of students in the classroom. Additionally, over the past 2 decades, departments have struggled to recruit enough students to keep up with the ever-growing demand for computing professionals.[1] Although enrollment is rebounding from the declines experienced during the early 2000s,[2] many students entering college today do not ever consider CS when selecting a major.[3,4]

Enrollment trends in CS departments nationwide have experienced dramatic shifts over the past 15 years.[2] After reaching a high point in 2001, a sharp decline began in 2002 and continued through 2007. Since that time, there has been a steady increase in enrollment numbers. However, over that same time span, the demand for computing professionals has continued to rise unchecked. The result is a deficit in the number of graduates that continues to grow. In fact, the Bureau of Labor Statistics indicates that by the year 2020, half of all STEM related jobs, and more than 60% of all *new* STEM positions will be in computing fields.[1] Their findings show that there are over 150,000 new job openings in computing per year. Computer science departments are not producing nearly that number of graduates each year. A National Science Foundation report shows that in 2006 there were

58,588 bachelor's degrees awarded in Mathematics and Computer Science nation-wide.[5] The National Center for Education Statistics reports the number for just computer science at 43,072 in 2010. Simply put: there is a need to find ways to increase the number of students that choose to enter CS departments.

For those students who do choose CS as their career path, there is a wide range of opinions on how and what should be taught. This can be readily observed in the sub-field of cybersecurity, where the range of approaches and content varies dramatically between universities.[6,7,8,9,10] Recent growth and changes in this field have occurred so rapidly that educators have not had the time to assess whether or not they are teaching this material in a manner that will promote student success in the classroom and encourage students to continue to pursue degrees and careers in the field.

I am approaching these issues in computer science education from two directions. I have demonstrated the effectiveness of a novel approach for increasing the introduction of CS topics into K-12 classrooms and helping to improve student exposure in CS concepts prior to college. I have also developed and performed initial testing of an assessment tool which will allow educators to evaluate the effects their pedagogical decisions are having within cybersecurity classrooms.

In both of these areas, I am focusing on student interest and self-efficacy. Self-efficacy is defined by Albert Bandura as: "the belief in ones capabilities to organize and execute the courses of action required to manage prospective situations." In other words, it is a person's confidence in his or her ability to perform a given task. I am focusing on these two areas because they have both been shown to improve students' persistence when faced with challenges, time on task, and selection of field of study.[11,12,13,14,15]

1.1 K-12 Education

1.1.1 Current Status

Unfortunately, many students never have any exposure to CS concepts at the K-12 level. Studies have shown that this lack of exposure and a variety of misconceptions which students have about CS are factors in students not choosing to pursue degrees in the field.[3,16,17] There are several reasons these issues exist. Computer science is currently not considered to be a core subject by many educational bodies and thus is not a high priority for school districts. This leads to a lack of positions for teachers with CS skills, which in turn leads to a lack of training for teachers either during their college education or as part of their continuing education after they enter the work force.[1] As an result, since 2005, the number of introductory CS courses offered at the high school level has decreased 17%.[18] Currently, of the just over 42,000 high schools in the nation, only 3249 offered an AP course in CS, with total enrollment of 31,117 students in 2013 compared to 8,444 which offered a course in Chemistry with enrollment of over 140,000.[19] The effect of this lack of support for CS can be seen nationwide. The percentage of high school students taking courses in various STEM areas has increased over the last 20 years for all disciplines except CS, where it has dropped from 25% to 19%.[20]

Opportunities for education majors to learn about CS topics do exist. For example, at Kansas State University, math and business teachers are required to take a programming course as part of their curriculum. This course is a version of the introductory programming course taught to computer science majors. While this can be useful material for teaching a programming or computer science course that is offered at the K-12 level, given that such courses currently are not offered by many school districts, the teachers who take this class will likely have little opportunity to apply the material from this course in their future teaching positions.

1.1.2 Attempts to Improve Computer Science Exposure Within K-12 Classrooms

There has been extensive effort put forth to provide in-service teachers with opportunities to learn about CS and to develop CS-related content.[21,22,23,24] While there has been measured success with such programs, there are several factors which limit the impact such approaches can achieve. First, not every school district will support a stand-alone computer science course.[18] Second, many of the teachers that attend these courses never use the material in their existing non-CS courses, even though faculty and staff at the university level expend time and resources to provide these opportunities. Finally, these approaches only connect with teachers who are *already* interested in CS. Unfortunately, based on my discussions with K-12 teachers, many suffer from the same misconceptions about CS that their students do, resulting in limited interest in these efforts from in-service teachers.

1.1.3 Approach

Taking advantage of an opportunity to work with pre-service teachers during a STEM summer camp, I first introduced these future teachers to computer programming and then helped them teach those topics to middle school students. Within a few weeks, pre-service teachers from the College of Education were able to successfully teach lessons involving basic programming constructs such as sequence, selection and iteration to middle school students.[25] More importantly, these teachers were able to do this while working with content and ideas from their own areas of expertise (Art and Music). My focus in this work was to observe how these pre-service teachers responded to learning and incorporating basic computer science concepts into topics related to their areas of interest and also assess student responses to the lessons presented by the pre-service teachers.

1.1.4 Introducing CS to More K-12 Teachers

My goal in working with pre-service teachers is to increase interest among a large population of K-12 teachers in using CS tools and concepts in their every-day classrooms. If this can be achieved and the teachers actually implement these ideas, then it is possible for them to teach computer science content as part of existing curriculum. For example, simulations can be used in various science courses such as Biology and Physics, students can build models to demonstrate social phenomena, and numerous development environments can be used for creative writing/thinking activities. I have been working with others to plan and hold workshops with Scratch for several years, with good attendance from existing K-12 instructors. Unfortunately, these programs only reach a small percentage of teachers. Using Scratch, I have shown that pre-service teachers quickly overcome fears of programming and survey responses show that they see potential ways to utilize such a tool within their future classrooms.

I believe that my results open the door for a new approach to CS outreach. This material can be introduced to teachers *while they are still in school*. In this environment, pre-service teachers have time to consider new ways to employ this technology within the K-12 curriculum. Additionally, it can be introduced *within existing courses for every pre-service teacher*. For example, there is already an educational technology course in most education programs and this material fits well within that context as the concept of such a course is to introduce ways that technology can improve or facilitate learning other subjects. In fact, I helped present Scratch to several sections of such a course during the Fall 2013 semester with excellent results. If successful, this approach can provide K-12 students with exposure to CS concepts within various academic contexts such as math, science, art, music and creative writing throughout their academic career. The results of my work in this area are discussed further in Chapter 3.

1.2 Cybersecurity Education at the University Level

1.2.1 Current Status

Within CS, the sub-field of cybersecurity has seen tremendous growth over the past 2 decades.[26] With the growth of the internet, the proliferation of network-capable computing devices, and the vast quantity of data being stored in digital formats, the need for professionals capable of securing communication channels and information storage has become a critical task for government entities, businesses, and individuals. This growth has occurred so rapidly that the academic pipeline has not been able to keep up.[27]

Examining the current status of cybersecurity education, it is evident that there are currently no definitive "best practices."[28] This problem is compounded by the variety of stakeholders attempting to address it.[29] Government and industry are trying to keep up internally by developing their own standards and training employees on systems that are continuing to change even as the training occurs. Meanwhile, colleges and universities are adding courses each year to help students gain an understanding of the problems they will face upon graduation. As with industry, these courses are continually updated to reflect the ongoing changes to both the technologies and the attacks which students will encounter upon entering the cybersecurity workspace. Additionally, there are certification companies that provide custom training courses at a variety of levels of expertise. Finally, professional organizations are attempting to develop curriculum standards which both provide guidance to academic institutions and meet the demands of industry and government employers.[30,31] My focus will be at the university level, where currently there are a wide variety of pedagogical methods, content and training environments being used, with little progress being made to verify what is truly working and what is not.[8,10,9,7]

Within academic institutions, the approach has been the introduction of one or more courses focused on cybersecurity. The depth and diversity of the content within such a course varies greatly among universities. These courses may look at the same topics in any of several different ways. For example, a course may investigate cryptography and the mathematical principles and algorithms used to protect data or might instead focus on system-level protocols.[6] While these two courses might cover the same ideas, the student perspectives and expected outcomes are drastically different. The first providing a theoretical understanding while the second provides practical, hands-on knowledge. The goals of courses vary as well, including teaching cybersecurity as practical vocation skills, as good engineering practices, or as academic theories. Finally, the methods used to teach these courses are just as varied as their goals and content.

1.2.2 Approach

The second part of my work focuses on developing a method of assessing student interest and self-efficacy outcomes from cybersecurity educational experiences. As was mentioned above, there is tremendous effort being put into developing training programs by various agencies, businesses and schools. However, there is little effort being made to determine if this is being done correctly. Over the course of two semesters, I spent time assessing both cybersecurity and introductory programming courses at Kansas State University. My goal was to collect information concerning student self-efficacy and interest related to cybersecurity concepts and develop an initial assessment tool measuring these factors which can be used to assess the outcomes of cybersecurity courses.[32]

Such an instrument will provide educators with a means of assessing students' perspectives of the effectiveness of the various pedagogical choices they make within their courses and thus allow them to make incremental and informed decisions in developing or changing courses in the future. Improving cybersecurity education in such an informed manner is critical to the long-term security of government, industry and individual computer systems. Current approaches, while necessary given the immediate needs of our field, do not provide for such long-term improvement.

Without such a tool, CS departments will continue to develop courses with no way to verify that they are doing so in a beneficial manner. With this in mind, and given the rapidly changing environment that is cybersecurity, my tool is focused on measuring student self-efficacy and interest in further work in cybersecurity. The results of my work in this area are presented in Chapters 4 and 5.

1.3 Contribution

In my work, I have shown that it is possible for pre-service teachers to gain interest and confidence in using CS concepts within a teaching environment in a very short period of time. I demonstrate that this can be at least partially achieved within an *existing* course offered by the College of Education at Kansas State University. With additional cooperative efforts between the Computing and Information Sciences and Education departments, this work can be extended to include classroom experience (both learning about programming and also applying programming concepts within lessons they are teaching) for most if not all education students. This provides an important step forward in the introduction of computing concepts into K-12 classrooms.

In addition to this work with K-12 teachers, I have developed and piloted a survey designed to be utilized as a pre- and post- instrument for measuring changes in student interest and self-efficacy levels within cybersecurity classrooms. This work has shown that differences can be measured between self-selected students enrolled in existing cybersecurity courses and those enrolled in a general introductory programming course. Due to limited enrollment in the courses which were available for assessment, there is additional work to be completed to validate this tool, but the initial results do show that effects can be measured using this instrument.

Chapter 2

Background

2.1 Job and Enrollment Trends in Computer Science

In 2013, the Computer Science Teacher's Association (CSTA) published a report that includes data from the Bureau of Labor Statistics predicting that, by the year 2020, 50% of all STEM related jobs will be in computing fields. Their data shows that the growth rate of jobs in computing fields is expected to exceed 150,000 per year over that time span.[1] While not all of these positions will be filled by CS graduates, the need for more students entering all fields of computing, including CS, is undeniable.

Meanwhile, CS undergraduate enrollment numbers are still recovering from a rapid decline in enrollment that occurred between 2001 and 2007. This decline and the subsequent, ongoing, recovery can be seen in the enrollment numbers displayed in figure 2.1. This graph is from the 2013 Computing Degree and Enrollment Trends report, an annual study conducted by the Computing Research Association to provide information on enrollment and graduation trends in computer science departments at PhD granting institutions in North America.[2] This data shows that undergraduate enrollment numbers at reporting institutions dropped approximately 50% from 2001 to 2007. By 2013, enrollment has nearly recovered.

The TauRUs report, a comparable study for non-PhD granting institutions, shows a

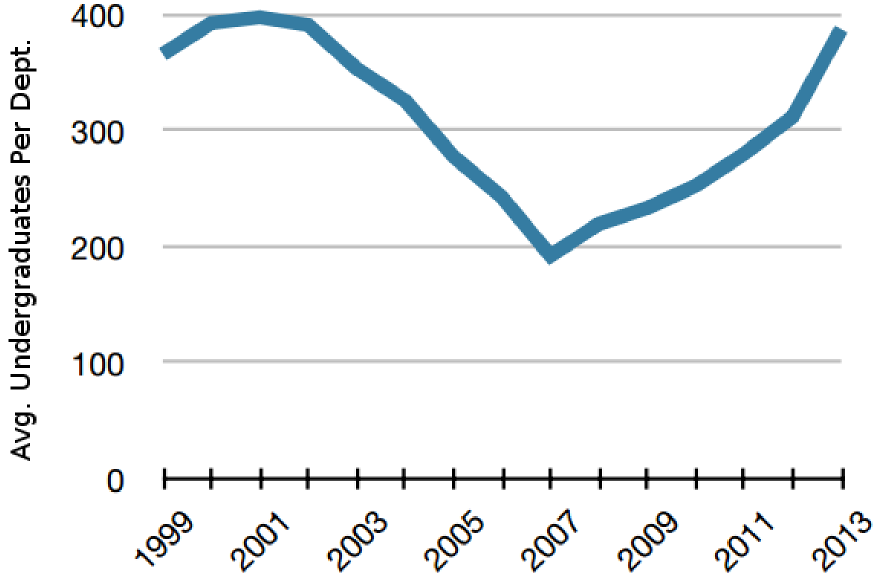

Figure 2.1: *Computing degree and enrollment trends shows the decline in enrollment in computer science at PhD granting institutions from 2001 - 2007 and the subsequent rebound. Source: The 2012-2013 CRA Computing Degree and Enrollment Trends Report*

similar pattern for its responding schools since 2006 (this study does not contain data from before 2006).[33] The results from both of these studies show that while enrollment numbers have been consistently rising over the past 6 years, they are only nearing recovery from the decline of the early 2000s.

The National Center for Education Statistics (NCES) tracks the number of undergraduate degrees conferred by post-secondary institutions within the United States each year in their Digest of Education Statistics.[34] The number of degrees conferred per year from 1971 through 2011 in computing and information sciences is shown in Figure 2.2. This graph shows that the recent decline in enrollment in CS departments is very similar to a decline which occurred in the late 1980s although the recovery pattern this time appears to be trending upward more quickly.

The graph also shows us that the recent surge in enrollment documented by both the Taulbee and TaURus surveys has not yet made it through to graduation, but the trends

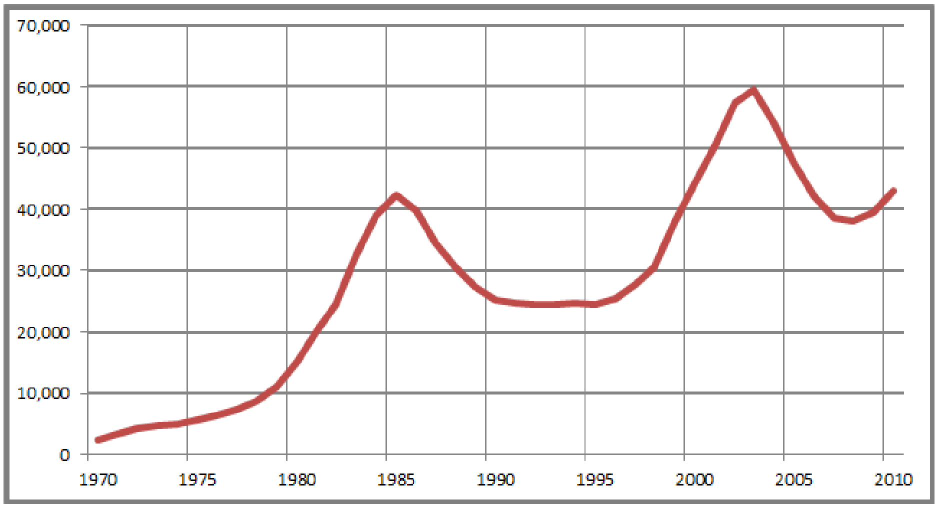

Figure 2.2: *Conferred undergraduate degrees in computer and information sciences per year. Source: National Center for Education Statistics 2012 Digest of Education Statistics*

within the NCES graph do reflect the beginning of the recovery seen in those reports. This graph is expected to mimic the continued upward trend of the enrollment data as the more recently enrolled students reach graduation. Even with this growth, the number of graduates (between 40,000 and 45,000) is still well behind the expected *growth* in demand for computing professionals reported by the CSTA study.

2.2 Computer Science in the K-12 Classroom

2.2.1 Introductory and AP Courses

As enrollment at the university level continues to lag behind the demand for CS graduates, support for CS at the K-12 level is struggling to gain momentum, and has in fact declined in some areas. CSTA reports that the number of introductory CS courses offered in high schools nationwide has decreased 17% since 2005.[18] A primary factor in this decrease is the fact that CS concepts are not considered part of the "core" curricular material in K-12 classrooms. Feeling pressure to limit costs and meet testing benchmarks, school districts are deciding not to offer CS courses. As a result, students have little, if any, exposure to computing concepts in their K-12 classrooms.

For comparison, data from the College Board's 2013 Advanced Placement (AP) exam shows that while the number of students taking the AP Chemistry exam nationwide was 140,006, the number of students taking the AP CS exam was only 31,117.[35] A somewhat positive note from this report was that the growth in CS from the 2012 exam (19%) was much larger than that of Chemistry (6%). These statistics do not show the number of students that are being taught computing concepts in K-12 classrooms since some schools may have non-AP computer science courses. However, they are an indication of the lack of learning opportunities available for students in CS vs other fields. This is yet another factor leading to the lack of interest students have in CS when looking to select a college major.

2.2.2 State Approaches

The CSTA's 2013 report provides insight into the current status of CS education in K-12 classrooms across the nation.[1] According to their research, there is great disparity among states in how (or even if) CS education is being incorporated into the K-12 curriculum. At the national level, CS is not yet recognized as a core subject. Additionally, states determine independently how (or if) teachers are certified to teach CS. The lack of curriculum recognition trickles down through state and local educational bodies, resulting in a reduced number of teaching positions and thus lowered demand for teachers specializing in CS. This lack of demand results in universities not placing a priority on preparing teachers who are capable of teaching this material. As an example, the CSTA study found that, in Florida, teachers who wish to be certified to teach CS are required to take a course that is not offered in any teacher preparation program in the state. So, even when states do recognize CS as a subject which teachers can be certified to teach, that does not necessarily indicate that the state is placing a priority on educating students in that field.

States also vary in how they view CS courses and how these courses apply towards high-school graduation. Only 14 states currently allow a CS course to be counted toward graduation requirements in mathematics, science or computer science.[1] A 2010 CSTA survey found that 35 states considered computer science to be a general elective, 8 considered it to be a mathematics elective, 1 considered it a science elective and 6 left the classification up to each school district.[18] Until this trend changes, school districts will be reluctant or unable to offer stand-alone CS courses and content. Alternative ways to incorporate these topics into the K-12 classroom need to be found.

2.2.3 Consequences

There is evidence that many of the primary factors leading to students not entering the computing fields relate to this lack of exposure to relevant CS topics in the K-12 classroom.[3,4] In their work, Cassel et al. list several factors that contribute to this problem, many of which could be alleviated by additional early introduction of CS concepts.[36] Their list includes several misconceptions that students hold about the field such as the perception of little human contact, boring work environment and limited career options for graduates with degrees in CS. These findings were supported by interviews of high school students performed by Yardi and Bruckman who found that teenagers viewed CS as being "boring, solitary and lacking real-world context."[16]

Grover et al.[37] show that even when schools do attempt to remedy the problem (such as introducing a programming course), there can still be misconceptions among students about computer science. However, the authors demonstrate that these misconceptions can, at least temporarily, be dispelled with activities such as guest speakers or even videos which directly address those misconceptions.

Such misconceptions about CS exist among university students as well. For example, Martin asked students in an introductory computer science course to draw an image of a computer scientist.[38] Based on the images drawn, students viewed computer scientists as white males with (as she describes it) "various degrees of geekiness." However, when she asked advanced undergraduate computer science students to do the same thing, they drew pictures of more "normal" individuals. In fact, all of those students reportedly drew an image similar to themselves.

2.2.4 Resolving the Issues

There are many potential approaches to changing the perceptions which younger students have about CS. Unfortunately, we are limited in the ways in which CS content can be introduced into the K-12 classroom. Programs such as No Child Left Behind have focused school districts on testing-based curriculum centered around math and reading and removed incentive to provide educational opportunities that do not support improvements in these testing results. Subjects such as computer science have been cut in order to focus resources elsewhere. Given the difficulties of changing core curriculum standards at the national or even state level, approaches which target individual teachers appear to hold the most short-term promise. In her paper titled "Computational Thinking," Wing identifies the need for the concept of computational thinking to be presented to a broader audience, including pre-college students.[39] She explains that these topics can be successfully introduced within a variety of courses which are not directly related to computing such as life sciences, and includes her belief that they should become as fundamental within education as reading, writing and arithmetic.[40]

This "branching out" of computer science into other fields can be seen in course listings at many universities. Courses such as computational chemistry (cheminformatics) and biology (bioinformatics)[41] are becoming more popular, with some schools even offering minors or whole degrees in these areas. Computing has also been integrated into other, non-scientific arenas such as music, videography and art at the university level.[42,43] However, this integration has not yet made its way into the majority of K-12 classrooms. Given the wide array of introductory software development tools which are available today, this idea is no longer out of the realm of possibilities.

2.2.5 STEM Summer Institute

The STEM Summer Institute is a summer enrichment program funded by a U.S. Department of Defense Education Activity grand, coordinated by the Manhattan-Ogden USD 383 School District and hosted by Kansas State University's College of Education. During the 4 week camp, nearly 200 sixth through ninth grade students are able to select a different course of study each week, with 12 courses being offered in 2013. Courses meet for 4 days, with students spending about 4 hours in the classroom each day. Over the 4 weeks, 69 students elected to take the programming course which I had the opportunity to help lead. The institute has two goals: 1) increase student achievement in STEM subjects within the school district by encouraging students to explore these fields through hands on activities, and 2) provide pre-service teachers with STEM teaching experience in a fun, energetic environment.

For secondary education majors at Kansas State University, teaching experience is broken into three "blocks." During the first of these blocks, pre-service teachers must participate in a minimum of 30 hours of field experience activities in a secondary education setting. The hours spent working with the camp fulfill this requirement. Many of the students who take the summer section teach 'specials' classes such as music, art, physical education, etc. which have limited opportunities for field experience in regular classrooms during the school year. I requested art and music teachers as there has been work showing that graphical programming environments can be effectively integrated into these content areas at the university level.[43,44,45] I also decided to use Scratch as the development environment for the camp. Scratch was created by the MIT Lifelong Kindergarten Group[46] and I have used it in a number of outreach programs and workshops with excellent results.

As part of the credit, pre-service teachers are responsible for developing lesson plans that fit the context of the course to which they are assigned. Additionally, during each successive week of the camp, the pre-service teachers take on more of the teaching responsibilities. By the fourth week, the pre-service teachers are responsible for a large percentage of the content being presented. Their lesson plans and teaching activities are reviewed and evaluated by an in-service teacher and an instructor from the College of Education, while content is evaluated by the in-service teacher running that particular course.

Each camp course is led by an in-service teacher selected from the school district based on performance and teaching abilities within a STEM discipline. A team of pre-service teachers (in our case, five) is paired with the in-service teacher to create the instructional team for the course. The pre-service teachers worked in 2 shifts, with 2 helping during the first 2 hours and the other 3 helping during the last 2 hours each day. This allowed them to also be enrolled in a Department of Education course running during the same 4 weeks. In several of the courses, this group works with a STEM department on campus to gain access to various resources not readily available to the school district (laboratory space, equipment, personnel, etc.). Since the in-service STEM teachers from the school district had limited experience teaching programming, I was given the opportunity to assist with designing and teaching the programming course. A more detailed description of the camp and the outcomes of this work is presented in Chapter 3.

2.3 Cybersecurity Education

Given the difficulties in recruiting students into computer science departments discussed above, departments should be doing everything within our power to provide a high quality, engaging experience for those students who choose to enroll in our classes. One area where this is especially needed is cybersecurity. Over the past 2 decades, the need for cybersecurity professionals has grown at a rapid pace. According to a report from Burning Glass Technologies, annual online postings for cybersecurity positions grew 74% from 2007 to 2013.[47] The U.S. Department of Labor predicts that the growth rate for information security analysts will be twice the rate of all computer occupations over the next decade and more than triple the growth rate of all occupations combined.[48] This growth in demand for cybersecurity graduates combined with the overall dip in CS enrollment numbers discussed in Section 2.1

has led to an employment void that universities and employers have been scrambling to fill.

2.3.1 Diverse Educational Experiences

Compared to other areas of study, cybersecurity is a very new field. Universities first began offering courses in cybersecurity during the 1970s.[29] Since that time, the importance and need for a workforce skilled in cybersecurity has grown rapidly.[26] Due to this rapid growth, and the breadth of content areas that fall under the umbrella of cybersecurity, a wide array of curriculum and pedagogical practices have been incorporated into today's cybersecurity classrooms.

While this diversity reflects the reality of cybersecurity education, it is a major hindrance to the development of a comprehensive model for cybersecurity education which would allow for consistent and continuous improvement. For example, knowledge areas which could be incorporated into cybersecurity include: computer architecture, criminology/law, cryptography, databases, human-computer interaction, information retrieval, information theory, management/business, mathematics, military science, mobile computing, networks, operating systems, digital forensics, philosophy/ethics, programming languages, software engineering, statistics/probability, and web programming.[7] Course focus may range from cryptography and the mathematical principles and algorithms used to protect data to system-level protocols.[6] Additionally, the expected outcomes from these courses may vary dramatically, with schools teaching cybersecurity as practical vocation skills, as good engineering practices, or as academic theories.

The pedagogical methods used to teach these courses are just as varied as the goals and content. Some courses focus on laboratory-based, experimental lessons.[8,9] Others are lecture-based and involve the review and discussion of literature, and still others are challenge based courses where instructors and students work together to solve problems.[10] This wide array of content and approaches shows how challenging it is to determine what might constitute the "best practices" in cybersecurity education.

Unfortunately, given this challenging environment, there is currently no valid way for those who are teaching cybersecurity courses to systematically measure and improve student outcomes within their classrooms. Adding to this problem is the fact that new technology and new vulnerabilities are thrown into the collective mix continuously, resulting in a constantly changing body of knowledge that must be incorporated into such courses. As part of my work, I am developing a tool which will allow cybersecurity educators to measure student interest and self-efficacy in relation to cybersecurity, providing a way to determine the effects that pedagogical decisions have on students in their classrooms.

2.3.2 Curriculum Standards Development

While the scope of the field is daunting, there is progress being made to improve the situation. In 2008, the ACM Special Interest Group for Information Technology (IT) Education (ACM-SIGITE) approved and published a model IT curriculum. Overarching all other pillars within the framework was information assurance and security (along with professionalism).[31] Similarly, in 2013, for the first time, the ACM and the IEEE have recognized information assurance and security as a separate knowledge area within their recommended Computer Science Curricula.[30] As with the IT curriculum, the CS curriculum incorporates components of cybersecurity throughout the various other computer science knowledge areas. While both guidelines provide recommended cybersecurity topics that should be covered within the respective curricula, they do not include any pedagogical best practices to guide instructors as to how they should engage students within the cybersecurity classroom or how to assess their own pedagogical methods.

2.4 Assessing Pedagogical Performance

There have been numerous research projects investigating ways to measure and improve student outcomes within introductory computer programming courses.[49,50,51,52] These types of courses have existed since the earliest days of computing, and while the details may have changed in that time, the core concepts and pedagogical approaches of these courses have become relatively stable from year to year and even from university to university. This allows researchers to develop surveys which measure student beliefs about Physics and learning Physics. Unfortunately, given the varied nature of cybersecurity courses, the approaches taken in these surveys do not readily apply.

Additionally, given the number of students enrolled in the introductory courses these surveys focus on (the introductory programming course in our program had over 110 students enrolled this past semester), researchers are able to conduct quantitative surveys and achieve statistically significant results over the course of a single semester as well as compare results with similar courses at various schools. Given the widely varying ways in which cybersecurity is currently being taught, the diverse content areas and the smaller number of students, I determined that while I wish to develop such a quantitative instrument for use in cybersecurity courses, I was not able to simply adapt the ideas used to develop the CLASS survey into a new survey for cybersecurity.

2.4.1 Qualitative Study

Cybersecurity Course Background

The initial target course being studied is a 3 credit-hour course offered once per year within the Computing and Information Sciences Department at Kansas State University. This is the 8th iteration of this course being taught by the same instructor. Enrollment ranges from 20-35 students per year. Students who enroll in the course are expected to have taken an operating systems or computer architecture course, or have comparable background. There are some computer engineering and information systems students who take the course. These students are expected to have a strong programming background although they likely will not have taken one of these pre-requisites but they have a comparable background to computer science students. The content is designed for and dual-listed for upper-division undergraduate or graduate students. Course content provides a broad overview of cybersecurity concepts, hands-on implementation of common software exploits (i.e. attacks against weaknesses in software applications), applications of cryptographic protocols, and discussion of various authentication methods, as well as concepts in network and web-based security. There are approximately 6 programming assignments and numerous published papers focused on cybersecurity research topics assigned for students to read. The course also includes a final paper on a current security topic of the student's choice.

Due to the small class enrollment and our lack of understanding where student interests and attitudes lie in relation to the field of cybersecurity, simply developing a survey-based assessment tool was not likely to produce statistically useful outcomes without several years of data. Additionally, with a survey-based assessment, significant aspects of student experiences and perceptions of cybersecurity of the course might have been overlooked entirely. For these reasons, I chose to utilize a qualitative approach rather than a quantitative approach for the first stage of this research. While these results do not lend themselves to elaborate statistical analysis, they do provide a greater breadth of information to explore how or why things have occurred within the population. The knowledge gained from this study will ensure that the quantitative instruments developed later are based on the perspectives, concerns and interests of the target population.

Interviews

Interviews are one of the most utilized qualitative methods in small-scale research. They tend to give high quality data with reduced ambiguity by allowing the interviewer to probe for clarification of a respondent's answers when necessary. The drawback is that both the interviews and analysis of responses take substantially more time than survey-based assessments.[53] Interview formats range from very structured sets of questions that are strictly followed to nearly unstructured formats with few guidelines, depending on the need and purpose of the research being performed.[54] This approach allows for new ideas to be uncovered and explored based on what the participants say, rather than potential preconceptions of researchers. The goal of this study is to gain an understanding of the participants' points of view concerning the course and its content.[55,56] Details and results of the qualitative study I performed are presented in Chapter 4.

2.4.2 Quantitative Study

While qualitative studies provide a way to discover new ideas and uncover hidden opinions that study participants have, they do not provide data which can be statistically analyzed to make decisions that concern a given population. In order to do this, a quantitative study, such as a survey, is useful. This is the method most frequently used in academic environments to assess student perceptions of course outcomes.

Despite the assessment efforts in introductory CS courses mentioned above, CS is lagging behind other fields such as physics, chemistry and mathematics when it comes to rigorous assessment of academic outcomes. One instrument in particular, the Colorado Learning Attitudes about Science Survey or CLASS, developed at the University of Colorado-Boulder, has produced excellent results and been used by numerous programs around the nation and world.[57,58] The CLASS measures student beliefs about physics and about learning physics in comparison to those of experts in the field. Student responses to survey questions are compared to those of experts, and scored to show the percentage of agreement between students and experts (which is considered to be a favorable response).

The CLASS has been successfully adapted for use in both Biology and Chemistry.[59,60] A version is currently being adapted for use in introductory CS courses as well, but it focuses on how student attitudes and beliefs about programming compare to those of experts in the field.[50] It is undergoing validation testing, and when that is completed, will provide a useful tool for the targeted courses. While such an instrument will be extremely useful, this particular adaptation will not work within the context of a cybersecurity course because of its focus on programming concepts rather than cybersecurity. The long-term goal of my work is to develop and validate an assessment tool which can be used to measure student interest and self-efficacy in relation to cybersecurity. The previous work by these authors provides examples and guidance for developing and validating such a tool. The details and result of my work towards developing this assessment tool are presented in Chapter 5.

Chapter 3

Teaching Computer Science in K-12 Classrooms

3.1 2013 STEM Summer Institute

As explained in section 2.2.5, the STEM Summer Institute provides an opportunity for middle school students to experience interesting, hands-on science enrichment programs. It also provides pre-service teachers at Kansas State University with an opportunity to gain valuable classroom teaching experience. When I was asked to work with the programming course, in addition to the chance to work with the K-12 students, I saw it as an opportunity to answer the questions:

- Are pre-service teachers capable of teaching basic programming concepts to students using a graphical programming environment?
- Can pre-service teachers effectively incorporate these concepts into the context of their own specialty teaching areas?
- Can we effectively introduce a graphical programming environment to pre-service teachers in a classroom setting in a way that builds their interest and self-efficacy in using these tools within a classroom setting?

3.1.1 Instructional Design

Previous work has shown that graphical programming environments are a good way to introduce programming to middle-school students.[61] Based on my own experiences, I selected Scratch as the development environment for this camp.[62] Due to the abbreviated length of each camp session and to help facilitate learning while reducing frustration, students worked in pairs.[63,64] To ensure all students had an opportunity to interact with the software, pairs were required to switch "drivers" (the person running the mouse and keyboard) as we progressed from topic to topic. Additionally, each student had a flash drive which contained a copy of Scratch and their project files. They were able to take these home each night if they wanted to work on or show off their projects.

Topics were presented in short lessons with the instructor and students first working together to solve a problem using a new concept and/or commands from the Scratch environment. This was followed by a challenge activity that required students to apply the new concept to solve some proposed problem. For each challenge activity, we had a primary challenge and then additional "alternative challenges" which were more advanced applications of the same concept for those students who finished early. Each day included a number of these lessons. This discretization of the material was very flexible and allowed for an easy transition as the pre-service teachers took over the teaching of individual lessons. It also allowed students to learn the material in steps, repeatedly employing previously used concepts for reinforcement while adding new topics to solve each new problem.

Based on previous experience, students first learning to use Scratch sometimes struggle with finding specific command blocks within the various Scratch menus or forget what options are available while working on their programs. To reduce the resulting frustration, I printed out and laminated a "cheat sheet" shown in Figure 3.1 that contained all of the Scratch menu options on a single piece of paper. While students were working in pairs, I noted the student not "driving" was often using this sheet to locate useful commands more easily. This helped reduce the need for teachers to repeatedly help locate for or suggest com-

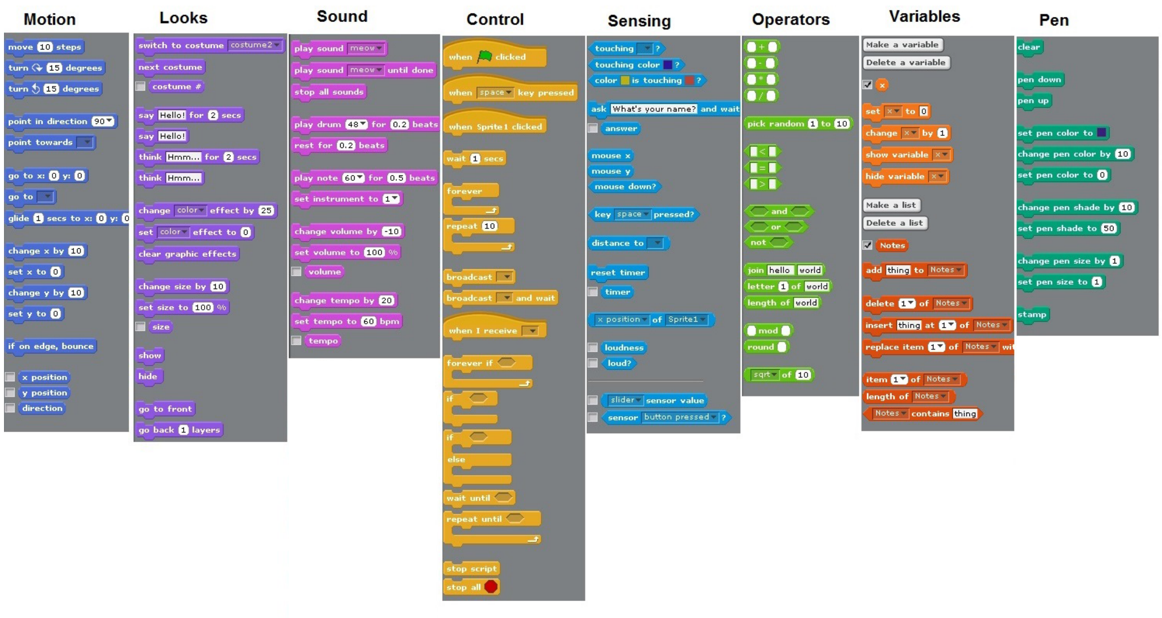

Figure 3.1: *"Cheat sheet" printed to help students locate commands more easily.*

mands to for the students. Students also used this sheet when they were designing program solutions on hand-held white boards prior to developing the program on the computer.

3.1.2 Day 1 - Programming Basics and Geometric Shapes

Day one covered the basic structured programming components: sequence, selection and iteration using Scratch. After going over basic commands and familiarizing the students with the Scratch environment, four lesson/challenge topics were covered, requiring students to develop algorithms that reproduced various geometric shapes by having a character on the screen (a sprite) "draw" the indicated pattern. Some examples of the patterns used are given in Figure 3.2. The in-service teacher, who taught middle-school math, appreciated the use of geometric figures (and the need to calculate angles and graphical dimensions) in these challenges.

Figure 3.2: *Progression of geometric figures used in the first set of lessons to first demonstrate programming concepts and then provide more challenging and creative projects for the students to build.*

3.1.3 Day 2 - Algorithm Design and Using Sound

On day two, we began to require students to think more carefully about how they might use the tools, giving less direct instructions for the challenges. For example, we demonstrated the functionality of (but did not show any pieces of the code for) a "worm" type application with a sprite which moved in a randomly shifting path across the screen, bouncing off the screen edges and leaving a colored trail as it moved, as shown in Figure 3.3. We then challenged the students to replicate this behavior. We wanted students to begin developing a habit of thinking about a solution rather than just guessing which blocks should be used. Therefore, beginning with this challenge, student pairs were required to design an algorithm for the challenge on their white board and have it approved before being allowed to program their solution. While the initial student designs were not always correct, the process of writing out a plan prior to "playing" with Scratch to make the program work is an important skill for young students to develop. This is where the "cheat sheet" was invaluable; students were able to look at the available commands for inspiration while sitting at work tables and thinking about how to solve the problem.

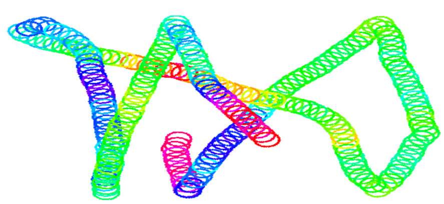

Figure 3.3: *Randomly moving "worm" sprites were the first challenge where we only modeled the program behavior for students.*

The second half of day two focused on using sound in Scratch. We briefly discussed how to use the sounds included with Scratch, and how to import songs and sound files from other sources including recording your own sounds. Writing music in Scratch has been used successfully in university-level general education courses,[43] and the pre-service music teachers were excited about using Scratch as a tool to teach music composition. With their help, we planned a set of lessons incorporating these ideas. We explained how to read notes from sheet music and convert them to the numeric values in Scratch, setting tempo, and how to determine the duration of each note. The students then translated one of several melodies from popular contemporary songs using sheet music we provided. The last demonstration of day two involved using broadcast messages (events) to coordinate multiple sprites playing a song in rounds using different instruments.

Day two ended with a planning session where pairs spent approximately 15 minutes brainstorming about their final project. These ideas were written down on paper and turned in. Many of the students struggled with this initially, but by asking a few leading questions, we were able to get them moving forward with a plan. The project plans were reviewed with the pre-service teachers and potential "trouble spots" which might be difficult for the students to solve were identified prior to day 3.

3.1.4 Day 3 - Broadcast Events, and Variables

Day three started with students writing an application which had sprites talking to one another (using speech bubbles). Students were able to synchronize the order of the conversation using broadcast messages (events) which we had discussed at the end of day 2.

The final topic we introduced was variables. Many groups were planning to develop some sort of game for their final project, so this topic was applicable to several pairs of students each week. In most cases, students were able to apply the concepts of incrementing, decrementing and initializing values to their own applications for variables such as score, levels and lives on their own after a brief introduction. The remainder of the work time for this day was spent working on final projects and taking a tour of the campus data center.

3.1.5 Day 4 - Final Project Wrap-up and Presentations

In previous outreach activities, I have had students ask about other languages compared to Scratch (with the insinuation that we were not teaching them "real" programming), so on day four I gave a brief discussion and demonstration of other programming languages. I compared constructs we had learned in Scratch with Java syntax which performed the same operations. I explained that while the Java code was more complex to write and provided greater flexibility, the students were, in fact, learning programming concepts that applied to other programming languages. The remainder of the day was spent finishing projects. We ended the day with students demonstrating their projects to the other camp attendees.

3.2 Pre-Service Teacher Progress

When I first met with the pre-service teachers 10 days prior to the start of camp, they were intimidated by the idea of teaching programming. They had little or no programming experience prior to the Summer Institute. For some of them, this was the first time they would be evaluated on their teaching methods in the classroom in front of students. The time between this initial meeting and the first day of the institute allowed the pre-service teachers to explore Scratch and get a feeling for the challenges we would be covering during the camp. As was mentioned in Section 2.2.5, the institute consisted of four 4-day camps, with the pre-service teachers becoming progressively more involved in the teaching from week to week. They were split into shifts, with 2 student teachers helping us during the first 2 hours each morning and the other 3 helping us during the second 2 hours. This gave them time to enroll in a course in the College of Education during the same 4 week period.

3.2.1 Week 1 - Observing, Helping and Ice Breakers

During the first week, the pre-service teachers were still learning Scratch and observing how I presented the material and interacted with the students. The pre-service music teachers had a much better understanding of music than I did, so they helped teach the musical encoding lessons. During other lessons they primarily helped answer questions when students were working on programming challenges. They also led the group (students and teachers) in ice-breaker activities to help us get to know one another and in "brain breaks" which were fun activities to help refocus the students between lessons.

3.2.2 Week 2 - Lead Enrichment Activity

During week two, in addition to the activities they performed in week one, each pre-service teacher presented an enrichment activity. These activities were about 15 minutes in length and supplemented the material we were covering in one of the existing lessons. For example, one of the activities involved students writing down a sequence of instructions that their partner then had to act out. This allowed us to discuss topics such as what commands were available to the "programmer," how a command might be interpreted differently by different "actors," and how some, more complex, commands are made up of several simpler commands. This led to open-ended questions such as "How many commands does it take to make a robot pick up a toothbrush?"

3.2.3 Week 3 - Group Teaching

For week three, the pre-service teachers continued to handle their previous responsibilities and each of the groups (a group of 2 and a group of 3) took over the responsibility for one of the lessons I had been teaching during the previous weeks. This included developing a formal lesson plan, presentation of the material and managing the challenge activities. The pre-service teachers covered the same general concepts I had covered during previous weeks, but they were free to change the details. Lesson plans were reviewed by the in-service teacher and evaluated by an instructor from the College of Education for pedagogical content and by me for technical content. The group then took over the teaching of that lesson during the week. This was where the pre-service teachers were able to utilize their education training and specialized background in art or music to improve the presentation and pedagogical approaches I had been using.

3.2.4 Week 4 - Solo Teaching

Finally, during week four, each pre-service teacher developed and taught a separate lesson independently. They were not allowed to repeat a lesson that they had taught during week three. The format and evaluation process were exactly the same as in week three except that the pre-service teachers worked individually on the lesson plans and presentations for this week.

3.3 Post-camp Discussions

After the camp concluded, I met with 4 of the pre-service teachers (3 music and 1 art) to get their impressions and experiences with Scratch and the camp. Their feedback was very positive:

- Three had shown Scratch to other pre-service teachers as well as relatives and/or friends
- All four reported that they were capable of teaching Scratch to their future students
- Two of the music teachers and the art teacher already had ideas of how they can incorporate Scratch into the classroom at various grade levels: *"I think I could apply it to any level, from elementary to high school. Teaching basic notes and rhythms would be easy on Scratch for my younger students and I could have high school students write their own compositions on Scratch."*

- One of the music teachers reported that Scratch likely would not fit with the musical concepts they anticipate teaching.

3.4 Lesson Plans

As an example of how the pre-service teachers developed a lesson plan, during week 3 one group took over the square drawing lesson presented on day 1. Their new lesson plan contained the following components:

- Target grade level
- Applicable standards and benchmarks covered
- Lesson goals
- Anticipated learner outcomes
- Vocabulary
- Assessment strategy and criteria
- Classroom set up and resources
- Experience activities
- How students will interact
- Transfer of learning strategy
- A scripted narrative of the lesson

They also developed a work sheet that allowed the students to move step by step through the thought process of building the program plan and had the students start by walking in a square. Students had to then describe these moves on their white boards using only 4 of the Scratch command blocks. Lesson plans developed in this manner are a valuable resource for other classroom teachers.

Self-Efficacy Measures		
Statement	**p**	**effect size**
I have the ability to earn an A in a computer programming class.	< 0.01	0.4812
I can learn how to write computer programs.	< 0.01	0.5182
I can learn to read code written using a computer programming language.	< 0.01	0.5823
Future Interest Measures		
Statement	**p**	**effect size**
I plan to continue writing programs after the Summer Institute is over.	< 0.01	0.5543
I would like to learn more about what computer scientists do.	< 0.01	0.2964
I would take a computer programming class in school if one is available.	0.180	0.1963
I want to learn how to program mobile devices such as phones and tablets.	0.970	-0.0197
Programming Enjoyment Measures		
Statement	**p**	**effect size**
I enjoy writing computer programs.	< 0.01	0.5973
I would enjoy a job that involves writing computer programs.	< 0.01	0.3054
I would enjoy writing computer programs that control robotic parts.	0.65	0.0954

Table 3.1: *Statistical summary of survey items calculated using Wilcoxon signed-rank test (n=52). Effect sizes: small* $\geq .10$*, medium* $\geq .30$*, large* $\geq .50$.

3.5 Student Survey Responses

Pre-camp and post-camp surveys were given to each participating K-12 student on the first and fourth days each week. The 10 item survey (see Table 3.1) utilized a 5 point Likert scale, with responses ranging from 1 = strongly disagree to 5 = strongly agree. The statements were designed to evaluate students' self-efficacy as it related to programming, future interest in programming after completion of the camp, and enjoyment of programming. Given the short notice prior to the start of camp, there was not an opportunity to pilot these statements prior to the STEM Summer Institute. Statements were reviewed by faculty members involved with the STEM Institute, other CS graduate students and the in-service teachers leading the programming course.

Given that the data being collected was ordinal and that I could not assume the data would be normally distributed, I selected the Wilcoxon signed-rank test for statistical significance. This test uses pairs of data taken from the same population. In this case, it requires pairing pre- and post-surveys from the same student. During week 1, I failed to collect the names of students on the pre-survey, so I was unable to use survey responses for that week when performing this paired data analysis. Additionally, some students missed either the first or last day of camp, and thus did not complete either the pre- or post-camp survey. These students' responses could not be used in the analysis either. This left 52 correlated, completed surveys for analysis out of the 69 students who took the programming course.

3.5.1 Self-Efficacy

Given that most middle-school-aged students have little or no experience with programming, I expected an increase in self-efficacy in relation to programming after the completion of the camp. I felt that this was an important metric because self-efficacy can be a determining factor in students' selection of career path.[11] The three statements chosen to measure student self-efficacy in programming are shown in Table 3.1. The results measured for all three items were found to be statistically significant using the Wilcoxon signed-rank test at the 1% level, meaning that the p-value for these results are less than 0.01. Values less than 0.05 are generally considered to indicate that there are statistically significant changes between the pre- and post-responses. In other words, when the p-value is less than 0.05, there is a measurable difference between the two samples and this difference is due to something other than random sampling effects. The table also shows the measured effect size for each

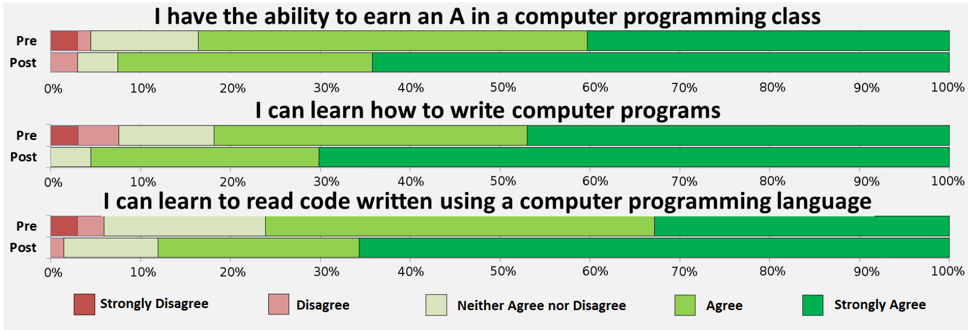

Figure 3.4: *Distribution of responses for statements related to self-efficacy in programming.*

statement. Effect sizes where $0.1 \leq e < 0.3$ are generally considered to be small, those where $0.3 \leq e < 0.5$ are considered to be moderate and those where $e \leq 0.5$ are considered to be large.

The responses were decidedly positive for all three items both before and after the camp, although it is of note that the percentage of students who strongly agreed with each statement increased by at least 50% for all three, and by 100% for 'I can learn to read code written using a computer programming language.' These distributions are shown in Figure. 3.4 Based on these results, the students appear to have come into the camp with moderate confidence in their ability to program, which would be expected in a self-selected summer camp. The students left with even stronger confidence in their abilities. Effect size was near or above 0.5 for all three, indicating a very positive effect from this camp.

3.5.2 Future Interest in Programming

The survey contained four measures of students' interests in further programming, given in Table 3.1. The results for these were mixed compared to those for self-efficacy. The change for 'I plan to continue writing programs after the Summer Institute is over' was significant (using the Wilcoxon signed-rank test), and again had a very positive effect size (0.554). This statement also had the greatest percentage increase in students who strongly agreed (120%), as shown in Figure 3.5. Student short-term excitement about programming increased significantly in just four days.

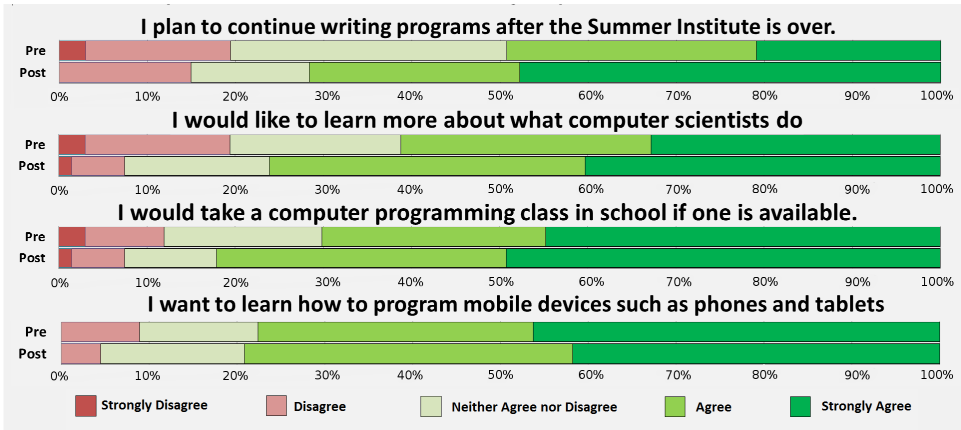

Figure 3.5: *Distribution of responses for statements related to future interest in programming.*

While the statement 'I would like to learn more about what computer scientists do' did not elicit as large of an effect size as other statements discussed so far, it was still significant. Considering that we did not spend much time discussing computer science explicitly, it does show that the experience caused increases in student interest in what we do. Interest in mobile device programming and in taking a computer programming class in school both had very small changes that were found to be statistically insignificant. This is likely due to the lack of discussion about these topics during the camp.

It is interesting to note that initially only 49% of students stated they had plans to program after the camp ended, but a large percentage *were* interested in learning how to program mobile devices (78%). By the end of the week, the percentage of positive responses for the two items are very similar (72% to 79%, respectively). Student interest in programming increased greatly while interest in mobile programming remained virtually unchanged. In other words, initially, the idea of programming mobile devices was more interesting to students compared to the general concept of programming. However, by the end of camp, students found the overall idea of programming to be just as interesting as programming mobile devices specifically.

3.5.3 Programming Enjoyment

There were three items in the survey that measured students' enjoyment of programming, as shown in Table 3.1. The first, 'I enjoy writing computer programs' showed the largest measured effect size of any statement on the survey. Just under 60% of students agreed initially to this statement, but by the end of the camp, 85% agreed (Figure 3.6). While students were not as enthusiastic about a job that involves writing computer programs, this item showed moderate growth which seems reasonable for middle school students who selected this course during a summer camp. The third statement, "I would enjoy writing computer programs that control robotic parts," performed similar to the previous question about mobile devices. Students showed interest at the start of the camp (72% agreement) and the change by the end of camp was found to be insignificant. As was discussed with mobile devices, we did not spend time discussing robots, so seeing little change in this measurement is understandable.

The outcomes from the summer institute verified the effectiveness of non-STEM teachers using Scratch (or some other, similar development environment) as a teaching tool within non-STEM related subjects. Given the teachers' very short preparation time and that this was their first experience being graded on their teaching in front of a classroom, it is possible the results achieved in a regular classroom over the course of several years could be much stronger. While this opportunity was a very important step, and it demonstrates that pre-service teachers were able to effectively teach programming content, it does not resolve the second issue of how to increase the number of teachers that are familiar with, and interested in using, such tools within their classrooms.

3.6 Education Technology Course

Based on the successful feedback received from the STEM Summer Camp, a faculty member at Kansas State was able to get an invitation to present a generalized and shortened version

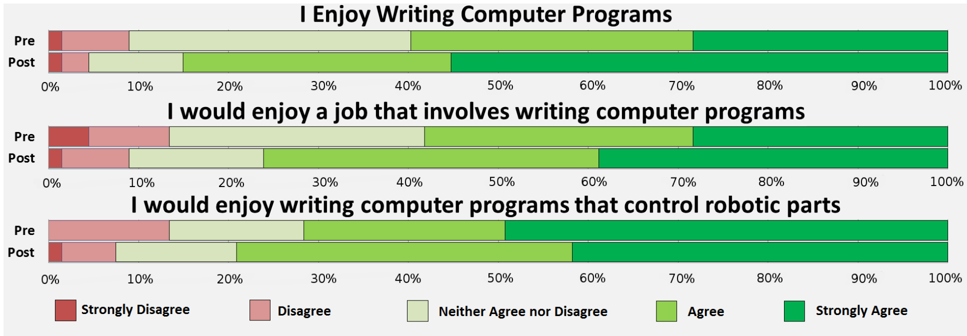

Figure 3.6: *Distribution of responses for statements related to enjoyment of programming.*

of the material to all sections of the Education Technology course at Kansas State University during the Fall 2013 semester . The goal of the presentation was to introduce the pre-service teachers to the idea of computer programming and to show them that it can be easily and successfully incorporated into lessons at the K-12 level. There were 5 sections of the course and we had a total of 108 responses to a survey which students completed at the start and end of the class period.

As part of the class session, we had the pre-service teachers develop programs that demonstrated ways in which programming could be used in courses such as math, science, music and art. We then discussed the skills that students develop when they are solving computer programming problems. We gave pre- and post-surveys to the pre-service teachers to measure their interest and self-efficacy in relation to using programming as a teaching tool in their future classrooms.

3.6.1 Methodology

Given that this was a short, one-day presentation, we elected to present the material in a workshop style. Pre-service teachers were each given laptops and we walked through some basic programs to get them familiarized with how Scratch works. We then utilized a process fairly similar to that used in the Summer Institute. We would present a simple program such as drawing a triangle then challenge the pre-service teachers to modify the code to draw other regular shapes such as a square, pentagon or hexagon. Finally, we challenged them to find a way to generalize this idea to draw any regular shape. On completion of the challenge, we discussed how it might be used in various educational contexts.

Another lesson we incorporated into the presentation involved coding music for a song and having the pre-service teachers develop multiple sprites playing the song in rounds. After that, we worked with animation, having sprites change appearance and inserting speech bubbles into the program. The final demonstration was a simulation which would calculate and plot the trajectories of a projectile as the initial angle of launching was changed from 0 to 90 degrees and then reported the angle at which the projectile travelled the furthest distance from the starting point. This allowed the pre-service teachers to observe both how Scratch could be used to help students think through and develop an experiment and also how it can be used to automate and visualize the results of a simulation. Each of these projects was selected to demonstrate how programming can be used within different academic fields (math, music, creative writing, physics, etc.).

3.6.2 Results

The survey consisted of 10 statements and was given before and after the in-class workshop. Responses to each statement were scaled on a 5 point Likert scale ranging from "strongly disagree" to "strongly agree" The survey statements, along with the p-value and measured effect size of each, are shown in table 3.2. Effect sizes greater than 0.5 are considered to be large, while p-values less than 0.01 indicate that there are statistically significant changes between pre- and post- responses. The distribution of pre-service teacher responses can be seen in figure 3.7. Prior to using Scratch, only the first statement "I feel confident using computer technology" received agreement from more than 35% of the respondents. This shows that while they were confident in their ability to use computer technology, they had little confidence in their ability to use or teach with programming.

Figure 3.7 shows the distribution of responses to the survey statements. In the post-class

	Statement	p	effect size
Q1	I feel confident using computer technology.	< 0.01	0.3645
Q2	I feel confident writing simple programs for the computer.	< 0.01	0.7678
Q3	I know how to teach programming concepts effectively.	< 0.01	0.8450
Q4	I can promote a positive attitude towards programming in my students.	< 0.01	0.6621
Q5	I can guide students in using programming as a tool while we explore other topics.	< 0.01	0.7465
Q6	I feel confident using programming as an instructional tool within my classroom.	< 0.01	0.6325
Q7	I can adapt lesson plans incorporating programming as an instructional tool to meet my students' learning.	< 0.01	0.6568
Q8	I can create original lesson plans incorporating programming as an instructional tool.	< 0.01	0.7108
Q9	I can identify how programming concepts relate to Common Core Standards.	< 0.01	0.7411
Q10	I can identify how programming concepts relate to Next Generation Science Standards.	< 0.01	0.7328

Table 3.2: *Statistical summary of survey items calculated using Wilcoxon signed-rank test (n=107). Effect sizes: small* $\geq .10$*, medium* $\geq .30$*, large* $\geq .50$*.*

survey , all of the statements had more than 40% agreement (students who either agreed or strongly agreed with the statement) and only 2 statements were below 60% agreement. After one 2-hour session spent using Scratch, over 60% of pre-service teachers felt confident in their ability to: write simple programs, promote a positive attitude towards programming in their students, use programming as a tool for exploring other topics, incorporate programming as an instructional tool, and relate programming to educational standards.

As mentioned above, there were 2 statements which did not have at least 60% agreement in the post-course survey. The first of these: "I know how to teach programming concepts effectively" only had 5.6% agreement to begin with and this grew to 44.8%. This statement did however have the highest measured effect size. The second statement: "I feel confident using programming as an instructional tool within my classroom" grew from 17.8% to 46.7%. So, while the pre-service teachers were confident in their ability to situate programming within their curriculum, they were still somewhat apprehensive about using it as a true instructional tool.

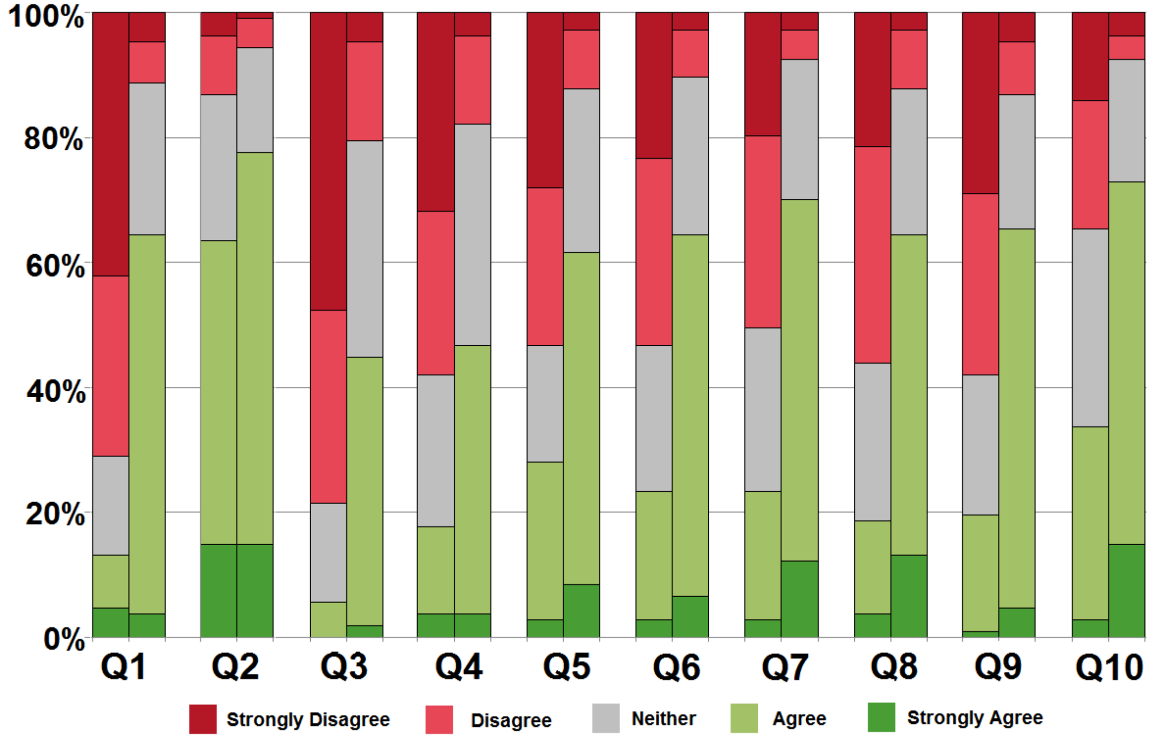

Figure 3.7: *Distribution of responses from pre-service teachers in education technology course.*

3.6.3 Discussion

The magnitudes of these results were even better than I had expected. The large effect sizes are likely due in part to the pre-service teachers' excitement at learning a new, novel, fun, concept using a very engaging tool. This is exactly what Scratch has been designed to do, and generating this type of excitement was our intent when we designed this workshop. I would expect pre-service teacher interest and confidence to level out or even regress somewhat for a while if the lessons were extended over multiple sessions. However, the results *do* demonstrate that pre-service teachers can develop an interest in learning about basic computer programming concepts and that they report being able to visualize ways in which tools such as Scratch might be useful in their future classrooms. Currently, most pre-service teachers are never exposed to this material, and so there is never an opportunity for them to even consider such a possibility. This work, combined with the summer institute described in section 3.1 shows that given the opportunity and background, some pre-service teachers likely would employ programming in their teaching and that they can do so effectively in such a way that student interest, self-efficacy and enjoyment will grow.

Chapter 4

Qualitative Study in Cybersecurity Education

As was mentioned in Section 2.3.1, cybersecurity education is a relatively new field of academic study, even within the still new field of computer science. Additionally, this sub-field is growing so rapidly that it has been difficult for educators to ensure they are developing consistent, quality teaching practices. However, change is coming. Groups such as ACM, IEEE, and ACM-SIGITE are developing and publishing curriculum standards for cybersecurity, showing what content industry and academics feel best fit within the computer science and information technology curriculum.[30,31] The long-term goal of my work in this area is to help the academic community take a step towards developing tools that can be used to assess student interest and self-efficacy in relation to studying and working in the field of cybersecurity. The first step involved interviewing students enrolled in a cybersecurity course at Kansas State University. The objective of these interviews was to better understand their experiences while enrolled in the course over the duration of the semester. The results of this work were then used to develop a survey-based instrument for assessing student outcomes in cybersecurity courses, the development and results of this instrument are discussed in Chapter 5.

4.1 Methodology

Student volunteers for the semester-long study were solicited from an introductory cybersecurity course and included both upper division undergraduate as well as graduate students. To encourage student participation and reduce the likelihood of participants not completing the study, volunteers who completed the entire study were promised and given a small financial incentive ($50) upon completion of the 3rd interview. From a course enrollment of 30 students, 18 initially agreed to participate. Participants were interviewed 3 times, as shown in Figure 4.1. Interviews took place during the 2nd, 10th and 15th weeks of the semester. Of the initial volunteers, 15 participated in the first round of interviews, 14 participated in the second round, and 12 participated in the third round. Of the initial 15 participants, 6 were undergraduate computer science students, 3 were undergraduate computer engineering students and 6 were graduate computer science students. Two of the graduate students were female; all other participants were male. The results presented within this chapter represent responses from all students who participated in a given round of interviews (I did not remove the responses of students who participated in round 1 but then did not participate in rounds 2 and/or 3).

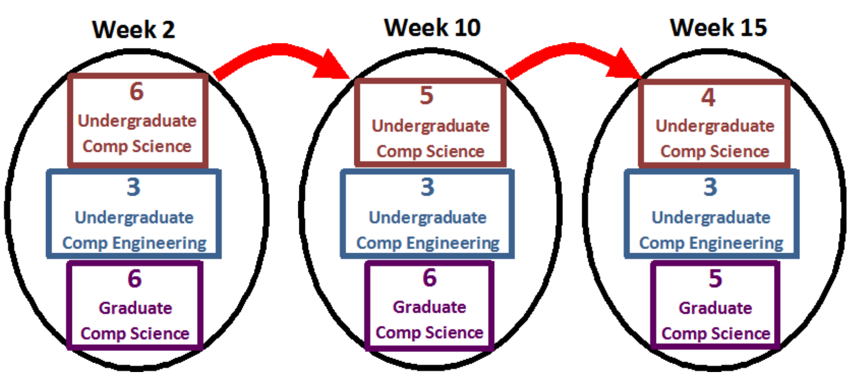

Figure 4.1: *Number of students and distribution among departments for each round of interviews.*

Interviews were semi-structured and lasted from 15 to 30 minutes. With the semi-structured format, a base set of questions was established for each interview but additional questions were asked when topics of interest were uncovered or points of clarification were needed.[55,54] The base set of questions were derived in discussions with a faculty member in cybersecurity courses and then reviewed by both graduate and undergraduate students within the CS department for clarity. A pair of pilot int Interviews was performed prior to the first round of interviews to identify questions that were unclear or that were not generating the expected results. These questions were modified to fix the identified problems. In cases where a new question would be useful for all students, these questions were carried over to the remaining interviews within that round. Interviews were video taped to allow for post-interview analysis. Some of the sound content during the first round of interviews was not intelligible, so some of that data was lost. In a semi-structured interview format, participants are encouraged to elaborate and discuss their answers, so they might mention several topics in response to a single question.[65] For these reasons, the number of items indicated for some questions does not always add up to the total number of participants. Topics covered included:

- reasons for taking the course
- experiences in the course
- specific topics of interest
- student's perception of his or her own ability in cybersecurity
- student's perception of his or her progress in the course
- interest in cybersecurity as a career path and/or a topic of further study or research
- how well students felt the course was preparing them for future studies or positions in cybersecurity.

4.1.1 Interview Response Bias

Bias is an important concern in qualitative research. The interviewer can affect the respondent's answer in many ways. One component of this is when the respondent attempts to please the interviewer by answering questions the way they believe the interviewer wants them to. I took several steps to reduce the likelihood of this type of bias. When first explaining the process to the students, I made it clear that I was not involved in the class in any way and that the instructor would never have access to the responses other than a summarized, anonymous report of my findings. I spent time giving my own background and explaining why I was performing the research. All of this was done to encourage an open and honest dialog with each student. I also approached topics beginning with broad questions requiring discussion type answers then moving towards more specific questions that might elicit a yes/no response. For example, near the end of the first interview, I had prepared the following sequence of questions:

- describe what you feel cybersecurity research might involve
  - Do you think you could be successful doing that kind of work? (why/why not?)
  - Is that something you would enjoy doing?
    - if yes, what parts?
    - is there a specific reason you would/would not enjoy it?

This sequence of questions eases the student into the topic, first describing what they believing cybersecurity research is prevents them from giving a single word answer and lets them contemplate the subject first without having to show their own feelings on the subject initially. Then I follow up by asking if they could be successful doing that and if they would enjoy it. For both of these questions, I ask additional questions to keep the student talking about how they view themselves within cybersecurity and limit their ability to continue with one word answers. In many cases, when the students responded to the yes/no questions, they would include the explanation without my having to ask for it. While I cannot guarantee that all bias is eliminated from the interviews, I did attempt to limit the opportunities that students had to resort to simply agreeing with a statement.

4.2 First Round Interview Results

During the first interview, students were asked why they chose to take the course. Several gave more than one reason, so I counted the number of times each reason was given:

- 5 students mentioned that it filled an elective for their degree
- 8 said it sounded interesting
- 3 called it important
- 6 mentioned that they wanted to learn how to make code more secure
- 2 indicated that it was part of their research area.

Additionally, students were asked to give a definition of cybersecurity during the first and second interviews. The goal was to see whether and how student views of cybersecurity changed over the course of the semester. I noted patterns in specific terms that students mentioned during each interview. In both interviews, students linked actions such as protecting, securing, preventing, defending and finding with components such as networks, hardware, systems, data, and programs. Table 4.1 shows the distribution of terms among the students. The most interesting change is the tripling in the number of students that mentioned data, files, or information as part of their definition. In fact, every 'category' was mentioned more frequently during the second set of interviews, showing that students were becoming more aware of the breadth of the threats faced today.

Students were also asked during the initial interview if they felt they could be successful conducting cybersecurity research. One student responded "absolutely" (this student withdrew from the course prior to the second interview). Other student responses included phrases such as "I think so," "probably," and "not sure." One student replied "I don't know enough yet to answer that question." Results are shown in Table 4.2.

As a follow-up question, I asked students if they thought that cybersecurity research was something they would enjoy doing. Ten students responded with "yes" or "sure" and 4 students gave less certain but positive responses such as "probably." When asked if they

Table 4.1: *Topics mentioned by students when asked to define the term cybersecurity.*

First Round		Second Round	
Protecting / securing / defending / preventing	14	Protecting / defending / preventing	17
Finding	2	Applying or using concepts	2
Network	6	Network / Internet	10
Hardware / computer / server	3	Hardware / computer / server	6
System	3	System	4
Data / files / information	4	Data / files / information	12
Software / programs	2	Bugs / weak spots	3
		Authentication / verification / authorization	4

felt that cybersecurity (not specifically research) would be involved in their career after graduation, all students felt that it would be involved in their career in some way, although 3 specified that it depends on the job. So, at the beginning of the semester, students felt cybersecurity would likely be involved in their future careers and they were interested in cybersecurity research although they were not very confident in their ability to do the work.

Results from asking students if they would seek out positions that involved aspects of cybersecurity specifically are also shown in Table 4.2. The overall positive responses from the students for these questions was expected given that these students are enrolled in an elective cybersecurity course.

During the first interview, I also investigated student experience with cybersecurity prior to the class. Students were asked to "Describe encounters you have had with cybersecurity prior to this course." While the students did mention some key concepts, nearly half (7) indicated they had little experience with or were "oblivious to" cybersecurity prior to the course. Some of the previously encountered concepts that were mentioned included stack "smashing" and buffer overflow attacks, which were discussed in previous courses, finding and removing viruses on personal computers, phishing attacks, and social account (such as email, Facebook, etc.) compromise.

Table 4.2: *First interview questions related to students' perception of future cybersecurity research and work.*

Do you feel you could be successful conducting cybersecurity research?	
Absolutely	1
I think so / probably	7
Not sure	4
I don't know enough to answer that question	1
Question not asked	1
Is cybersecurity research something that you would enjoy doing?	
Yes / sure	10
Probably	4
Do you see cybersecurity involved in your career after you graduate?	
Definitely / yes	11
Depends	3
Do you plan to seek out jobs that involve cybersecurity aspects specifically?	
Definitely / yes	5
No	2
There are other, more important, aspects to job search	7

4.3 Second Round Interview Results

The results from several of the questions asked during the second round of interviews are given in Table 4.3. First, students were asked how they were doing in the course. Three were unsure how they were doing (all were undergraduates), 2 felt they were doing poorly (one undergraduate). Another felt he was doing moderately well while 6 thought they were doing well. Several students were very focused on how they were doing compared to their peers. One student responded with the following: "in terms of all the undergrads, we're doing about the same." Despite these mixed feelings about their performance, when asked, every student was glad they were taking the course. Reasons given included that they felt they had learned something useful or interesting and that it gave them a broader understanding of the subject.

All 11 students who were asked if they thought the course was interesting agreed that it was, although the level of interest varied: "it is *very* interesting" versus "some of it is."

Table 4.3: *Second round interview questions. (Some questions were not asked during every interview)*

How are you doing in the class?	
Well	6
Moderately well	1
Poorly	2
Unsure	3
Are you glad you decided to take this course?	
Yes	12
No	0
Is this course interesting?	
Yes	11
No	0
Do you feel that you could be successful in a job that involves cybersecurity?	
Definitely	1
Yes	6
With additional work / help	4
Not sure	2

Another question I asked students was "Do you feel that you could be successful in a job that involves cybersecurity?" Seven students felt they could be, while another 4 students thought that with work/help they would be. Only 2 students were not sure if they could be successful. No students said they could not be successful in a cybersecurity job.

During the second round of interviews I also asked students what topics they would like to learn more about. The list of topics included:

- everything just deeper
- more about buffer overflow attacks
- correct implementation of PKI (public key infrastructure)
- mobile security (mentioned by multiple students)
- newer attacks; the buffer- and heap-based attacks are out of date
- biometric authentication
- web security.

I asked this same question during the third round of interviews with similar results.

During both the second and third interviews I asked if students planned to take additional cybersecurity courses. The department has several options for students to choose from. There is a cryptography course, a lab-based course focusing on common exploits and penetration testing tools, and an advanced (graduate level) cybersecurity course focused on network security, security protocols, and the composition of security tools from building blocks. Since several (5) of the students were planning to graduate or leave by the end of the academic year, there was some hesitation when answering this question. This was a fall course, so there was the possibility of taking a course during the spring term prior to graduation, though many of them were not sure if an additional course would fit into their graduation plans. Student responses are given in Table 4.4. Note the trend toward more definitive positive answers – the number of students answering a definitive "yes" did not go down, but students who were previously uncertain moved from more negative to more positive answers.

4.4 Third Round Interview Results

During the third interview, students were asked to rank themselves on a scale from 1 to 10, with 1 indicating no knowledge of cybersecurity and 10 indicating being a cybersecurity expert. Two students rated themselves between 1 and 3, 3 students rated themselves between 4 and 5 while 5 students rated themselves between 5.5 and 7 and 1 student selected 8. The

Table 4.4: *Do you plan to take additional cybersecurity courses? (Note that 2 students who participated in the second interview did not participate in 3rd interview)*

	Second Interview	Third Interview
Yes	6	5
Not sure	2	4
Depending on schedule	3	2
I don't think so	3	1

one remaining student first selected 4.7 then changed to 6.5 then 7 then finally selected 9.9. When asked about this decision process, the student stated that compared to the rest of the world, he knows substantially more than a vast majority of people at this point. This shows that the question needed a frame of reference. All of the students indicated that in order to rank higher, they felt they needed more experience and exposure to cybersecurity concepts either in the classroom, a work environment, or through research. One student noted that he might even rate himself lower as he learns more about cybersecurity. These results are shown in Table 4.5. During this final interview, I asked each student about his/her confidence in cybersecurity moving forward. The results were varied, with 6 students expressing confidence in their abilities, 3 being somewhat confident, 2 being unsure and 1 having little confidence in his ability.

Table 4.5: *Student self rating of cybersecurity competence.*

Rating	Number of students
1 - 3	2
4 - 5	3
5.5 - 7	5
8	1
9 - 10	0
(One additional student changed from 4.7 → 6.5 → 7 → 9.9)	

One of the main focuses of the course is having students perform attacks against various platforms. Several (7) students indicated that this was an eye opening or "aha" experience for them, making comments such as *"I was thinking to myself 'You can do that?' "* Other students also mentioned having a similar "aha" reaction when learning about how the Kerberos authentication system works.[66] There were numerous comments about this course being a good introduction and that the students were interested in learning more about various topics including: more complicated exploits, network security, cryptography, other attacks such as denial of service, and Wi-Fi security.

4.5 Case Studies

One of the major advantages of performing a qualitative study is the ability to investigate information that presents itself during the interview process. This allows the investigator to discover unknown or interesting pieces of information that would be difficult to discover using a quantitative method such as a survey. I selected two students to look at more closely. One is a senior in Computer Engineering and the other is a graduate student in Computer Science.

4.5.1 Student 1

I will call the first student I am investigating Aaron. Aaron is a senior in computer engineering who is working towards a networking specialization. This course was recommended to him by his adviser from a list of electives for the networking specialization because of his interest in programming. Throughout all 3 interviews, Aaron was fairly confident in his ability to succeed in the course as well as in a career that involved cybersecurity activities. He expects to work in programming of some sort upon graduation and had an internship that involved software development last summer.

Through responses to several questions, Aaron indicated that he expects cybersecurity to play an important role in this future work whether he chooses a career that specifically focuses on it or not. He also expressed interest in individual research projects involving cybersecurity. However, when asked if he would seek out a job specifically oriented towards cybersecurity, Aaron indicated that there were several other factors that were more important in making that decision such as what products or services the company produced.

As was mentioned previously, I asked students to define cybersecurity during the first interview. Here is the response Aaron had during that interview: *"Protecting something on a computer or network...securing a system in the cyber-world. Preventing people from doing things you don't want them to do."* I asked the same question during the second round of interviews 8 weeks later. He gave a similar answer: *"Trying to protect information or things in cyber-world or computer world or internet...like any other security system to keep information from getting into the wrong hands only this is on a computer or network of computers."*

While the responses are similar, there are a few interesting changes in the two responses. In the second response, Aaron starts off with the word trying. Considering the context of the course at this point, students have just spent several weeks learning about various exploits and how powerful they are. Most will also be realizing how difficult it is to defend against some of these attacks. Additionally, he now compares cybersecurity to "any other security system" during the second interview. So, he is now beginning to see that the threats in the cyber-world are as real and multi-faceted (notice he struggled somewhat in identifying precisely what is being protected) as those encountered in other aspects of areas.

A third difference is that in the first response, he uses the word "something" while in the second response he specifically mentions "information" as the item being protected. This is something I mentioned previously that many of the students became aware that information or data is really the object that needs to be protected. All three of these differences suggest that Aaron is becoming more aware of the details and difficulties of cybersecurity and is developing a larger-picture, more mature, view of the field as a result of this course.

Another question I asked during the third interview was if the students had experienced any 'aha' moments that stood out. Aaron mentioned the first time they were shown how to perform an exploit: *"I was like...you can do that!?"* This type of excitement was common among a large number of students, and this particular activity was the most commonly remembered moment. It is a point in the class where students are able to observe just how vulnerable some systems can be due to both poor programming practices and a determined adversary.

Also during the third interview, I asked where he felt he ranked on a scale of 1 to 10 as far as understanding of cybersecurity. He responded: *"I want to put it lower because* *there is so much depth...maybe at a 6 or 7, maybe an 8 depending on the topic. A lot of it is still black box and I don't feel super confident about it."* This was one of the higher self-rankings among the students I interviewed. Although he obviously does not feel he has a fundamental understanding of the material yet, Aaron felt confident in his ability to work within the cybersecurity field and succeed in the classroom in all 3 of the interviews.

Aaron was a good representative of several undergraduates that I interviewed. These students were confident in their ability to tackle whatever content was presented in the course. However, when asked to rate themselves, they were more conservative, and without a specific frame of reference, tended to rate themselvees near the center of the scale. Perhaps the confidence is due to trusting the academic setting and not real confidence in their overall knowledge. While the students were very interested in learning about cybersecurity and did not rule out taking jobs in cybersecurity upon graduation, few expressed an interest in seeking out cybersecurity jobs specifically. This is to be expected given that this is an single elective course and students have a wide range of sub-fields within computing to choose from.

4.5.2 Student 2

I will call the second student I am presenting Brad. Brad is a master's student in computer science. He mentioned that he had written a few web applications prior to enrolling for his graduate degree and that they were not very secure. In the second interview he went on to mention that he is looking to become a web developer and/or a data analyst and that learning how to make web applications more secure was his primary motivation for taking the course. He was simultaneously enrolled in the cyber-defense basics course and mentioned several times that taking the two courses at the same time really helped him grasp the concepts well. Unfortunately, the sound for the "define cybersecurity" question during the first interview was not working, so that response was not available. During the second interview, he gave the following definition: *"Knowing what kind of attacks a system* *might go through and trying to protect against them."*

Brad had a very narrowly defined purpose for taking the course initially. However, by the third interview, his perspective had changed quite a bit as can be seen by his response to my question *"how do you see cybersecurity fitting into your future?"*: *"...I was a web developer. I wanted to integrate security into my web apps so that's what I thought about doing after this course and now my scope has widened and I will consider other ways to make them secure. In the future I would like to explore or try out an attack using our knowledge to try on something else... in the future I would like to dig deeper into these fields I want to explore how attacks are done not just apply it."*

This response shows that the Brad now wants to investigate how attacks work on his own and would be interested in pursuing research. Interestingly, in a previous interview, I had asked about his interest in a cybersecurity job and his response indicated that he had not considered that possibility before.

As with Aaron, Brad mentioned the first time he saw an overflow attack work as a moment that stood out in the class. He rated himself as a 6.5 or 7, stating: "I think my networking basics are not up to the marks, that's why I rated myself very low. I think a strong background in networking would help a lot." When asked about future plans during the third interview, Brad was not as directed as Aaron. His response to questions about seeking a job in cybersecurity was *"yes, why not? ... If I'm offered a job, I would definitely take it."*

Brad definitely experienced a change in his view of what cybersecurity involves over the course of the semester. The combination of the two courses appears to have had a significant impact on how comfortable he felt with and interested he was in the material. He became more aware of the breadth of cybersecurity and was notably more interested in areas other than his initial interest in securing web applications. He did still have uncertainties in his ability, but overall his interest in the field grew. As with Aaron, he rates his understanding of cybersecurity slightly above the middle on a scale of 1 to 10. Additional questions showed that he had confidence in his ability to be successful in cybersecurity given additional instruction and/or study time.

4.5.3 Comparing Cases

Aaron and Brad came into the course with very different objectives. Brad was taking the course to learn how to secure his web applications while Aaron was directed to the course by his adviser since it was an elective for computer engineers working towards the networking specialty and was programming oriented. From the start, Aaron was open to the potential of a cybersecurity job or possibly research, while Brad started out more focused on other sub fields. However, by the end of the course Brad "would definitely take it" if a job was offered to him in cybersecurity. Over the course of the semester, interest in cybersecurity for both students grew noticeably.

Both students were hesitantly confident in their abilities throughout the interview process. However, when asked to rank their ability on a scale, they chose numbers only slightly above the middle. This was very common among students and I interpret this as confidence in their ability to learn and succeed in this sub-field but not necessarily in their complete understanding of the material at this point. Students were quite varied in how they described their own abilities within this field although most responded similarly (fairly confident in ability to succeed but rated themselves lower when quantifying their ability).

One place where these two students differ is in their future plans. Aaron does not have plans for graduate school, so he was less inclined towards research although he did mention that he felt he could do research in cybersecurity. Brad on the other hand, as a current graduate student, was interested in research in general at the start of the course and by the end he was very interested in intrusion detection systems and various other aspects of cybersecurity research.

4.6 Qualitative Study Summary

I performed 3 rounds of interviews with students enrolled in an upper-division cybersecurity course. My primary goal was to study student interest and self-efficacy in relation to cybersecurity. Specifically, I was interested in learning about student interests in cybersecurity as they pertain to future plans such as careers, research, and classwork. I also wanted to investigate student self-efficacy as it changes during the course of the semester: are students gaining confidence in their ability to deal with cybersecurity issues?

During the interviews, and the subsequent analysis of the student responses, it became clear that the perspective of each student can vary greatly when responding to a given question. This was evident in the responses to the question asking students to rate themselves on a scale of 1 to 10. One student listed several different choices and when asked to explain why he changed his mind, he explained that his opinion changed as he realized how he compared first to cybersecurity experts and then to larger groups of less-knowledgeable people. This is an issue that will need to be considered when performing additional validation on the survey instrument.

Based on responses, there is student interest in cybersecurity at all levels (courses, research, and jobs) although most of the students see it as a component of their overall education and career plans, not the main focus. All students were glad they had taken the class and found the material at least somewhat interesting if not very exciting. This was expected given that the course is an upper-division elective course with many alternatives available to students.

Students' self-efficacy did not rate as highly as their interest. In fact, for some students, it decreased over the course of the semester. Given that students admitted to having very little knowledge of cybersecurity at the beginning of the semester and taking into account the enormous scope of the field, this was not unexpected. This is evident in case of Brad, who explicitly mentions his *"scope widening"*. The fact that many were interested in continuing to learn more about cybersecurity and that they were glad they took the course shows that while the material was somewhat intimidating, they were not discouraged by the experience. The responses obtained from this investigation were then incorporated into a survey instrument for use in assessing student outcomes from cybersecurity courses.

Chapter 5

Quantitative Study in Cybersecurity Education

In order to systematically improve the quality of cybersecurity education, an instrument which can assess student interest and self-efficacy in relation to cybersecurity from training experiences (which may include classes, workshops, seminars, or tutorials) needs to be developed and validated. Cybersecurity is a new and growing field, and there has been little work towards developing such an assessment tool. This chapter presents the initial development of, and results from, a survey designed to measure student interest and self-efficacy outcomes from cybersecurity courses. It builds upon the findings from an initial qualitative investigation presented in Chapter 4.

Surveys allow us to provide quantitative analysis of student experiences in the classroom. They enable us to efficiently measure the impact of various pedagogical decisions and thus help us identify teaching methods that are producing the desired effects in the classroom. My goal in pursuing this research path is to create an instrument which instructors can utilize in cybersecurity courses to assess changes in student interest and self-efficacy as it relates to cybersecurity. I do not focus directly on the level of knowledge students possess about individual topics as courses can vary widely in their coverage and goals with respect to each individual topic.

As was discussed in Section 2.3, a major obstacle in the development of such a tool is the wide variety of content and instructional methods used in teaching such courses. At Kansas State University, the target course for this work is CIS 551/751 which is an introductory cybersecurity course covering a broad range of topics. Students enrolling in the course are typically junior/senior-level undergraduate or graduate students. The course work includes approximately 6 programming assignments, numerous outside articles to read, a final paper and an in-class presentation. Class time tends to focus on explaining how various software exploits work during the first half of the semester and then moves into discussing security protocols and components during the second half. The last 2-3 weeks of the class are reserved for students presenting papers that are part of the assigned reading for the course.

5.1 Methodology

5.1.1 Survey Development

Given the wide variety of pedagogical methods and goals among cybersecurity courses discussed in Section 2.3, I chose not to focus on quantifying student knowledge of specific cybersecurity topics. Instead, the survey I have developed focuses on students' interest and self-efficacy as it pertains to a variety of cybersecurity topics ranging from "Install and run malware checking software on a home computer" to "Manage security for a Fortune 500 company." This choice was made because the field of cybersecurity is very broad and rapidly changing. Additionally, the depth of coverage of specific topics is likely to vary greatly between courses. Thus, knowledge-based metrics would be difficult, if not impossible, to validate and would likely be outdated quickly.

Measuring student interest, indicates how well a given course is motivating students to pursue further knowledge or work in this sub-field. Building long-term student interest is vital within a new, fast-changing, field such as cybersecurity. Likewise, measuring student self-efficacy is important because it has been linked with outcomes such as persistence on task and academic as well as long-term career success.[11,13]

To measure these 2 attributes, I first developed 22 topic statements like those mentioned above. A majority of these topics statements were derived from the interviews discussed in chapter 4. These were topics which students showed interest in learning more about, that they mentioned when discussing job plans or were topics that stood out in their descriptions of experiences in the class. Additional topics were chosen to provide data concerning specific areas of interest such as *"Take additional courses focused on cybersecurity."* The goal in selecting the topics was to cover a wide range of material as well as a variety of ways in which students might further engage with cybersecurity material in both academic and work environments.

I then devised 3 general statements which allow students to indicate their level of interest and confidence in each of the 22 topics. An estimated time to accomplish the task measurement was included to allow students to differentiate between topics they felt they already were capable of and those they *could* become capable of with enough time. This resulted in 66 data points per survey. A copy of the survey developed as part of this work is included in Appendix A. The statements, and options students can choose from include:

- My interest in this topic
  - Very interested, Somewhat interested, Not very interested, Not interested, I don't know what this is
- My confidence in undertaking and succeeding in this activity
  - Very confident, Somewhat confident, Not very confident, Not confident, I don't know what this is
- Estimated time for me to prepare for and accomplish this
  - At most a few days, A few weeks, Between a month and a year, A year or more, I would not be able to do it on my own

For each statement above, there are 4 options which provide students with the ability to rank their interest, confidence and anticipated time to prepare for and accomplish a give topic. I chose 4 items to avoid students selecting a middle value when responding while still providing a balanced number of options on either side (2 generally positive and 2 generally negative). By removing this neutral response, this forces students to select whether they lean towards either interested/not interested and confident/not confident. Responses to these statements will provide us with an idea of how each student believes they fit within this scale. Issues with analyzing ordinal data such as this are discussed in the results section below.

The survey was reviewed for face validity by more than 20 graduate and undergraduate students within the computer science department prior to being used. Primarily this was done to check the clarity of statements and to verify that students within the target audience would understand how to respond to the survey.

5.1.2 Participant Selection

Ideally, I would prefer to survey a large number of students enrolled in the target cybersecurity course. This would provide a statistically significant evaluation of the course and allow us to determine how the instrument performs in such an environment. However, the target cybersecurity course at Kansas State University (CIS 551/751) is only offered during the fall term. This was the course which was used for the interviews discussed in Chapter 4 during the Fall, 2013 term. Two other cybersecurity courses were offered during the Spring 2014 semester, although the enrollment in these courses was very limited (CIS 490 had 14 students and CIS 755 had 6 students). The 490 course, Cyber Defense Basics, is a 1 credit hour lab-based course designed for advanced undergraduate and graduate students. This course is focused on learning about the common applications and tactics currently employed in cyber-attacks and discussing ways to defend against such attacks. The 755 course, Advanced Computer and Information Security, is primarily designed for graduate students (though a few undergraduate students take the course). In this course, students learn about current security tools and best practices for effectively protecting systems and information. I performed pre- and post-course surveys of the students enrolled in both of the courses during the Spring 2014 semester.

Due to the small number of students in these 2 courses, and the fact that they were primarily upper level students, I also surveyed students enrolled in CIS 200, the introductory programming course for the department, which had an enrollment of 138 students. This course was selected because it has a fairly large enrollment and also because it would provide data from introductory students who are not likely to have much experience with cybersecurity topics. This provides a contrasting view of how student interest and self-efficacy in relation to cybersecurity change not only as students take cybersecurity courses, but also as they advance through the computer science department. The survey was administered in class during the 3rd week and the 13th week of the semester. Below is a summary of the student enrollment and participation totals from each surveyed course.

	CIS 200	CIS 490	CIS 755
Enrollment	138	14	6
Pre-Course Survey	93	11	5
Post-Course Survey	74	9	6
Both Surveys	61	8	5

Table 5.1: *Student enrollment and participation numbers from each course surveyed during Spring 2014 term.*

5.2 Results

One issue to remain aware of when analyzing this data is that the students from CIS 200 represent the general CS population. It is unclear if any of these students fit the profile for those who will go on to take the target cybersecurity courses. However, any student who enrolls in 1 or more of the cybersecurity courses must first take CIS 200 (or somehow obtain that pre-requisite knowledge). Looking at the enrollment numbers in CIS 200 and those in

CIS 551, it can be estimated that 10%-20% of the students in CIS 200 will go on to take CIS 551. CIS 551 and the other cybersecurity courses are electives, so the students enrolled in these course have already displayed some level of interest in this field. A secondary, long-term, goal of developing this survey instrument is to enable educators to identify students within the general CS population who are likely to pursue coursework and/or careers in cybersecurity.

5.2.1 Overall Averages

As mentioned above, there are 66 data points per student for each pre- and post-survey (132 total for students who filled out both surveys). The first step towards data analysis was cleaning the data. The first 4 options for each statement can be ordered and thus are assigned a value from 1 to 4, with 1 indicating very interested/confident or the least amount of time to achieve the task. The 5th option indicates the student felt unable to answer the question and thus responses with this result are handled separately.

At this point I should discuss the issues of analyzing ordinal data. The scales used for interest and confidence allow students to select from options which have an inherent order. However, the gaps between these values are not necessarily equivalent, and likely vary from person to person. This limits the precision of the results obtained from data analysis. However, trends indicating changes in confidence and interest can still be identified. Additionally, since the same students are taking both the pre- and post-survey, they likely interpret the scales the same way each time they took the survey, resulting in fairly consistent handling of each student's responses. So, averaging all students' pre- and post-course responses together and calculating the change in the average for each statement, produces a value that is useful in studying class trends over the course of a semester. For example, if there is a reasonably large sample size and on average there is a 1 step shift towards "not interested," on the 4 step scale, this indicates that there was some sort of change that occurred in the population during the time between the two surveys.

To analyze the data, I first averaged all responses with a value of 1 to 4 on pre- and post-surveys that were collected. Given the large proportion of students that are enrolled in the CIS 200 course, the results of this analysis will only be able to show general trends in the data, and can be compared to the results from each course discussed in Section 5.2.2 to see the impact that they have on the overall results. This data is summarized in Table B.1. These results show a central tendency for each statement, giving an idea of which topics students were more interested or confident in. Comparing the pre- and post-survey averages shows the change in this central tendency for each statement over the course of the semester. This table also contains symbols indicating p-values < 0.1 (discussed in Section 5.2.3). Values of $p < 0.05$ indicate a statistically significant change was found.

Overall, within the pre-survey responses, students indicated greater levels of interest than self-confidence in all but two of the topics based on the average response values. The two exceptions are "read articles/web posts about cybersecurity on your own" and "install and run malware checking software on a home computer." These are 2 of the top 3 rated values among the confidence measures. These are topics which most students are likely already at least somewhat familiar with and might have already performed on their own. This would explain their greater confidence in performing these tasks. The estimated time selections tended to be conservative, with the average in most cases being between a month and a year or more. The exceptions to this are the same 2 topics mentioned above along with "Remove detected threats from a home computer." For these statements, average time estimates fall between a few days and a few weeks. Again, this can be the result of the students being more familiar with these activities, and therefore more comfortable with a shorter time estimate to complete them.

Post-survey results are similar, although the difference between interest level and confidence level is reduced, with more cases (6) where the student indicated confidence level exceeds the interest level. In fact, for 15 of the 22 statements, changes indicated student interest decreased, while confidence levels on 19 of the 22 statements increased. This would seem to indicate that as students became more confident in their ability in cybersecurity, they became less interested in the subject. One possible reason for this would be a large bias from the CIS 200 students, who did not have direct exposure to cybersecurity topics. An increase in overall student confidence after completing the introductory programming course would be reasonable. At the same time, it is possible that, for younger students, interest in general decreases towards the end of the semester.

5.2.2 Analyzing Changes Between Pre- and Post-Course Surveys

Figures B.1 and B.2 show the change measured for each interest and confidence statement in the survey, respectively. To make it easier to locate interesting data values, each graph is ordered so that the questions with the largest change are at the top or bottom of the graph. Graphs are included for both the overall average change values as well as the average changes for each course. Looking at the interest graphs, it can be readily seen that the overall average change values are highly influenced by those from the CIS 200 course, as mentioned above. While the order is not exactly the same, it is nearly so, and the size of the changes are extremely similar to one another.

Looking at the cybersecurity courses, it is somewhat surprising to see the large number of questions which exhibit a loss of interest (remember that higher values indicate less interest). Given the small sample sizes for these courses, 1 or 2 students can have a significant effect on the average, so a change of 0.5 is not necessarily a strong indicator of a trend in these courses. However, these results do show us which topics should be monitored in future assessments.

The results for the confidence statements where exactly opposite of those from the interest statements. The overall averages showed slight changes trending towards more confidence. As with the interest data, the CIS 200 graph and the overall graph look nearly identical both in shape and in the order of the questions.

In looking at the confidence graphs for the cybersecurity courses, unlike the interest graphs, for some questions trending towards less confidence and for others trending towards more confidence. Question order also varies. For example, question 6: "Discover ways to protect personal data on the Internet" trends toward less confidence for the 490 course while it trends toward greater interest for the 755 course. Again, with such small sample sizes and an ordinal data scale, the magnitude of the changes in these courses does not provide a precise measure of the effects of the course, just a suggestion of how student interst and self-efficacy are trending. Future surveys with greater sample sizes are expected to provide stronger results.

5.2.3 Statistical Analysis

In order to verify that the measured values from the pre- and post-course surveys were statistically different, I performed a statistical analysis of the data. To do this, only students who gave a valid response between 1 and 4 to a given statement on both surveys could be counted. If a student only participated in one of the surveys, I was unable to use that survey in the analysis. Likewise, if a student either did not answer a given question, marked multiple answers, or indicated they did not know what the topic was for a question, that student's response for that question was not used in the analysis. Because of this, the number of data points per measure varies.

I used the Wilcoxin signed-rank test to validate the data. This test is good for non-parametric data with matched samples which cannot be assumed to have normally distributed data. The test gives a p-value which can be used to determine if the indicated effect size is likely to be the result of randomness in the data. A small value of p will indicate this is unlikely and that the effect size is instead a significant outcome. The results of this analysis are shown in Table B.2. Results with a p-value less than the 0.05 threshold value indicate that the calculated effect sizes are significant. There are 8 data points with p-values below this threshold. For those results, the effect sizes are considered to be statistically significant. Effect size values ≥ 0.5 are considered to be large, ≥ 0.3 are considered medium and ≥ 0.1 are considered to be small effects.[67] I also calculated the Hedges' g values for each measure. This is a more conservative calculation of effect size. These results were similar to those found using the Wilcoxon test.

Statement	p-value	effect size	Hedges' g
Confidence in pursuing an advanced degree focused on cybersecurity	< 0.01	0.338	0.30
Interest in taking additional courses focused on cybersecurity	< 0.05	-0.242	-0.22
Interest in discovering ways to protect personal data on the Internet	< 0.05	-0.264	-0.22
Confidence in learning how to use SSL certificates	< 0.05	0.288	0.26
Confidence in learning how to intercept and read network traffic	< 0.05	0.280	0.21
The time it would take to learn how to intercept and read network traffic	< 0.01	0.383	0.33
The time it would take to learn how to verify a digital signature	< 0.01	0.408	0.34
Interest in reading articles/web posts about cybersecurity on their own	< 0.01	0.307	0.23

Table 5.2: *Statements with p-values* < 0.05 *showing statistically significant changes between pre- and post-surveys and the measured effect sizes. Positive effect sizes indicate students became more interested or confident in the topic, or felt it would require less time to complete. Effect sizes: small* $\geq .10$*, medium* $\geq .30$*, large* $\geq .50$

For these values, a negative effect size indicates that students have less interest/confidence in the statement, or felt it would require more time. While it would be better to have all measurements showing significance, there were at least two factors which made this unlikely. First, a large majority of the students surveyed were not enrolled in a cybersecurity course, and there was little expectation for large changes in the measures among those students. Second, the number of students enrolled in the two cybersecurity courses who completed both parts of the survey (13) is too small to provide statistical power.

5.2.4 Per Course Analysis

The next question to investigate is how the responses are distributed among the 3 courses. For example, are introductory students more or less interested and more or less confident in learning how to use SSL certificates than students enrolled in cybersecurity courses? To determine this, I broke the responses down by course then averaged the responses for each option. As before, I only used those responses that were one of the first 4 options.

Table B.3 contains the data from the interest responses for all 3 courses separated by course. There is a very noticeable difference between the central tendencies of CIS 200 students and those of the more advanced students. This is expected since the students enrolled in the cybersecurity courses are self-selected and should display a greater interest than a population of students enrolled in a general computer science course. For many of the statements, students appear to become less interested over the course of the semester. The small sample sizes in CIS 490 and 755 make this data less reliable for measuring the effects within a given course, but the results do show that the survey measures differences between the two sets of students (those that have chosen to take a cybersecurity course and the general population of CS students).

I performed the same analysis with the confidence responses. This data is shown in Table B.4. There is a discernible difference between the populations within this data as well, but it is not as clear as the difference seen in the interest response data. Considering that all of the students surveyed are enrolled in computer science courses, they would be expected to have confidence in their ability to solve problems within a computer science context. Some differences can also be seen between the 490 course and the 755 course in these responses. For example, the statement "Learn how to intercept and read network traffic" averages 2.68 ± 0.13 for CIS 200, and 2.63 ± 0.26 for CIS 490 while the average for CIS 755 was 1.8 ± 0.58. This shows that the 755 students were more confident than students in the other 2 courses in their ability to learn how to intercept and read network traffic. Again, the sample sizes were small for the upper division courses, but the ability to see a difference in the measures is promising.

5.2.5 CIS 200 Statistical Analysis

I repeated the Wilcoxon Signed Rank test using only the responses from the CIS 200 course to see if there were more or less significant changes within this data set. Given that the student population is more uniform, there was the potential that this might occur. However, since these students had little or no exposure to cybersecurity subjects, I expected little change in values over the course of the semester. The results are given in Table B.5. Inspecting the values reveals that 4 results had a p-value below the 0.05 threshold. These topics and the associated effect sizes are shown in table 5.3

Statement	p-value	effect size	Hedges's g
Confidence in pursuing an advanced degree focused on cybersecurity	< 0.05	0.276	0.258
The time it would take to learn how to intercept and read network traffic	< 0.05	0.33	0.357
Interest in writing an algorithm that uses asymmetric encryption to authenticate a user	< 0.05	-0.326	-0.38
Interest in reading articles/web posts about cybersecurity on their own	< 0.05	0.319	0.28

Table 5.3: *Statements from CIS 200 course with p-values < 0.05 showing statistically significant changes between pre- and post-course surveys and the measured effect sizes. Positive effect sizes indicate students became more interested or confident in the topic, or felt it would require less time to complete. Effect sizes: small $\geq .10$, medium $\geq .30$, large $\geq .50$*

Three of these statements were also found within the results when all students responses were analyzed together. The one new statement is "Interest in writing an algorithm that uses asymmetric encryption to authenticate a user."

5.3 Summary

Based on information gathered from the qualitative study discussed in Chapter 4, I developed a survey instrument that focuses on student interest and self-efficacy in relation to several cybersecurity topics. These include various topics focused on jobs, classes and/or research focused on cybersecurity. This survey was given to students enrolled in 3 courses at Kansas State University at the beginning and end of the Spring 2014 semester. The courses included a graduate level cybersecurity course, a graduate/undergraduate dual-listed laboratory based course focused on introducing students to current cybersecurity utilities, and an introductory programming course for all computer science students. The introductory programming course was included due to the small enrollment in the two cybersecurity courses, which did not provide a large enough sample size to produce statistically significant results. The inclusion of the introductory course provided a much larger sample size and an opportunity to compare results from introductory level students with those from upper level students.

I first looked at the average responses for all participants. These values are shown in table B.1. Pre-course results showed that for most topics, student interest was higher than confidence. These differences were reduced, and in some cases reversed in the post-course data. Further analysis of the individual course results showed that there is a distinct difference in student responses between courses, as would be expected. Upper division students in the 2 cybersecurity courses had greater interest and confidence than the introductory students in all but 4 of the interest and self-efficacy items, as would be expected. Further analysis of the combined data from the 3 courses showed 8 topics which produced statistically significant effects as shown in Table 5.2. A similar analysis of data from the introductory programming course indicated 4 topics which produced statistically significant effects as shown in Table 5.3.

These results show that the survey is capable of distinguishing survey results from a cybersecurity course when compared to results from an introductory programming course.

It was also able to detect significant changes between the pre- and post-course responses for several of the topics. Unfortunately, the enrollment in the cybersecurity courses was too small to allow for statistical analysis of the responses from those courses individually. This would be the next step in the development of this instrument.

Chapter 6

Conclusions

Computer science is a relatively new academic field. As such, it has not reached the levels of academic maturity that other fields such as Math, Physics and Chemistry have. Computer science is struggling for recognition as a core subject within the K-12 curriculum. The number of graduates per year nationally has fluctuated dramatically over the last 40 years, ranging from a few thousand in the early 1970s, peaking near 60,000 in the 2003 before dropping to below 40,000 by 2007. Content within various courses changes frequently, with new languages or pedagogical approaches being introduced to improve student outcomes every few years. This constant change has lead to confusion for students, and uncertainty about the best approaches to teaching computer science topics. There is even uncertainty as to where CS fits within higher education. At some universities, CS is housed within the College of Engineering while other schools incorporate the department into Business, Mathematics or Arts and Sciences. Yet other schools have created a separate branch to house computing.

From the proper perspective, this diversity exemplifies the wide-reaching impact of the field. Nearly every aspect of life today involves computing. The demand for computing graduates is growing at an incredible rate, and computer science departments need to improve recruitment and retention of students within the field. My research efforts approach this problem on two fronts. First, I believe we need to address the shortage of teachers at the K-12 level that have experience in computing. they do not need to be experts in programming, but they do need to be interested and to be able to pass that interest on to their students at every grade level. Second, I believe we need to do a better job of retaining the students that do choose to enroll in our programs, and encourage them to focus on areas of high demand such as cybersecurity. In order to do this, we must develop better pedagogical strategies.

6.1 Improving Enrollment and Recruiting Efforts

Providing ways to incorporate computer science topics into K-12 classrooms will enable teachers to introduce students to computing concepts *before* those students make decisions about what to study in college. Few school districts currently offer any sort of computing concepts within their existing curriculum. This leaves students relatively uninformed when the are making this fairly important decision, and has been shown to be a major factor in students *not* selecting CS as a major.

By introducing Scratch to pre-service teachers enrolled in the Education Technology course, I was able to show how quickly they became interested and excited about the idea of using this type of tool in the classroom. Both interest and self-efficacy results were extremely promising and I am looking forward to taking this work to the next step. I plan to work with faculty in the College of Education at my new university to find ways to introduce tools such as Scratch into the education curriculum as a teaching tool, not just as a programming tool.

The results of my work with pre-service teachers within the Summer STEM Institute shows that pre-service teachers are capable of introducing these basic programming concepts within art and music contexts. The students with whom these pre-service teachers worked showed strong increases in interest and self-efficacy in programming over the course of a 4 day summer camp. The K-12 students were genuinely excited about programming by the end of the camp. I do not expect all of the students to become CS majors, and I realize that what they experienced is only a small piece of what constitutes computer science. However, I do believe that even after this short experience, they have a taste of what computer science might be about. I also believe that if enough students encounter that type of experience, then more of them will choose to enter the field when they attend college. Given more exposure over the span of several years, these students would have a much clearer view of the scope and depth of computer science and thus would be able to make a more informed decision when selecting a career path.

6.2 Improving Content and Retention Efforts

Once students do choose to enroll in a computer science program, we have to ensure that we are providing a high quality educational experience that prepares students for careers today and into the future. While we can look at arguments such as high placement rates and claim that current teaching methods are adequate, looking at other academic fields such as Math, Physics and Chemistry shows that we are not looking closely enough at what and how we are teaching. These other fields approach the topic of education as a scientific pursuit in and of itself. They have developed and validated instruments for measuring student outcomes and perform detailed studies on the effects of various pedagogical decisions in a very systematic manner, identifying which approaches produce the best results (i.e. improved student performance).

Developing an assessment instrument for cybersecurity is a requisite first step towards maturity for a field that is still growing. Such a tool, once developed and validated, will allow cybersecurity educators to more effectively adjust teaching practices within their classrooms. Without such a feedback mechanism, instructors are left guessing about what changes need to be made and which of those changes are most effective. With the work presented here, I have taken the first step in this direction. By first performing a longitudinal qualitative study of students in a cybersecurity course, I was able to identify important topics and concepts that shaped students' views of cybersecurity and track how these changed over the course of a semester. I then used this information to develop an initial survey instrument for assessing student interest and self-efficacy as it relates to cybersecurity.

This survey was then used to measure interest and self-efficacy among students in 3 different courses. Unfortunately the 2 cybersecurity courses offered during the semester when the study occurred did not have large enough enrollment numbers to provide statistically significant results, so I also surveyed the introductory programming course. This allowed me to see the difference in survey results of students just beginning their computer science studies and those that are near to graduation. It also gave me the ability to see differences between students who have chosen to take a cybersecurity course versus the general population of computer science students.

6.3 Future Work

There is still much work to be completed on both fronts.

6.3.1 K-12

While I have shown that we can incorporate CS concepts into material that pre-service teachers can then use in the classroom, this concept is yet to be implemented on a large scale. I plan to extend this effort at my new institution, working with the College of Education faculty to find ways to implement lessons similar to those presented within the Kansas State Educational Technology course on a permanent basis. The objective of this work will be twofold. First, I expect to introduce *every* student teacher to programming concepts. This may only be a few lessons, but by doing so, I hope to provide them with a view of the potential that programming can offer to their students.

The second objective is to have these student teachers develop lesson plans which incorporate programming concepts into their individual areas of expertise. This will cause these teachers to consider new ways to introduce the material within various academic fields and also begin to produce a library of lesson plans, created by teachers, that incorporate computing concepts into each of those academic fields. This material will be organized and made available as a library to other teachers looking for ways to utilize programming in the classroom.

6.3.2 Cybersecurity Assessment

I also plan to continue development of the cybersecurity assessment tool. With the initial development complete, the next step will involve adjusting the survey to improve student response rates and clarify questions. I will have access to at least 1 large cybersecurity course in the Spring 2015 semester where further testing will be performed. This should provide a large enough sample size for statistical analysis and factor analysis of survey results from cybersecurity students. Further adjustments will become possible based on these results. Given the size of most cybersecurity courses, this will be a long process and once I have adjusted the survey it will involve efforts of faculty at more than 1 university in order to complete.

Bibliography

[1] Computer Science Teachers Association. Bugs in the system: Computer science teachers certification in the U.S. http://csta.acm.org/ComputerScienceTeacherCertification/sub/CSTA_BugsInTheSystem.pdf, 2013.

[2] Stuart Zweben and Betsy Bizot. 2011 Taulbee survey: Continued increase in undergraduate cs degree production; slight rise in doctoral production. *Computing Research Association*, 24(3):7–24, 2012.

[3] Lori Carter. Why students with an apparent aptitude for computer science don't choose to major in computer science. *ACM SIGCSE Bulletin*, 38(1):27–31, 2006.

[4] Lissa Pollacia and William Lomerson. Analysis of factors affecting declining cis enrollment. *Issues in Information Systems*, 7(1):220–225, 2006.

[5] NSF. Science and engineering degrees: 1966-2006, 2008. Retrieved from http://www.nsf.gov/statistics/nsf08321/.

[6] Matt Bishop. Teaching computer security. In *SEC*, pages 65–74, 1993.

[7] Eugene F Spafford. Teaching the big picture of infosec. In *2nd National Colloquium for Information System Security Education*, 1998.

[8] David Carlson. Teaching computer security. *ACM SIGCSE Bulletin*, 36(2):64–67, 2004.

[9] John Hill, Curtis A Carver Jr, Jeffrey W Humphries, and Udo W Pooch. Using an isolated network laboratory to teach advanced networks and security. In *ACM SIGCSE Bulletin*, volume 33, pages 36–40. ACM, 2001.

[10] Ronald S Cheung, Joseph P Cohen, Henry Z Lo, and Fabio Elia. Challenge based learning in cybersecurity education. In *Proceedings of the 2011 International Conference on Security & Management*, volume 1, 2011.

[11] Albert Bandura, Claudio Barbaranelli, Gian Vittorio Caprara, and Concetta Pastorelli. Self-efficacy beliefs as shapers of children's aspirations and career trajectories. *Child development*, 72(1):187–206, 2001.

[12] Deborah R Compeau and Christopher A Higgins. Computer self-efficacy: Development of a measure and initial test. *MIS quarterly*, 19(2):189–211, 1995.

[13] Robert W Lent, Steven D Brown, and Kevin C Larkin. Self-efficacy in the prediction of academic performance and perceived career options. *Journal of counseling psychology*, 33(3):265, 1986.

[14] Robert W Lent, Steven D Brown, and Kevin C Larkin. Relation of self-efficacy expectations to academic achievement and persistence. *Journal of counseling psychology*, 31 (3):356, 1984.

[15] Dale H Schunk. Self-efficacy and academic motivation. *Educational psychologist*, 26 (3-4):207–231, 1991.

[16] Sarita Yardi and Amy Bruckman. What is computing? Bridging the gap between teenagers' perceptions and graduate students' experiences. In *Proceedings of the third international workshop on Computing education research*, pages 39–50. ACM, 2007.

[17] Phoenix Moorman and Elizabeth Johnson. Still a stranger here: attitudes among secondary school students towards computer science. In *ACM SIGCSE Bulletin*, volume 35, pages 193–197. ACM, 2003.

[18] Cameron Wilson, Chris Stephenson, Mark Stehlik, and Leigh Ann Sudol. Running on empty: The failure to teach computer science in the digital age, 2010.

[19] College Board. Program summary report, 2013. Retrieved from http://research.collegeboard.org/programs/ap/data/participation/2013.

[20] Exploring computer science, 2014. retrieved April 13, 2014 from: http://www.exploringcs.org/resources/cs-statistics.

[21] Jiangjiang Liu, Cheng-Hsien Lin, Ethan Philip Hasson, and Zebulun David Barnett. Introducing computer science to K-12 through a summer computing workshop for teachers. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 389–394. ACM, 2011.

[22] Judy Clark, Michael Rogers, and Carol Spradling. Scratch the workshop and its implications on our world of computing. *Journal of Computing Sciences in Colleges*, 26(5): 235–243, 2011.

[23] Amy Bruckman, Maureen Biggers, Barbara Ericson, Tom McKlin, Jill Dimond, Betsy DiSalvo, Mike Hewner, Lijun Ni, and Sarita Yardi. Georgia computes!: Improving the computing education pipeline. In *ACM SIGCSE Bulletin*, volume 41, pages 86–90. ACM, 2009.

[24] Lenore Blum and Thomas J Cortina. CS4HS: An outreach program for high school CS teachers. In *ACM SIGCSE Bulletin*, pages 19–23. ACM, 2007.

[25] Scott Bell, Tim Frey, and Eugene Vasserman. Spreading the word: Introducing pre-service teachers to programming in the K-12 classroom. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 187–192. ACM, 2014.

[26] Dale C Rowe, Barry M Lunt, and Joseph J Ekstrom. The role of cyber-security in information technology education. In *Proceedings of the 2011 conference on Information technology education*, pages 113–122. ACM, 2011.

[27] Michael E Locasto, Anup K Ghosh, Sushil Jajodia, and Angelos Stavrou. The ephemeral legion: Producing an expert cyber-security work force from thin air. *Communications of the ACM*, 54(1):129–131, 2011.

[28] CR Dodge, Costis Toregas, and Lance Hoffman. Cybersecurity workforce development directions. In *Proceedings of the Sixth International Symposium on Human Aspects of Information Security and Assurance*, 2012.

[29] Stephen Cooper, Christine Nickell, Victor Piotrowski, Brenda Oldfield, Ali Abdallah, Matt Bishop, Bill Caelli, Melissa Dark, Elizabeth K Hawthorne, Lance Hoffman, et al. An exploration of the current state of information assurance education. *ACM SIGCSE Bulletin*, 41(4):109–125, 2010.

[30] IEEE ACM. Computer science curricula 2013, 2013. Retrieved from http://ai.stanford.edu/users/sahami/CS2013//final-draft/CS2013-final-report.pdf.

[31] ACM-SIGITE. IT2008 Model Curriculum, 2013. Retrieved from http://www.sigite.org/.

[32] Scott Bell, Eleanor Sayre, and Eugene Vasserman. A longitudinal study of students in an introductory cybersecurity course. In *Proceedings of the 121 ASEE Conference*. ASEE, 2014.

[33] Jodi L Tims and Susan R Williams. The TauRUs project: a complement to the taulbee report. *ACM Inroads*, 3(1):62–73, 2012.

[34] Digest of educational statistics, 2012. http://nces.ed.gov/pubs2014/2014015.pdf, Dec 2013.

[35] College Board. AP Program Participation and Performance Data 2013, 2013. http://research.collegeboard.org/programs/ap/data/participation/2013.

[36] Lillian Boots Cassel, Andrew McGettrick, Mark Guzdial, and Eric Roberts. The current crisis in computing: What are the real issues? *ACM SIGCSE Bulletin*, 39(1):329–330, 2007.

[37] Shuchi Grover, Roy Pea, and Stephen Cooper. Remedying misperceptions of computer science among middle school students. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 343–348. ACM, 2014.

[38] C Dianne Martin. Draw a computer scientist. *ACM SIGCSE Bulletin*, 36(4):11–12, 2004.

[39] Jeannette M. Wing. Computational thinking. *Commun. ACM*, 49(3), March 2006.

[40] Chi-Cheng Lin, Mingrui Zhang, Barbara Beck, and Gayle Olsen. Embedding computer science concepts in K-12 science curricula. In *ACM SIGCSE Bulletin*, volume 41, pages 539–543. ACM, 2009.

[41] Vladimir Brusic. The growth of bioinformatics. *Briefings in bioinformatics*, 8(2):69–70, 2007.

[42] Mark Guzdial. A media computation course for non-majors. In *ACM SIGCSE Bulletin*, volume 35, pages 104–108. ACM, 2003.

[43] Alex Ruthmann, Jesse M Heines, Gena R Greher, Paul Laidler, and Charles Saulters II. Teaching computational thinking through musical live coding in Scratch. In *Proceedings of the 41st ACM technical symposium on Computer science education*, pages 351–355. ACM, 2010.

[44] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.

[45] Robert Bryant, Richard Weiss, Genevieve Orr, and Kathie Yerion. Using the context of algorithmic art to change attitudes in introductory programming. *J. Comput. Sci. Coll.*, 27(1), October 2011.

[46] MIT. Scratch, MIT Lifelong Kindergarten. http://scratch.mit.edu, 2013.

[47] Burning Glass Technologies. Job market intelligence: Report on the growth of cyber-security jobs. http://www.burning-glass.com, 2014.

[48] Bureau of Labor Statistics. Occupational outlook handbook. http://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm#tab-6, 2012.

[49] Simon Cassidy and Peter Eachus. Developing the computer user self-efficacy (CUSE) scale: Investigating the relationship between computer self-efficacy, gender and experience with computers. *Journal of Educational Computing Research*, 26(2):133–153, 2002.

[50] Brian Dorn and Allison Elliott Tew. Becoming experts: measuring attitude development in introductory computer science. In *Proceeding of the 44th ACM technical symposium on Computer science education*, pages 183–188. ACM, 2013.

[51] Vennila Ramalingam and Susan Wiedenbeck. Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research*, 19(4):367–381, 1998.

[52] Eric Wiebe, Laurie Williams, Kai Yang, and Carol Miller. Computer science attitude survey. *computer science*, 14(25):0–86, 2003.

[53] Eric Drever. *Using Semi-Structured Interviews in Small-Scale Research. A Teacher's Guide.* ERIC, 1995.

[54] Jerry Willis. *Qualitative Research Methods in Education and Instructional Technology.* IAP, 2008.

[55] Robert C Bogdan and Sari Knopp Biklen. *Qualitative Research in Education. An Introduction to Theory and Methods.* ERIC, 1998.

[56] Jack R Fraenkel and Norman E Wallen. *How to design and evaluate research in education.* McGraw-Hill New York, 1993.

[57] Wendy K Adams, Katherine K Perkins, Michael Dubson, Noah D Finkelstein, and Carl E Wieman. The design and validation of the colorado learning attitudes about science survey. In *2004 Physics Education Research Conference*, volume 790, pages 45–48, 2004.

[58] Wendy K Adams, Katherine K Perkins, NS Podolefsky, M Dubson, ND Finkelstein, and Carl E Wieman. A new instrument for measuring student beliefs about physics and learning physics: The colorado learning attitudes about science survey. *Physical Review Special Topics-Physics Education Research*, 2:01–14, 2006.

[59] Wendy K Adams, Carl E Wieman, Katherine K Perkins, and Jack Barbera. Modifying and validating the colorado learning attitudes about science survey for use in chemistry. *Journal of Chemical Education*, 85(10):1435, 2008.

[60] Katharine Semsar, Jennifer K Knight, Gülnur Birol, and Michelle K Smith. The colorado learning attitudes about science survey (class) for use in biology. *CBE-life sciences education*, 10(3):268–278, 2011.

[61] Colleen M Lewis. How programming environment shapes perception, learning and goals: logo vs. Scratch. In *Proceedings of the 41st ACM technical symposium on Computer science education*, pages 346–350. ACM, 2010.

[62] Scott Bell, Linda Heeler, and Phil Heeler. Computer programming with middle school students. *Journal of Computing Sciences in Colleges*, 24(5):161–165, 2009.

[63] Grant Braught, L Martin Eby, and Tim Wahls. The effects of pair-programming on individual programming skill. In *ACM SIGCSE Bulletin*, volume 40, pages 200–204. ACM, 2008.

[64] Laurie Williams, Eric Wiebe, Kai Yang, Miriam Ferzli, and Carol Miller. In support of pair programming in the introductory computer science course. *Computer Science Education*, 12(3):197–212, 2002.

[65] Joseph C Hermanowicz. The great interview: 25 strategies for studying people in bed. *Qualitative sociology*, 25(4):479–499, 2002.

[66] Steven P Miller, B Clifford Neuman, Jeffrey I Schiller, and Jermoe H Saltzer. Kerberos authentication and authorization system. In *In Project Athena Technical Plan*. Citeseer, 1987.

[67] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Routledge Academic, 2013.

Appendix A

Cybersecurity Survey

Cyber Security Survey

Your eID: ________________________ (the first part of your email address)

What course are you taking this survey in?__________________________

This is a research study to determine your opinions and interests about cybersecurity to improve our courses and research programs. This is voluntary, and you can turn in a blank sheet if you like. If you have any questions, please ask the person administering this survey or contact Dr. Eugene Vasserman (eyv@ksu.edu). To complete the survey, for each topic/activity listed, circle the letters that match your interest in, confidence in undertaking and the time you feel it would take you to accomplish that item. For example, the first topic is "doing 1 digit multiplication," so you might select "not interested," "very confident," and that it would take "at most a few days" as shown.

		My interest in this topic					My confidence in undertaking and succeeding in this activity					Estimated time for me to prepare for and accomplish this				
		Very interested	Somewhat Interested	Not very interested	Not interested	I don't know what this is	Very confident	Somewhat confident	Not very confident	Not confident	I don't know what this is	At most a few days	A few weeks	Between a month and a year	A year or more	I wouldn't be able to do it on my own
X	Perform 1 digit multiplication	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
1	Pursue an advanced degree(s) focused on cybersecurity	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
2	Find ways to exploit vulnerabilities in existing software	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
3	Perform research focused on cybersecurity	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
4	Learn how to crack users' passwords	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
5	Take additional courses focused on cybersecurity	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
6	Discover ways to protect personal data on the Internet	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
7	Write software that is safe from buffer overflow attacks	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
8	Manage security for a Fortune 500 company	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
9	Implement a protocol to allow data to be sent securely over a network	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
10	Perform network penetration tests for companies	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
11	Learn how to use SSL certificates	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
12	Find a job which involves cybersecurity	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
13	Learn how to intercept and read network traffic	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
14	Write an algorithm that uses asymmetric encryption to authenticate a user	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
15	Work for an organization that researches ways to make computing more secure	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
16	Learn how to verify a digital signature	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
17	Have cybersecurity concepts incorporated into other courses that I take	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
18	Remove detected threats from a home computer	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
19	Read articles/web posts about cybersecurity on your own	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
20	Install and run malware checking software on a home computer	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
21	Learn how to detect cyber attacks	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
22	Find a job which is specifically oriented towards cybersecurity	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e

Appendix B

Quantitative Analysis Data

Table B.1: Average values for pre and post surveys from participants during the Spring 2014 term. Notations after the change values indicate level of statistical significance: (.) < 0.1, (*) < 0.05, (**) < 0.01, (***) < 0.001

Pursue an advanced degree(s) focused on cybersecurity	2.38 ± 0.11	2.58 ± 0.1	3.65 ± 0.1	2.44 ± 0.12	2.31 ± 0.11	3.69 ± 0.08	0.06 ± 0.24	-0.27 ± 0.21 (**)	0.04 ± 0.18
Find ways to exploit vulnerabilities in existing software	1.93 ± 0.11	2.78 ± 0.11	3.08 ± 0.11	2.08 ± 0.11	2.65 ± 0.1	2.94 ± 0.12	0.15 ± 0.22	-0.13 ± 0.22	-0.15 ± 0.23
Perform research focused on cybersecurity	2.42 ± 0.13	2.64 ± 0.11	3.2 ± 0.11	2.55 ± 0.12	2.58 ± 0.11	3.05 ± 0.11	0.12 ± 0.25 / (.)	-0.06 ± 0.23	-0.15 ± 0.21
Learn how to crack users' passwords	2.01 ± 0.12	2.47 ± 0.12	2.56 ± 0.13	1.93 ± 0.11	2.34 ± 0.1	2.47 ± 0.11	-0.08 ± 0.23	-0.12 ± 0.23	-0.09 ± 0.24
Take additional courses focused on cybersecurity	1.99 ± 0.12	2.13 ± 0.11	3.03 ± 0.09	2.21 ± 0.12	2.11 ± 0.11	3.09 ± 0.09	0.23 ± 0.23 (*)	-0.03 ± 0.22	0.05 ± 0.18
Discover ways to protect personal data on the Internet	1.65 ± 0.09	2.14 ± 0.11	2.8 ± 0.13	1.84 ± 0.11	2.22 ± 0.1	2.72 ± 0.11	0.19 ± 0.2 (*)	0.08 ± 0.21	-0.08 ± 0.24
Write software that is safe from buffer overflow attacks	2.03 ± 0.13	2.74 ± 0.12	3.23 ± 0.12	2.21 ± 0.13	2.75 ± 0.11	3.04 ± 0.11	0.18 ± 0.26	0.01 ± 0.23	-0.19 ± 0.23
Manage security for a Fortune 500 company	2.49 ± 0.13	3.03 ± 0.11	3.53 ± 0.12	2.57 ± 0.12	2.94 ± 0.1	3.46 ± 0.11	0.08 ± 0.25	-0.08 ± 0.21	-0.06 ± 0.23

Continued on next page

Table B.1 – continued from previous page

	Pre-Survey			Post-Survey			Change (p)		
	Interest	Confidence	Time	Interest	Confidence	Time	Interest	Confidence	Time
Implement a protocol to allow data to be sent securely over a network	2.29 ± 0.11	2.79 ± 0.11	3.29 ± 0.1	2.26 ± 0.12	2.58 ± 0.1	3.13 ± 0.11	-0.03 ± 0.23	-0.22 ± 0.21 (.)	-0.16 ± 0.21
Perform network penetration tests for companies	2.1 ± 0.13	2.89 ± 0.12	3.14 ± 0.11	2.23 ± 0.13	2.64 ± 0.11	3.13 ± 0.09	0.13 ± 0.26	-0.24 ± 0.23 (.)	-0.01 ± 0.21
Learn how to use SSL certificates	2.21 ± 0.14	2.54 ± 0.14	2.73 ± 0.15	2.16 ± 0.12	2.29 ± 0.12	2.75 ± 0.12	-0.04 ± 0.26	-0.25 ± 0.26 (*)	0.03 ± 0.28
Find a job which involves cybersecurity	2.42 ± 0.14	2.51 ± 0.13	3.47 ± 0.09	2.46 ± 0.13	2.47 ± 0.11	3.41 ± 0.1	0.04 ± 0.27	-0.05 ± 0.24	-0.07 ± 0.19
Learn how to intercept and read network traffic	2.13 ± 0.12	2.61 ± 0.12	3.09 ± 0.12	2.27 ± 0.13	2.41 ± 0.12	2.79 ± 0.11	0.14 ± 0.26	-0.2 ± 0.23 (*)	-0.3 ± 0.23 (**)
Write an algorithm that uses asymmetric encryption to authenticate a user	2.11 ± 0.11	2.74 ± 0.13	3.02 ± 0.13	2.44 ± 0.13	2.75 ± 0.11	2.92 ± 0.12	0.32 ± 0.24 (.)	0.01 ± 0.23	-0.11 ± 0.24
Work for an organization that researches ways to make computing more secure	2.19 ± 0.13	2.6 ± 0.12	3.53 ± 0.08	2.28 ± 0.11	2.53 ± 0.11	3.36 ± 0.1	0.09 ± 0.24	-0.07 ± 0.23	-0.17 ± 0.18
Learn how to verify a digital signature	2.14 ± 0.11	2.51 ± 0.12	3.04 ± 0.14	2.17 ± 0.12	2.41 ± 0.12	2.69 ± 0.12	0.03 ± 0.23	-0.1 ± 0.24	-0.34 ± 0.26 (**)

Continued on next page

Table B.1 – continued from previous page

	Pre-Survey			Post-Survey			Change (p)		
	Interest	Confidence	Time	Interest	Confidence	Time	Interest	Confidence	Time
Have cybersecurity concepts incorporated into other courses that I take	2.04 ± 0.13	2.25 ± 0.12	2.93 ± 0.13	2.08 ± 0.12	2.17 ± 0.11	2.85 ± 0.11	0.04 ± 0.25	-0.09 ± 0.23	-0.08 ± 0.24
Remove detected threats from a home computer	1.65 ± 0.1	1.93 ± 0.12	1.91 ± 0.12	1.61 ± 0.1	1.79 ± 0.09	1.78 ± 0.11	-0.04 ± 0.2	-0.14 ± 0.21	-0.13 ± 0.23 (.)
Read articles/web posts about cybersecurity on your own	2.4 ± 0.13	1.93 ± 0.13	1.77 ± 0.14	2.15 ± 0.12	1.79 ± 0.11	1.64 ± 0.12	-0.25 ± 0.25 (**)	-0.14 ± 0.24	-0.13 ± 0.26 (.)
Install and run malware checking software on a home computer	1.97 ± 0.12	1.59 ± 0.1	1.59 ± 0.12	1.92 ± 0.12	1.56 ± 0.1	1.53 ± 0.11	-0.05 ± 0.24	-0.04 ± 0.2	-0.07 ± 0.23
Learn how to detect cyber attacks	1.71 ± 0.1	2.35 ± 0.11	2.62 ± 0.12	1.8 ± 0.11	2.33 ± 0.1	2.57 ± 0.1	0.09 ± 0.21	-0.02 ± 0.21	-0.05 ± 0.22
Find a job which is specifically oriented towards cybersecurity	2.55 ± 0.13	2.62 ± 0.13	3.52 ± 0.1	2.52 ± 0.13	2.49 ± 0.12	3.33 ± 0.11	-0.03 ± 0.26	-0.13 ± 0.24	-0.19 ± 0.21

Table B.2: *Statistical values from Wilcoxin Signed-Rank Test and Hedges' g for each data point for students surveyed during the Spring, 2014 term. Highlighted cells indicate statistically significant results.*

	Interest				Confidence				Time			
	n	p	effect	Hedges' g	n	p	effect	Hedges' g	n	p	effect	Hedges' g
Pursue an advanced degree(s) focused on cybersecurity	73	0.514	-0.079	-0.06	**70**	**0.004**	**0.338**	**0.30**	52	0.202	-0.194	-0.06
Find ways to exploit vulnerabilities in existing software	71	0.195	-0.156	-0.16	68	0.284	0.13	0.14	54	0.268	0.148	0.16
Perform research focused on cybersecurity	73	0.09	-0.203	-0.11	71	0.433	0.096	0.06	51	0.16	0.198	0.18
Learn how to crack users' passwords	74	0.256	0.136	0.08	73	0.197	0.154	0.13	56	0.133	0.199	0.09
Take additional courses focused on cybersecurity	**75**	**0.036**	**-0.242**	**-0.22**	74	0.551	0.07	0.03	62	0.325	-0.13	-0.07
Discover ways to protect personal data on the Internet	**72**	**0.026**	**-0.264**	**-0.22**	71	0.663	-0.049	-0.09	59	0.71	0.049	0.09
Write software that is safe from buffer overflow attacks	56	0.399	-0.112	-0.17	53	0.351	0.135	-0.01	38	0.267	0.19	0.23
Manage security for a Fortune 500 company	73	0.587	-0.064	-0.08	68	0.179	0.163	0.09	32	0.73	0.081	0.08
Implement a protocol to allow data to be sent securely over a network	70	0.954	0.007	0.03	70	0.05	0.236	0.24	47	0.142	0.224	0.20
Perform network penetration tests for companies	69	0.322	-0.119	-0.12	68	0.054	0.234	0.25	46	0.115	0.246	0.01
Learn how to use SSL certificates	47	0.531	0.094	0.05	**48**	**0.048**	**0.288**	**0.26**	42	0.266	0.179	-0.03
Find a job which involves cybersecurity	71	0.714	-0.043	-0.04	69	0.364	0.112	0.05	56	0.576	0.082	0.09
Learn how to intercept and read network traffic	70	0.451	-0.092	-0.13	**64**	**0.024**	**0.28**	**0.21**	**52**	**0.006**	**0.383**	**0.33**
Write an algorithm that uses asymmetric encryption to authenticate a user	60	0.058	-0.244	-0.32	56	0.632	-0.064	-0.01	40	0.281	0.176	0.12
Work for an organization that researches ways to make computing more secure	72	0.265	-0.134	-0.09	70	0.329	0.118	0.08	48	0.411	0.126	0.24
Learn how to verify a digital signature	68	0.944	-0.006	-0.03	65	0.274	0.137	0.10	**47**	**0.005**	**0.408**	**0.34**
Have cybersecurity concepts incorporated into other courses that I take	71	0.809	-0.03	-0.04	68	0.325	0.121	0.09	47	0.351	0.138	0.08
Remove detected threats from a home computer	73	0.646	0.065	0.05	72	0.133	0.177	0.15	66	0.067	0.225	0.14
Read articles/web posts about cybersecurity on your own	**73**	**0.009**	**0.307**	**0.23**	72	0.292	0.125	0.13	65	0.082	0.217	0.12
Install and run malware checking software on a home computer	73	0.632	0.059	0.05	72	0.668	0.051	0.05	66	0.162	0.173	0.07
Learn how to detect cyber attacks	75	0.181	-0.156	-0.10	72	0.525	0.079	0.02	59	0.324	0.127	0.05
Find a job which is specifically oriented towards cybersecurity	75	0.561	0.071	0.02	70	0.108	0.194	0.13	45	0.392	0.131	0.24

Table B.3: *Average interest values for students in each course. Lower average values on pre- and post-test results indicate greater interest.*

	Pre-Test Interest			Post-Test Interest			Change		
	CIS 200	CIS 490	CIS 755	CIS 200	CIS 490	CIS 755	CIS 200	CIS 490	CIS 755
Pursue an advanced degree(s) focused on cybersecurity	2.53 ± 0.12	1.63 ± 0.38	1.8 ± 0.37	2.58 ± 0.13	1.75 ± 0.37	1.8 ± 0.37	0.05 ± 0.25	0.13 ± 0.74	0 ± 0.75
Find ways to exploit vulnerabilities in existing software	2 ± 0.12	1.5 ± 0.38	1.75 ± 0.48	2.15 ± 0.12	1.75 ± 0.37	1.8 ± 0.37	0.15 ± 0.24	0.25 ± 0.74	0.05 ± 0.85
Perform research focused on cybersecurity	2.57 ± 0.14	1.63 ± 0.38	2 ± 0.55	2.66 ± 0.12	1.88 ± 0.48	2.2 ± 0.58	0.09 ± 0.25	0.25 ± 0.85	0.2 ± 1.13
Learn how to crack users' passwords	2.06 ± 0.13	1.63 ± 0.42	2 ± 0.45	1.98 ± 0.12	1.38 ± 0.26	2.2 ± 0.49	-0.08 ± 0.25	-0.25 ± 0.68	0.2 ± 0.94
Take additional courses focused on cybersecurity	2.11 ± 0.13	1.38 ± 0.38	1.4 ± 0.24	2.27 ± 0.12	1.5 ± 0.38	2.6 ± 0.51	0.16 ± 0.25	0.13 ± 0.75	1.2 ± 0.75
Discover ways to protect personal data on the Internet	1.75 ± 0.1	1.13 ± 0.13	1.25 ± 0.25	1.92 ± 0.13	1.63 ± 0.26	1.2 ± 0.2	0.17 ± 0.23	0.5 ± 0.39	-0.05 ± 0.45
Write software that is safe from buffer overflow attacks	2.08 ± 0.14	2 ± 0.46	1.5 ± 0.29	2.28 ± 0.15	2 ± 0.42	1.8 ± 0.37	0.2 ± 0.29	0 ± 0.89	0.3 ± 0.66
Manage security for a Fortune 500 company	2.57 ± 0.14	2 ± 0.38	2.25 ± 0.63	2.65 ± 0.13	2.13 ± 0.4	2.4 ± 0.51	0.07 ± 0.27	0.13 ± 0.78	0.15 ± 1.14
Implement a protocol to allow data to be sent securely over a network	2.42 ± 0.12	1.75 ± 0.37	1.6 ± 0.4	2.39 ± 0.13	1.71 ± 0.29	1.4 ± 0.24	-0.03 ± 0.25	-0.04 ± 0.65	-0.2 ± 0.64
Perform network penetration tests for companies	2.23 ± 0.14	1.38 ± 0.38	1.8 ± 0.37	2.3 ± 0.14	1.75 ± 0.37	2.2 ± 0.49	0.07 ± 0.28	0.38 ± 0.74	0.4 ± 0.86
Learn how to use SSL certificates	2.44 ± 0.14	1.5 ± 0.38	1.5 ± 0.29	2.29 ± 0.14	1.88 ± 0.35	1.4 ± 0.24	-0.15 ± 0.28	0.38 ± 0.73	-0.1 ± 0.53
Find a job which involves cybersecurity	2.59 ± 0.15	1.38 ± 0.38	2 ± 0.55	2.62 ± 0.14	1.5 ± 0.38	2 ± 0.32	0.03 ± 0.28	0.13 ± 0.75	0 ± 0.86
Learn how to intercept and read network traffic	2.33 ± 0.13	1.25 ± 0.16	1.2 ± 0.2	2.39 ± 0.15	1.75 ± 0.41	1.6 ± 0.4	0.05 ± 0.28	0.5 ± 0.58	0.4 ± 0.6
Write an algorithm that uses asymmetric encryption to authenticate a user	2.17 ± 0.12	2 ± 0.27	1.8 ± 0.37	2.55 ± 0.15	2 ± 0.38	1.8 ± 0.37	0.39 ± 0.27	0 ± 0.65	0 ± 0.75
Work for an organization that researches ways to make computing more secure	2.35 ± 0.14	1.5 ± 0.38	1.4 ± 0.24	2.41 ± 0.12	1.88 ± 0.3	1.4 ± 0.24	0.06 ± 0.26	0.38 ± 0.67	0 ± 0.49
Learn how to verify a digital signature	2.22 ± 0.13	1.88 ± 0.35	1.6 ± 0.24	2.25 ± 0.13	1.88 ± 0.4	1.6 ± 0.4	0.03 ± 0.25	0 ± 0.75	0 ± 0.64
Have cybersecurity concepts incorporated into other courses that I take	2.17 ± 0.14	1.63 ± 0.42	1 ± 0	2.19 ± 0.14	1.63 ± 0.32	1.4 ± 0.24	0.02 ± 0.28	0 ± 0.74	0.4 ± 0.24
Remove detected threats from a home computer	1.69 ± 0.11	1.5 ± 0.38	1.4 ± 0.24	1.62 ± 0.1	1.5 ± 0.38	1.6 ± 0.4	-0.07 ± 0.22	0 ± 0.76	0.2 ± 0.64
Read articles/web posts about cybersecurity on your own	2.6 ± 0.14	1.5 ± 0.38	1.4 ± 0.24	2.31 ± 0.13	1.38 ± 0.26	1.4 ± 0.24	-0.29 ± 0.27	-0.13 ± 0.64	0 ± 0.49
Install and run malware checking software on a home computer	1.95 ± 0.13	1.88 ± 0.4	2.4 ± 0.68	1.87 ± 0.12	2 ± 0.42	2.4 ± 0.51	-0.08 ± 0.25	0.13 ± 0.82	0 ± 1.19
Learn how to detect cyber attacks	1.81 ± 0.11	1.38 ± 0.38	1 ± 0	1.87 ± 0.12	1.5 ± 0.38	1.4 ± 0.24	0.06 ± 0.23	0.13 ± 0.75	0.4 ± 0.24
Find a job which is specifically oriented towards cybersecurity	2.74 ± 0.13	1.38 ± 0.38	2 ± 0.55	2.65 ± 0.14	1.75 ± 0.41	2.2 ± 0.49	-0.1 ± 0.27	0.38 ± 0.79	0.2 ± 1.04

Table B.4: *Average confidence values for students in each course. Lower average values on pre- and post-test results indicate greater confidence.*

	Pre-Test Confidence			Post-Test Confidence			Change		
	CIS 200	CIS 490	CIS 755	CIS 200	CIS 490	CIS 755	CIS 200	CIS 490	CIS 755
Pursue an advanced degree(s) focused on cybersecurity	2.6 ± 0.12	2.38 ± 0.26	2.6 ± 0.4	2.38 ± 0.11	2 ± 0.46	2 ± 0.45	-0.23 ± 0.23	-0.38 ± 0.73	-0.6 ± 0.85
Find ways to exploit vulnerabilities in existing software	2.82 ± 0.13	2.63 ± 0.26	2.5 ± 0.29	2.71 ± 0.12	2.5 ± 0.33	2.2 ± 0.37	-0.11 ± 0.25	-0.13 ± 0.59	-0.3 ± 0.66
Perform research focused on cybersecurity	2.73 ± 0.12	2.13 ± 0.35	2.4 ± 0.51	2.71 ± 0.12	1.88 ± 0.35	2.2 ± 0.58	-0.02 ± 0.24	-0.25 ± 0.7	-0.2 ± 1.09
Learn how to crack users' passwords	2.56 ± 0.14	1.88 ± 0.23	2.2 ± 0.58	2.42 ± 0.11	1.88 ± 0.35	2.2 ± 0.49	-0.15 ± 0.25	0 ± 0.58	0 ± 1.07
Take additional courses focused on cybersecurity	2.27 ± 0.12	1.38 ± 0.18	1.6 ± 0.4	2.23 ± 0.12	1.5 ± 0.27	1.6 ± 0.4	-0.04 ± 0.24	0.13 ± 0.45	0 ± 0.8
Discover ways to protect personal data on the Internet	2.2 ± 0.13	1.63 ± 0.18	2.25 ± 0.25	2.32 ± 0.1	2 ± 0.38	1.4 ± 0.24	0.12 ± 0.23	0.38 ± 0.56	-0.85 ± 0.49
Write software that is safe from buffer overflow attacks	2.84 ± 0.13	2.63 ± 0.26	1.75 ± 0.48	2.84 ± 0.12	2.63 ± 0.38	1.75 ± 0.25	0.01 ± 0.24	0 ± 0.64	0 ± 0.73
Manage security for a Fortune 500 company	3.07 ± 0.12	2.88 ± 0.13	2.5 ± 0.5	3.02 ± 0.11	2.5 ± 0.33	2.8 ± 0.58	-0.05 ± 0.23	-0.38 ± 0.45	0.3 ± 1.08
Implement a protocol to allow data to be sent securely over a network	2.85 ± 0.12	2.5 ± 0.33	2.6 ± 0.4	2.67 ± 0.11	2.5 ± 0.27	1.6 ± 0.24	-0.18 ± 0.23	0 ± 0.59	-1 ± 0.64
Perform network penetration tests for companies	2.95 ± 0.14	2.5 ± 0.38	2.67 ± 0.67	2.7 ± 0.11	2.5 ± 0.38	2.2 ± 0.37	-0.25 ± 0.25	0 ± 0.76	-0.47 ± 1.04
Learn how to use SSL certificates	2.76 ± 0.16	2.13 ± 0.3	1.25 ± 0.25	2.44 ± 0.13	2 ± 0.38	1.2 ± 0.2	-0.32 ± 0.28	-0.13 ± 0.67	-0.05 ± 0.45
Find a job which involves cybersecurity	2.68 ± 0.13	1.5 ± 0.19	2 ± 1	2.52 ± 0.12	2 ± 0.33	2.5 ± 0.65	-0.15 ± 0.25	0.5 ± 0.52	0.5 ± 1.65
Learn how to intercept and read network traffic	2.68 ± 0.13	2.63 ± 0.26	1.8 ± 0.58	2.54 ± 0.12	1.88 ± 0.35	1.75 ± 0.48	-0.15 ± 0.25	-0.75 ± 0.61	-0.05 ± 1.06
Write an algorithm that uses asymmetric encryption to authenticate a user	2.8 ± 0.14	2.71 ± 0.29	2.2 ± 0.58	2.84 ± 0.12	2.63 ± 0.32	1.75 ± 0.48	0.04 ± 0.26	-0.09 ± 0.61	-0.45 ± 1.06
Work for an organization that researches ways to make computing more secure	2.68 ± 0.13	2.25 ± 0.25	2.2 ± 0.58	2.63 ± 0.11	2.13 ± 0.44	2 ± 0.45	-0.06 ± 0.24	-0.13 ± 0.69	-0.2 ± 1.03
Learn how to verify a digital signature	2.66 ± 0.13	2 ± 0.27	1.5 ± 0.29	2.58 ± 0.12	2 ± 0.38	1.2 ± 0.2	-0.08 ± 0.25	0 ± 0.65	-0.3 ± 0.49
Have cybersecurity concepts incorporated into other courses that I take	2.35 ± 0.13	1.63 ± 0.18	2 ± 0.58	2.29 ± 0.12	1.63 ± 0.38	1.6 ± 0.4	-0.06 ± 0.25	0 ± 0.56	-0.4 ± 0.98
Remove detected threats from a home computer	2.03 ± 0.14	1.5 ± 0.19	1.4 ± 0.24	1.9 ± 0.1	1.38 ± 0.26	1.2 ± 0.2	-0.13 ± 0.24	-0.13 ± 0.45	-0.2 ± 0.44
Read articles/web posts about cybersecurity on your own	2.08 ± 0.15	1.25 ± 0.16	1.2 ± 0.2	1.9 ± 0.12	1.5 ± 0.38	1 ± 0	-0.18 ± 0.27	0.25 ± 0.54	-0.2 ± 0.2
Install and run malware checking software on a home computer	1.69 ± 0.12	1.25 ± 0.16	1 ± 0	1.61 ± 0.11	1.38 ± 0.26	1.2 ± 0.2	-0.08 ± 0.23	0.13 ± 0.43	0.2 ± 0.2
Learn how to detect cyber attacks	2.44 ± 0.13	2 ± 0.19	1.75 ± 0.48	2.45 ± 0.1	1.88 ± 0.4	1.6 ± 0.4	0.01 ± 0.23	-0.13 ± 0.59	-0.15 ± 0.88
Find a job which is specifically oriented towards cybersecurity	2.77 ± 0.13	1.75 ± 0.25	2 ± 0.71	2.58 ± 0.12	2 ± 0.46	2.2 ± 0.58	-0.19 ± 0.25	0.25 ± 0.71	0.2 ± 1.29

Table B.5: *Statistical calculations from Wilcoxin Signed-Rank Test for each data point for students in CIS 200 course*

	Interest				Confidence				Time			
	n	p	effect	Hedges' g	n	p	effect	Hedges' g	n	p	effect	Hedges' g
Pursue an advanced degree(s) focused on cybersecurity	60	0.718	-0.049	-0.048	58	0.043	0.276	0.258	45	0.459	-0.136	-0.077
Find ways to exploit vulnerabilities in existing software	59	0.401	-0.109	-0.159	57	0.529	0.091	0.119	47	0.701	0.057	0.085
Perform research focused on cybersecurity	60	0.216	-0.171	-0.095	60	0.817	0.037	0.023	49	0.76	0.046	0.054
Learn how to crack users' passwords	62	0.395	0.118	0.083	62	0.216	0.161	0.15	48	0.398	0.125	0.043
Take additional courses focused on cybersecurity	62	0.214	-0.165	-0.164	62	0.501	0.087	0.048	53	0.126	-0.22	-0.193
Discover ways to protect personal data on the Internet	60	0.111	-0.214	-0.189	61	0.665	-0.057	-0.132	53	0.927	0.02	0.054
Write software that is safe from buffer overflow attacks	49	0.436	-0.115	-0.188	49	0.28	0.167	-0.007	37	0.284	0.2	0.356
Manage security for a Fortune 500 company	61	0.817	-0.036	-0.068	61	0.453	0.103	0.054	32	1	0	0.025
Implement a protocol to allow data to be sent securely over a network	59	1	0.002	0.031	59	0.204	0.174	0.203	39	0.447	0.144	0.224
Perform network penetration tests for companies	57	0.733	-0.048	-0.066	59	0.088	0.229	0.258	41	0.21	0.219	-0.034
Learn how to use SSL certificates	36	0.341	0.17	0.167	38	0.062	0.315	0.35	32	0.503	0.138	-0.038
Find a job which involves cybersecurity	59	0.96	-0.008	-0.027	59	0.103	0.221	0.157	48	0.97	0.027	0.03
Learn how to intercept and read network traffic	57	1	0	-0.05	57	0.126	0.212	0.156	46	0.035	0.33	0.357
Write an algorithm that uses asymmetric encryption to authenticate a user	48	0.026	-0.326	-0.38	49	0.582	-0.081	-0.044	34	0.911	0.037	-0.023
Work for an organization that researches ways to make computing more secure	60	0.49	0.093	-0.059	60	0.624	0.071	0.06	45	0.824	0.053	0.221
Learn how to verify a digital signature	58	0.988	0.01	-0.031	58	0.375	0.123	0.079	45	0.073	0.278	0.277
Have cybersecurity concepts incorporated into other courses that I take	59	1	-0.003	-0.022	60	0.493	0.097	0.065	48	0.4	0.13	0.098
Remove detected threats from a home computer	61	0.559	0.085	0.077	61	0.246	0.156	0.142	55	0.165	0.191	0.121
Read articles/web posts about cybersecurity on your own	60	0.016	0.319	0.28	61	0.291	0.139	0.173	58	0.32	0.134	0.061
Install and run malware checking software on a home computer	60	0.499	0.095	0.082	61	0.49	0.09	0.088	56	0.377	0.123	0.04
Learn how to detect cyber attacks	62	0.504	0.09	-0.071	62	0.856	0.038	-0.016	52	1	0	-0.045
Find a job which is specifically oriented towards cybersecurity	62	0.305	0.137	0.091	61	0.127	0.201	0.198	43	0.257	0.196	0.355

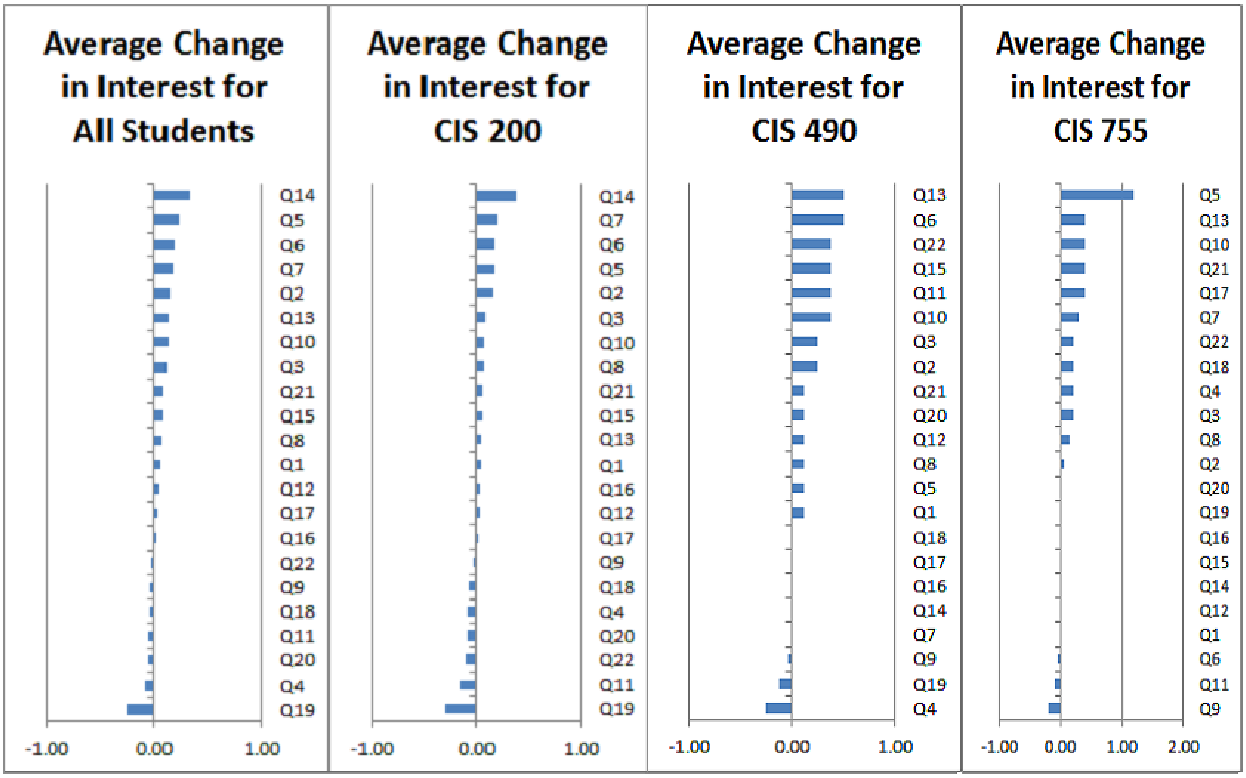

Figure B.1: *Average change in student interest for each survey question from pre- to post-survey. Note that a positive change indicates student interest decreased.*

Q1	Pursue an advanced degree(s) focused on cybersecurity
Q2	Find ways to exploit vulnerabilities in existing software
Q3	Perform research focused on cybersecurity
Q4	Learn how to crack users' passwords
Q5	Take additional courses focused on cybersecurity
Q6	Discover ways to protect personal data on the Internet
Q7	Write software that is safe from buffer overflow attacks
Q8	Manage security for a Fortune 500 company
Q9	Implement a protocol to allow data to be sent securely over a network
Q10	Perform network penetration tests for companies
Q11	Learn how to use SSL certificates
Q12	Find a job which involves cybersecurity
Q13	Learn how to intercept and read network traffic
Q14	Write an algorithm that uses asymmetric encryption to authenticate a user
Q15	Work for an organization that researches ways to make computing more secure
Q16	Learn how to verify a digital signature
Q17	Have cybersecurity concepts incorporated into other courses that I take
Q18	Remove detected threats from a home computer
Q19	Read articles/web posts about cybersecurity on your own
Q20	Install and run malware checking software on a home computer
Q21	Learn how to detect cyber attacks
Q22	Find a job which is specifically oriented towards cybersecurity

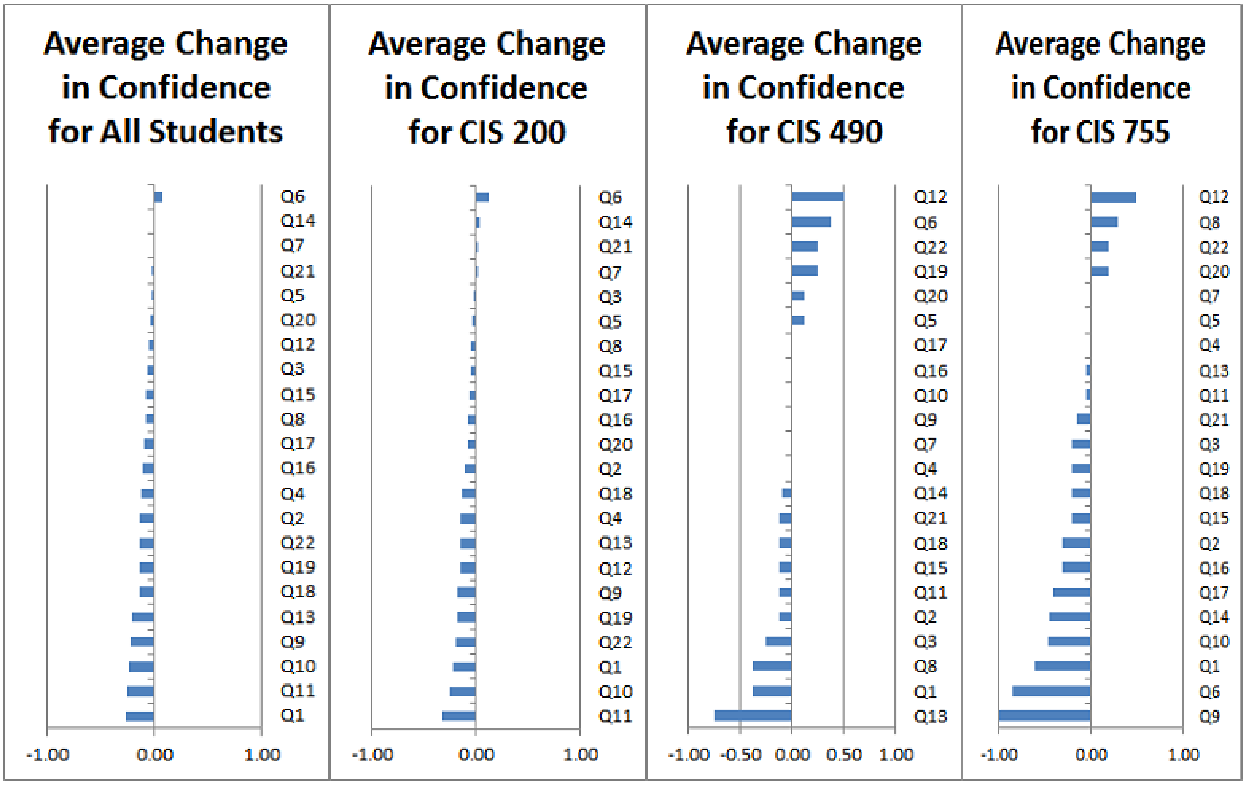

Figure B.2: *Average change in student confidence for each survey question from pre- to post-survey. Note that a positive change indicates student confidence decreased.*

Q1	Pursue an advanced degree(s) focused on cybersecurity
Q2	Find ways to exploit vulnerabilities in existing software
Q3	Perform research focused on cybersecurity
Q4	Learn how to crack users' passwords
Q5	Take additional courses focused on cybersecurity
Q6	Discover ways to protect personal data on the Internet
Q7	Write software that is safe from buffer overflow attacks
Q8	Manage security for a Fortune 500 company
Q9	Implement a protocol to allow data to be sent securely over a network
Q10	Perform network penetration tests for companies
Q11	Learn how to use SSL certificates
Q12	Find a job which involves cybersecurity
Q13	Learn how to intercept and read network traffic
Q14	Write an algorithm that uses asymmetric encryption to authenticate a user
Q15	Work for an organization that researches ways to make computing more secure
Q16	Learn how to verify a digital signature
Q17	Have cybersecurity concepts incorporated into other courses that I take
Q18	Remove detected threats from a home computer
Q19	Read articles/web posts about cybersecurity on your own
Q20	Install and run malware checking software on a home computer
Q21	Learn how to detect cyber attacks
Q22	Find a job which is specifically oriented towards cybersecurity