12-2014

Serious Toys: Teaching Computer Science Concepts to Pre-Collegiate Students

Yvon Feaster
Clemson University, yfeaste@clemson.edu

Serious Toys: Teaching Computer Science Concepts to Pre-Collegiate Students

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Computer Science

by
Yvon H. Feaster
December 2014

Accepted by:
Dr. Jason O. Hallstrom, Committee Chair
Dr. Timothy A. Davis
Dr. Brian A. Malloy
Dr. Murali Sitaraman

Abstract

Advancements in science and engineering have driven innovation in the United States for more than two centuries. The last several decades have brought to the forefront the importance of such innovation to our domestic and global economies. To continue to succeed in this information-based, technologically advanced society, we must ensure that the next generation of students are developing computational thinking skills beyond what was acceptable in past years. Computational thinking represents a collection of structured problem solving skills that cross-cut educational disciplines. There is significant future value in introducing these skills as early as practical in students' academic careers. Over the past four years, we have developed, piloted, and evaluated a series of outreach modules designed to introduce fundamental computing concepts to young learners. Each module is based on a small embedded device —a *"serious toy"*— designed to simultaneously engage visual, auditory, and kinesthetic learners through lectures, visual demonstrations, and hands-on activities. We have piloted these modules with more than 770 students, and the evaluation results show that the program is having a positive impact. The evaluation instruments for our pilots consist of pre- and post-attitudinal surveys and pre- and post-quizzes. The surveys are designed to assess student attitudes toward computer science and student self-efficacy with respect to the material covered. The quizzes are designed to assess students' content understanding. In this dissertation, we describe the modules and associated serious toys. We also describe the module evaluation methods, the pilot groups, and the results for each pilot study.

Dedication

For Toby, Russell, and Katie

Acknowledgments

Over the past five years, I have been encouraged and inspired by so many supportive individuals as I made my way on this educational journey. While I am thankful for them all, there are a few that I would especially like to acknowledge. I express my sincere gratitude and appreciation to Dr. Jason Hallstrom for the guidance and wisdom he has provided as my advisor and mentor. During the most challenging days, he helped me believe in myself because I knew that he believed in me.

I also would like to thank my committee members, Dr. Timothy Davis, Dr. Brian Malloy, and Dr. Murali Sitaraman. They have been a great source of inspiration to me and have provided invaluable assistance and encouragement.

I would like to thank Farha Ali and Jiannan Zhai who not only provided hours of assistance and encouragement but also invaluable friendship. The journey was so much more enjoyable through our shared experiences.

Finally, I would like to thank my husband, Toby Feaster, and my children, Russell and Katie Feaster. Without their encouragement, patience, and support this would not have been possible. They are the joy of my life.

This work was supported by National Science Foundation CAREER grant CNS-0745846 and the Graduate Research Fellowship Program DGE-075127.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the last several decades, computing technology has become fundamental to society's growth and economic expansion. According to a recent report published by the U.S. Congress Joint Economic Committee [22], at least half of the country's economic growth is attributed to technological advances, with concomitant growth in STEM-related jobs. However, the U.S. ranks 27^{th} out of 30 industrialized countries in the percentage of STEM bachelor degrees awarded. Numerous computing-related outreach programs have been developed to address the shortfall of computing degrees in the US [28, 50, 60, 86, 96]. These efforts have had an impact. According to the most recent Taulbee Survey [97], the number of bachelor degrees awarded in Computer Science (CS) in the U.S. experienced increases for three consecutive years, beginning in the 2009-2010 academic year. This is encouraging news. However, considering that undergraduate CS degrees awarded during the first decade of this century decreased by more than 50% [97], coupled with the 24% projected growth in computer-related jobs by 2020 [15], there is still a potential shortfall in the domestic labor market.

While a CS degree is not for all students, rapid technological advances demand that every student be introduced to computing at some level, because daily lives of most will be heavily influenced by computing [7]. Therefore, computer science educators must find new ways to reach out to potential CS recruits, as well as introduce students to computational skills.

1.1 Problem Statement

Structured education begins with the K-12 curriculum. For more than three decades, the third grade has been recognized as a milestone year in gauging students' future academic success. Reading comprehension has been a focal point. A recent national study commissioned by the Annie E. Casey Foundation [40, 70] reports that students who lag behind in reading performance at the end of the third grade are four times less likely to earn a high school diploma. Those with the lowest reading performance are six times less likely to earn a diploma. The explanation for this trend is simple: Third grade marks a transition from "learning to read" to "reading to learn". Reading comprehension skills are assumed in subsequent grades, and independent learning outside of the classroom is emphasized.

We posit a connection between reading comprehension skills and computational thinking skills. As early as kindergarten, many students begin to use computers in their classrooms. As they progress to more advanced grades, computers become essential tools to facilitate learning. But the importance of computing does not end there; computing is more than learning to use a computer. Computational thinking represents a discipline of structured problem solving, encompassing skills that are broadly useful to students throughout their academic careers [93]. As the academic challenges students encounter become progressively more complex, students who possess strong computational thinking skills are likely to outperform those who do not. It is safe to assume that there is a "pivot year", similar to the reading comprehension scenario, where these skills become essential to more advanced learning. While there is no evidence available to identify the particular year when this occurs, it would be prudent to introduce computational thinking as early as practical in the K-12 curriculum.

In this dissertation, we set out to investigate a new approach to teaching CS concepts with tailored curriculum modules and embedded computing manipulatives. The results are expected to be broadly applicable across topic areas within computing. We focus on three in this initial exploration, selected because they are among the most important computing concepts today.

1.2 Approach

In the United States, computing curriculum is virtually nonexistent in the K-12 system [91]. To address this deficiency, we have developed a new teaching approach based on the use of tailored curriculum content and supporting embedded manipulatives designed to teach focused computing concepts. The approach targets pre-colligiate students using focused instructional outreach sessions. The instructional style is designed to simultaneously engage visual, auditory, and kinesthetic learners through lectures, visual demonstrations, and hands-on activities. The approach is designed to be broadly applicable across concepts and topic areas. However, we have selected three specific topics which have been identified as being difficult to teach and learn, or which are particularly relevant to modern computing applications.

The first module introduces the fundamentals of the binary number system. The second is focused on networks, protocols, and algorithms. The third centers on sensors, sensor networks, and their significance in today's society. Although the modules are designed to enable incremental, independent adoption, together they form a coherent thread of instruction.

1.2.1 Modules

We present curriculum modules covering (i) binary number systems; (ii) networks, protocols, and algorithms; and (iii) sensors and sensor networks.

Module 1 – *The Binary Number System.* Binary is one of the first topics introduced in the undergraduate curriculum because it is fundamental to so many subsequent topic areas. Proficiency in binary arithmetic is a requisite skill in the study of computer architecture, data storage, networking, and myriad other content areas.

Unfortunately, the binary number system is a subject that is often laborious to teach and learn. A number of experience reports note disproportionately low student engagement and satisfaction when compared to other topics [31, 50]. Motivating young learners to learn the binary number system is often more difficult than motivating students to learn robotics or computer graphics. Topics like robotics and computer graphics have a strong visual component and naturally appeal to visual learners. Similarly, these topics lend themselves to hands-on activities that provide immediate visual and/or tactile feedback, appealing to kinesthetic learners. Traditionally, modules to teach binary conversion and arithmetic lack both visual and kinesthetic appeal, resulting in student and teacher disinterest in a subject that is fundamental to computing.

To introduce the binary number system in the K-12 curriculum, we have developed a curriculum module designed to appeal to all students, regardless of their preferred learning styles:

visual, auditory, or kinesthetic. The module begins with a review of the base-10 number system. Using ten fingers, we illustrate that base-10 can represent 9 objects before a new digit column must be added. We further illustrate that ten groups of ten objects require a third column to be added, and so on. Using this same illustration, we introduce the concept of binary, including the process of converting decimal numbers to binary numbers and vice-versa, as well as binary arithmetic. Next, we introduce students to the "serious toy", providing them conversion, addition, and subtraction problems to solve using the toy. We conclude with a discussion on the importance of the binary number system in computer science. The primary learning objectives for this module call for students (i) to learn to convert decimal numbers to binary numbers and vice-versa; (ii) to learn to add two binary numbers of arbitrary length; (iii) to learn to subtract two binary numbers of arbitrary length; (iv) to understand the relevance of binary numbers to computer science; and (v) to learn to apply the concepts learned to number systems of arbitrary bases.

According to the National Council of Teachers of Mathematics [66], number systems should be introduced to students as early as kindergarten. Similarly, K-12 curriculum recommendations published by the Computer Science Teachers Association [87] include an introduction to binary number concepts for students as young as K-3. Their report indicates that by the end of grade nine, all students should have a basic understanding of binary numbers and their importance to computer science. Our module provides an effective means to achieve these goals.

Module 2 – ***Networks, Protocols, and Algorithms.*** Networking concepts have been ubiquitous, if only implicitly, for centuries. The human body is a network of organs that must coordinate to survive. The postal service is an example of a network that connects individuals world-wide. Today, networks play an important role in computing — from networks of sensors collecting and recording data, to social networks, to the most complex (non-organic) network of all, the Internet. Like the postal service, the Internet connects people throughout the world, providing a fast and efficient gateway for data transmission as well as face-to-face communication.

Observing the importance of networking concepts in computing, we have developed a curriculum module designed to introduce pre-collegiate students to the fundamentals of networks, protocols, and algorithms. We begin the module by defining a *network* as a group of communicating entities. We then ask students to identify familiar networks (*e.g.*, family, friends, and social networks). To prompt students to consider how a network operates to deliver information, we ask students to suggest a network that connects people world-wide. As expected, students most often suggest the Postal Service and the Internet. We then provide a comparison of the delivery of mail through the Postal Service with the delivery of email through the Internet. During this discussion, we define a *protocol* as a set of rules. Again, using the post office as an example, we discuss the importance of postal workers following specified protocols to assure the delivery of mail. We then relate these concepts to the delivery of email. Next, we define *algorithms* as a clear set of instructions and invite students to discuss the steps necessary to complete a specific task, such as programming a washing machine to wash a load of clothes, or baking a cake. Using these examples allows us to relate the importance of following step-by-step instructions. We illustrate how the instructions in a program must often be completed in a specific order — a washing machine cannot process a spin cycle before adding water. We also illustrate how some steps can be performed in multiple ways, without concern for operation ordering — adding milk to cake batter before adding eggs is perfectly acceptable.

Students are next given a set of the “serious toys” developed for this module and instructed how to connect them to form a network. As a demonstration of network robustness, students are encouraged to add and delete nodes from the network. Students are then introduced to a network of sensor nodes and allowed to interact with the sensors. Through the use of a desktop application, students are able to visualize the sensor data being collected and communicated. Finally, students participate in a game designed to demonstrate the importance of giving a robot concise, step-by-step instructions — an *algorithm*. The primary learning objectives for this module call for students (i) to understand the definitions of a network, a protocol, and an algorithm; (ii) to understand how networks, protocols, and algorithms are related; and (iii) to understand the relevance of networks, protocols, and algorithms in computer science.

As infants, we instinctively know how to communicate our needs for basic necessities. As we mature, we become part of various personal and professional networks, where we intuitively learn and adapt to the proper communication protocols for each network. Just as networks and protocols are common aspects of our daily lives, they are also important to many aspects of computing. Considering the rapid advancements in computing technology and a growing reliance on the Internet, young learners should have a basic understanding of computer networks, protocols, and algorithms. This module offers an effective means of teaching students the basics of these concepts.

Module 3 – ***Sensors and Sensor Networks.*** The use of sensors has become pervasive in many aspects of our daily lives. Virtually all appliances in our homes, businesses, and schools include one or more sensors. Networks of sensors are used in applications spanning home and medical monitoring [8,43], data collection for the military [33,49], and environmental management [29]. Just as the human body is a network of organs, it is also a network of sensors. To name a few, the human eye reacts to changes in light, the ear to vibrations, and the nose to vapors.

Recognizing the importance of sensors and building on the Networks, Protocols, and Algorithms module, we developed a module to introduce students to sensors and sensor networks. Motivating students to consider various sensors, we begin by asking them to identify familiar sensors, and then discussing these sensors and their associated uses — namely, to measure physical or environmental conditions. To communicate the importance of sensors, we discuss various types of sensors, networks of sensors, and how their applications continue to improve the quality of our daily lives. One sensor we discuss is a passive infrared (PIR) motion sensor. We discuss common applications of PIR sensors, and then present a demonstration of a PIR motion sensor connected to an LCD display that provides a count of the number of times the sensor detects motion. Next, we assist students in making a basic moisture sensor, as adapted from [14]. Finally, using a set of MoteStacks [29], we assist students in testing their moisture sensors. Students observe data being collected from their sensors using a web-based application. This exercise allows each student to compare the readings of her homemade sensor with those of other classmates, and to discuss explanations for anomalies in their observations. The learning objectives for this module call for students (i) to have a basic understanding of how sensors work, (ii) to understand the construction of basic sensors, (iii) to be familiar with various types of sensors, and (iv) to understand the relevance of sensors and sensor networks in computer science.

A significant recent thrust in computing centers on solving "real world" problems. Many such problems involve sensors. In any given day, we typically encounter numerous sensors. Understanding sensing concepts allows us to recognize that the automatic door at the supermarket opens due, in part, to a sensor, and that every sound we hear is the result of an internal sensor. In addition to recognition, having a basic understanding of how sensors work provides students the knowledge needed to reason about what change was detected when the automatic door opened, or how a car automatically identifies the need to activate the passenger-side airbag. This module provides students with a basic understanding of sensors and networks of sensors.

Combined Program. Three curriculum modules were designed to enable incremental, independent adoption. However, when combined they form a coherent thread of instruction. When presented in the order outlined above, the latter modules build on the former. As an example, the discussion on the process of sending an email presented in the Networks, Protocols, and Algorithms module allows us to reiterate the importance of binary numbers in computing. Presented as part of the Sensors and Sensor Networks module discussion and demonstration of networked sensors, we are able to further reinforce the concepts covered in the Binary Number System and Networks, Protocols, and Algorithms modules. As an example, we remind students that data from a sensor is ultimately stored on a computer using a sequence of 0's and 1's. We revisit the definitions of networks, protocols, and algorithms. We also reiterate the importance of binary numbers, networks, protocols, and algorithms in CS.

Our experimentation shows that each of the above modules has had a positive impact on student interest in computer science. The modules are designed to enable content understanding in at least the first three categories of Bloom's Taxonomy of Educational Objectives [11]: i) knowledge, ii) comprehension, iii) application, iv) analysis, v) synthesis, and vi) evaluation. Indeed, for each module, students often exhibited understanding beyond the first three categories.

1.3 Pilot Groups

The pilot groups were chosen from local middle and high schools, as well as students participating in summer outreach programs at Clemson University. The target groups were students perceived to be of average ability within their respective programs.

The (i) Binary Number System (BNS); (ii) Networks, Protocols, and Algorithms (NPA); and (iii) Sensor and Sensor Networks (SSN) modules were individually piloted with over 500 students — six groups of high school students, and eleven groups of middle school students. Five of the high school groups were participants in a summer program sponsored by Clemson University. The sixth high school group consisted of students from a local high school statistics class. All high school students were juniors and seniors. The middle school groups consisted of students from two local middle schools. One participating school provided six groups, consisting of two classes each of sixth, seventh, and eighth grade students. The second participating school provided five single-gendered groups, consisting of four classes of seventh, and one class of eighth grade students. Two of the seventh grade classes were all females; the remaining classes were all males. The combined program was piloted with approximately 270 students — eleven groups from two local middle schools, and one group from a local high school. The first middle school pilot group consisted of two classes each of sixth, seventh, and eighth grade students. The second middle school pilot group consisted of five single-gendered classes, two groups of sixth grade boys, and one each of seventh grade boys, sixth grade girls, and seventh grade girls. The high school group consisted of eighth, ninth, and tenth grade students. Overall, the work presented in this dissertation impacted approximately 770 students, spanning grades sixth through twelfth.

1.4 Evaluation

Each pilot was evaluated using pre- and post-surveys, as well as content quizzes. The surveys consisted of Likert-style statements designed to evaluate students' attitudes toward computer science, and students' self-efficacy with respect to the material presented. To determine students' content understanding, we administered pre- and post-quizzes consisting of true/false, multiple choice, and fill-in-the-blank questions. In each pilot, statistical analysis was completed to determine if there was a significant change in the pre- and post data. The preliminary evaluations showed mostly positive results. In addition to the individual pilots, we piloted and evaluated the three modules as a combined unit of instruction. The evaluation instruments again consisted of pre- and post-surveys and pre- and post-quizzes, with additional module-specific quizzes introduced after the completion of each module.

The evaluation results are largely positive. One particularly exciting outcome is that the proposed teaching approach, focused on serious toys, applies equally well both to high performers and low performers. This is detailed in the evaluation results for our pilot studies.

1.5 Novelty

The work presented in this dissertation describes a novel approach to introducing computer science concepts to pre-collegiate students. While numerous programs have been developed to introduce young learners to computer science, e.g., [10, 60, 76, 86], there are few that focus on specific computer science concepts. Many concentrate on teaching programming using a range of programming environments. The work described here involves curriculum modules designed to introduce students to specific computer science concepts, not just programming. Further, the approach itself is novel. Each module relies on a digital manipulative[1] –a serious toy– to reinforce the concept being taught. Manipulatives have been used in the classroom since Friedrich Froebel founded the "children's garden," better known as "kindergarten" in 1837. Froebel provided his students with a set of "gifts" –balls, blocks, and sticks– to encourage learning through playing [74, 92]. There are numerous programs that use platforms such as computers, tablets, video games, and mobile devices, to introduce computer science to pre-collegiate students, e.g., [16, 25, 59, 77]. There are also programs designed to teach a range of STEM technologies that rely on digital manipulatives, including, Lego®NXT [3], Makey Makey® [82,85], LittleBits® [9,42], and Blockuit [71]. The component that makes this program unique is that we tightly integrate our curriculum modules, which teach specific CS concepts, with digital manipulatives designed specifically to support the curriculum module.

1.6 Dissertation Organization

The remainder of the dissertation is organized as follows. Chapter 2 describes the *The Binary Number System* module. Chapter 3 describes the *Networks, Protocols, and Algorithms* module. Chapter 4 discusses *Sensors and Sensor Networks* module. Chapter 5 presents the *Combined Program*. Chapter 6 presents related work in the area. Finally, Chapter 7 concludes with a summary of contributions and pointers to future work.

[1] In 1998, Resnick [74] defined digital manipulative as "computationally-enhanaced versions of traditional children's toys." Similarly, in 2006 Raffle [73] said, "Digital manipulatives embed computation in familiar children's toys." However, Bennet [1] characterized a web-based application as a digital manipulative. For the purpose of this document, similar to Resnick and Raffle, we consider a digital manipulative to be a small embedded device.

Chapter 2

The Binary Number System

Motivated by the importance of binary in computing, we chose the binary number system as the topic of our first module. The binary number system is the lingua franca of computing, requisite to myriad areas, from hardware architecture and data storage to wireless communication and algorithm design. Given its significance to such a broad range of computing topics, it is not surprising that the binary number system plays a prominent role in K-12 outreach efforts [20,24,37]. It is even less surprising that the topic is often viewed as a dreary introduction to the discipline. Perception of the binary number system being inherently difficult, not only to learn, but also to teach at the K-12 level, has made it challenging to motivate K-12 institutions to include the topic in the curriculum. However, the National Council of Teachers of Mathematics [66] recommends that basic principles of number systems should be taught as early as kindergarten. The Computer Science Teachers Association similarly recommends including binary in the K-12 curriculum [87], including an introduction to binary for K-3 students. Their recommendation is that by completion of grade 9, all students should be familiar with binary numbers.

To gauge computing educators' perceived need for teaching binary to potential computing students, we developed a brief survey and sent it to 30 computing educators at 6 external institutions. The survey is shown in Listing 2.1. Participants were asked to rate their level of agreement with each of the five statements, choosing from *strongly disagree, disagree, moderately disagree, moderately agree, agree, and strongly agree.* We received 10 responses. While the sample size is too small to draw definitive conclusions, the results are informative, if only anecdotally: Table 2.1 shows that, overall, the results were in the affirmative regarding the need for computing students to understand binary.

```
1. It is important for computing students to understand number
   systems other than base-10.
2. It is advantageous for incoming computing students to understand
   number systems other than base-10.
3. The binary number system is important throughout computing.
4. The binary number system is important in the course(s) I teach.
5. Learning the binary number system can be fun for students.
```

Listing 2.1: Survey Statements

One interesting observation is that while 30% of respondents indicated that binary is relatively unimportant in the courses they teach (statement 4), *all* respondents agreed that understanding number systems other than base-10 is important for computing students (statement 1).

Table 2.1: Computing Educator Survey Results

Choices	**S1**	**S2**	**S3**	**S4**	**S5**
Strongly Agree	9	5	7	3	3
Moderately Agree	1	3	2	2	1
Agree	0	2	1	2	5
Disagree	0	0	0	1	1
Moderately Disagree	0	0	0	1	0
Strongly Disagree	0	0	0	1	0

2.1 Binary Module

To support the introduction of binary arithmetic in the pre-collegiate curriculum in a manner that is both informative and engaging, we have developed a new approach to teaching the topic. Our module spans two 60-minute class periods. The first session includes a lecture, facilitated by a series of questions and demonstrations designed to engage students. The second session provides hands-on activities using a serious toy designed to reinforce the lecture content. There are five primary learning objectives for this module. The learning objectives call for students (i) to learn to convert decimal numbers to binary numbers and vice-versa; (ii) to learn to add two binary numbers of arbitrary length; (iii) to learn to subtract two binary numbers of arbitrary length; (iv) to understand the relevance of binary numbers to computer science; and (v) to learn to apply the concepts learned to number systems of arbitrary bases. Appendix A provides a mapping between the learning objectives and categories in Bloom's Taxonomy of Education Objectives.

2.1.1 Lecture

We begin the lecture by reviewing base-10, motivating how and why we group by 10s. This is done by holding up two closed fists and asking students to count how many fingers are held up. When they respond with zero, we write the digit 0 on the board. Next, one finger is held up, and the exercise continues until we get to 10 fingers. We note that nine objects can be represented by a single digit, but another column is needed if we have ten or more objects. Similarly, if we have ten groups of ten objects each, a third column must be added. We conclude by noting one explanation for why we count by 10s: Humans typically have 10 fingers!

Next, we introduce the binary number system. To continue the finger counting example, we ask students how many digits we might use, and how high we might be capable of counting if we were aliens with only two fingers, similar to [34]. While it may sound silly, the exercise motivates the concept well. As with base-10, we discuss the grouping principle and note when a new column must be added. Using an example similar to that shown in Figure 2.1, we illustrate how elements can be grouped by twos, as well as tens. As an exercise, we give the students several small numbers and ask them to use grouping to write each number in decimal and binary.

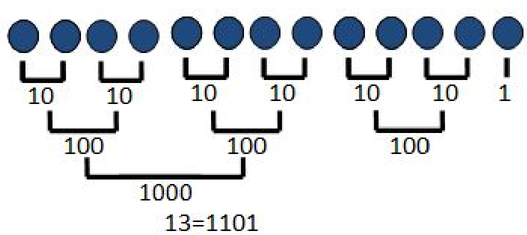


Figure 2.1: Grouping Example

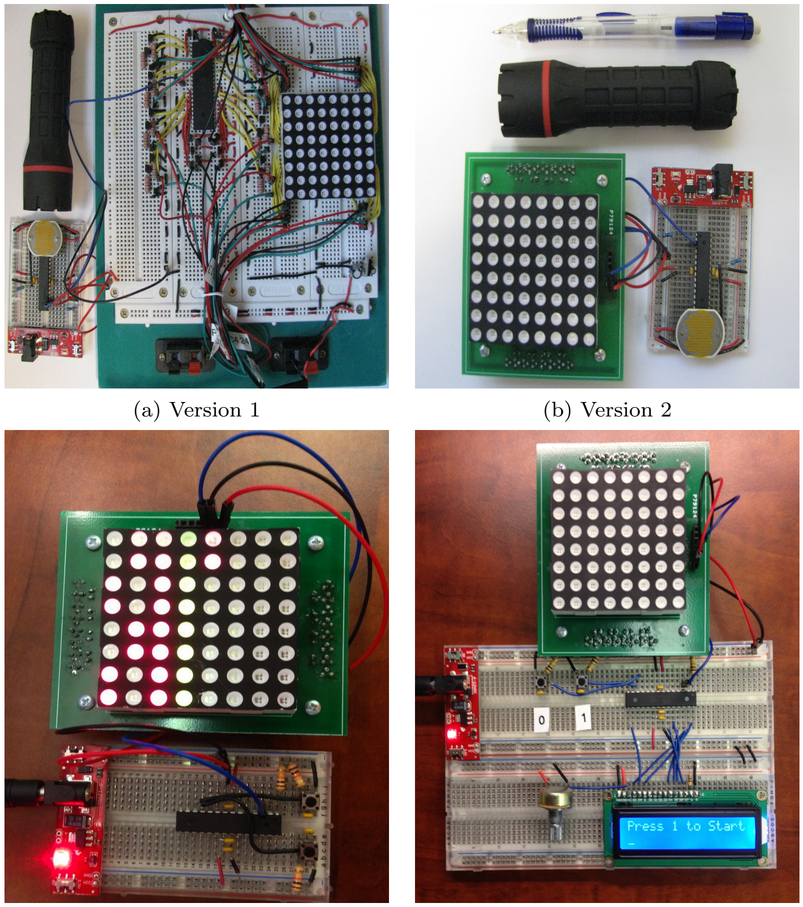
This exercise was designed to help students understand that 10 is effectively an arbitrary choice; any other base could be used. Next, we discuss how tedious this translation approach would be for large numbers, and then introduce a decimal-to-binary conversion approach based on repeated division by 2. While demonstrating this method, we explain how the quotient represents the number of groups that could be formed, and the remainder represents the remaining elements. We next demonstrate the reverse approach, converting a binary number to a decimal number. We start the discussion by noting the value of each digit in a decimal number. We explain that the value is the product of the digit and a power of 10; the power value depends on the digit's position. We then adapt this idea to binary numbers. When students are comfortable with the conversion process, we introduce binary addition and subtraction. To further engage students and provide opportunities for them to practice what they have learned, we give students several exercises that require them to convert decimal numbers to binary, and then to add or subtract the binary numbers. Next, we introduce a friendly competition. We divide the students into teams, provide two binary numbers, instruct each team to add or subtract the two numbers, and then convert the binary operands and resultant to decimal. Usually after several such exercises, some of the teams realize they can complete the problems in a more efficient manner by splitting the task among all team members.

We conclude with a discussion of why binary numbers are important in computer science, noting that the language used by computers is limited to 1s and 0s. We use the analogy of a light switch and explain that combinations of these switches allow computers to operate using large numbers.

2.1.2 Binary Toy

We begin the second class period with a review, and then introduce the serious toy, noting that the toy is just a simple computer, *i.e.*, it receives input, processes the input, and displays the results. The students are then divided into smaller groups, and the toy is demonstrated in each group. Students are assigned several binary conversion, addition, and subtraction problems and are instructed to perform the operations on paper before validating their work using the toy.

Architecture. The toy consists of an input device, two processors, and an output device. The development of the binary toy has evolved over four versions, all shown in Figure 2.2. The processors for each version consist of an ATMega168 [4] and an ATMega8515 [5]. The ATMega168 detects and receives data from the input device. Using serial communication, the ATMega168 transmits the data to the ATMega8515, which is responsible for driving the output device. The output device for each version consists of an 8x8 LED display. The fourth version, shown in Figure 2.2d, also includes an LCD screen that provides the user step-by-step instructions, as well as feedback on the correctness of their answers. The input device for the first and second versions, shown in Figures 2.2a and 2.2b, is a photoresistor wired to a microcontroller, triggered by a flashlight. When the flashlight beam is directed toward the photoresistor, a change in resistance is detected and read by the ATMega168



(a) Version 1 (b) Version 2

(c) Version 3 (d) Version 4

Figure 2.2: Evolution of the Binary Toy

microcontroller. The duration of the resistance change is recorded and compared with a specified threshold. If the duration exceeds the threshold, a 1 is recorded; otherwise, a 0 is recorded. Due to differences in lighting conditions from one classroom to another, the photoresistor proved to be unreliable, requiring a change in the input device for the third and fourth versions of the toy. The photoresistor was replaced by two tactile buttons, shown in Figures 2.2c and 2.2d; one representing a 0, and the other representing a 1.

Six of the eight rows on the LED display are used, as shown in Figure 2.3. The first two rows are used to represent operands. The third is used to display the predicted addition or subtraction result. The fourth represents the correct answer to the problem. The sixth is used to display the operation being performed. Finally, the eighth row displays the last bit position entered.

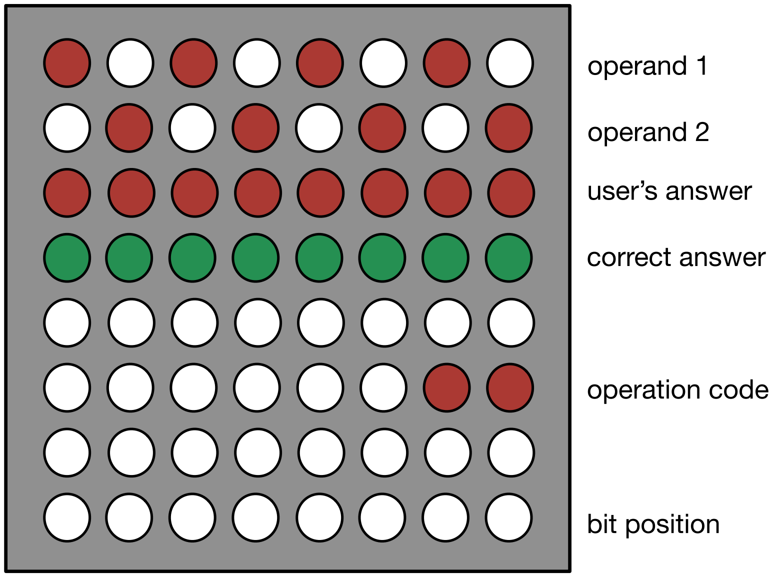


Figure 2.3: Display Setup

Toy Operation. The final version of the toy operates as follows. When power is supplied, the LCD screen prompts the student to "Press 1 to start". When the tactile button representing 1 is pressed, the display driver turns on the first row of red LEDs to indicate that it is ready to receive the first operand. The LCD screen also displays the message, "Enter the first operand, most significant bit first." The output display denotes a 1 by lighting a red LED; an unlit LED denotes a 0. To assist the student in keeping track of the bits entered, the eighth row displays a red LED for each bit entered. After all eight bits are entered, the LCD screen prompts the student to verify her entry, "Select 1 to continue or 0 to re-enter". If satisfied, a 1 is entered; otherwise, the board will clear the row and allow the student to re-enter the number. This process is repeated for the second operand. After accepting the second operand, the red lights on the sixth row turn on, indicating the need to enter the operator. The LCD screen prompts the student to enter "11" for addition, or "10" for subtraction. The student must also confirm her choice. At this point, the third row of LEDs is lit, indicating that the toy is ready to receive the predicted answer. The LCD screen prompts the student to enter the predicted result beginning from the least significant bit. Upon confirmation, the fourth row displays the correct answer in green lights. If the correct answer was entered, the LCD screen displays "Good Job! Press 1 to begin again." If the answer was incorrect, the LCD will display "Wrong Answer! Press 1 to begin again."

2.2 Pilot Groups

We piloted this approach with four groups of high school students. Three of the groups were participants in Emerging Scholars, a Clemson University summer program. The fourth group consisted of students from a local high school.

2.2.1 Emerging Scholars

The Emerging Scholars program was established in 2002 with a mission to reach out to high schools located in areas which, according to the US Census Bureau, have a high poverty rate [21,61]. Student participants are chosen because they exhibit the potential to succeed in higher education but lack the economic and social support needed to make attending college a reality. Participants in this program are provided summer experiences for three consecutive summers. During the summers, students are taught the importance of basic skills in reading, writing, math, and science. During the students' third summer in the program, they follow a class schedule that mimics that of a freshman. This affords students an opportunity to experience, in a small way, what it is like to be a college student.

2.2.2 Local High School

The participating high school students were chosen from a mathematics course, Statistical Analysis, classified as College Preparatory (CP). The class was chosen because, based on confidential feedback from the school counselor, the students' expected abilities appeared similar to those of the Emerging Scholars groups. As with the Emerging Scholars groups, these students had no predetermined interest in computer science.

2.3 Pilot Studies

In this section, we discuss three studies conducted using various versions of the binary toy.

2.3.1 Approach 1

During the summer of 2011, the pilot engaged two groups of junior and senior Emerging Scholars. Each group was divided into two sections. Each senior section, referred to as ES11Sa and ES11Sb, had 12 participating students. The junior sections, referred to as ES11Ja and ES11Jb, had 15 and 17 participants, respectively.

```
1. Computer science seems like it would be fun.
2. I might be interested in majoring in computer science in
   college.
3. I think I understand the need for alternate numbering systems.
4. I think I understand why humans use the decimal number system.
5. I think I understand the value of binary numbers in computer
   science.
6. I think I understand the concept of the binary numbering system.
7. I think I understand the relationship between decimal and binary
   numbers.
8. I think I can understand a number system with any base.
9. I think I could write the number 200 using only 0s and 1s.
10. I think I understand why 00000101 added to 00000010 = 7.
11. I think I understand why 00000010 subtracted from 00000101 = 3.
12. I think the teacher did an appropriate job explaining the
    material.
13. I like the format of this outreach program.
14. I would like to attend more outreach programs related to
    computer science.
15. I liked learning about binary numbers.
```

Listing 2.2: Survey Statements

Lecture Only. We met with the ES11Sa and ES11Sb for one day. They were taught the binary number system using only a lecture. This group was not introduced to the binary toy. Pre- and post-surveys were administered.

Lecture and Version 1 of the Binary Toy. We met with the ES11Ja and ES11Jb for two days. On the first day, we presented the lecture on the binary number system using the same lecture materials and style used with the ES11Sa and ES11Sb groups. Pre-surveys were administered before the lecture. On the second day, we reviewed the lecture material and demonstrated a prototype of the binary toy, shown in Figure 2.2a. We also discussed the development process used to construct the toy. Only two prototypes were available, so we divided the students into two smaller groups (7-8 students per group) and let them take turns practicing addition and subtraction using the prototypes. At the end of this meeting, students were given the post-survey.

Evaluation. The pre- and post-surveys consisted of 15 Likert-style statements, shown in Listing 2.2. The students were instructed to rate their level of agreement with each statement, choosing from *strongly disagree, disagree, moderately disagree, moderately agree, agree, and strongly agree*, weighted from 1 (strongly disagree) to 6 (strongly agree).

A statistical significance analysis was performed. Statements 12-15 were not included in the analysis because these statements could not be answered until after the presentation of the material. For the remaining 11 statements, a two sample F-test for variance was performed. Once the variance was determined, the appropriate two sample t-test was performed to determine if the pre/post difference was statistically significant (5% p-level). Table 2.2 summarizes the results of the statistical analysis. The first column represents the survey statements presented in Listing 2.2. The second, fourth, sixth, and eighth columns list the average scores for each statement of the pre-survey across all pilot groups. The third, fifth, seventh, and ninth columns list the average scores for each statement of the post-survey. The scores shown in bold in the third, fifth, seventh, and ninth columns represent statistically significant increases between the pre- and post-survey responses. Across the four pilot groups, 44 statistical analyses were performed, with 31 (70%) indicating a significant change in the mean response. We categorized the statements into three groups. Of the 15 total statements, two were related to student interest (statements 1-2), nine to student self-efficacy (statements 3-11), and four to student perception of the program (statements 12-15).

Survey Statement	**ES11Sa pre**	**ES11Sa post**	**ES11Sb pre**	**ES11Sb post**	**ES11Ja pre**	**ES11Ja post**	**ES11Jb pre**	**ES11Jb post**
S1	5.08	4.58	4.75	**3.5**	4.29	4.88	4.73	5
S2	3.67	3.33	2.67	2.17	2.47	2.94	2.93	3.67
S3	3.83	4.25	3.5	**4.67**	3.77	4.53	3.4	**4.53**
S4	3.75	**4.75**	3.58	4.33	4.11	4.52	3.8	4.53
S5	2.67	**4.67**	2.67	**4.42**	2.65	**4.94**	2.6	**4.73**
S6	2.33	**4.16**	2.6	**4.53**	2.53	**4.79**	2.6	**4.53**
S7	2.42	**4.08**	2.67	**4.75**	2.71	**4.18**	2.93	**4.67**
S8	3.8	4.31	2.67	**4**	3.06	**4.51**	3.27	**4.4**
S9	2.67	**4.33**	2.17	**4.75**	3.29	**4.65**	2.73	**5.07**
S10	2.25	**4.17**	2.08	**4.58**	2.06	**4.82**	2	**5.13**
S11	2.25	**4.17**	1.92	**4.5**	1.82	**4.88**	2.07	**4.87**
S12	—	4.83	—	5	—	5.47	—	5.2
S13	—	4.58	—	4.67	—	4.79	—	4.73
S14	—	3.83	—	3.33	—	3.32	—	4.33
S15	—	3.46	—	4.25	—	5.12	—	5.07

Table 2.2: Approach 1 – Binary Number System Pre-/Post-Survey Means

Interest in Computer Science. These statements were designed to measure the impact of the program on student interest in computer science. Although the change in mean for 7 of the 8 responses was not statistically significant (statement 1, ES11Sb, significant decrease), we do observe interesting results. The ES11Sa and ES11Sb groups both exhibited a decrease in interest, whereas the ES11Ja and ES11Jb groups had an increase in interest. As discussed above, the ES11Sa and ES11Sb groups were not introduced to the binary toy; the ES11Ja and ES11Jb groups were. The results suggest that the toy may have had a positive impact on students' interest in computer science as a discipline.

Content Understanding. These statements were designed to measure student understanding of the material presented, in particular the impact of the toy. The statistical analysis showed that 30 of the 36 t-test analyses represent significant changes between the pre- and post-surveys. It is also notable that *all* of the results for this category indicate an increase in (perceived) content understanding. We focus on statements 9-11, which relate to student understanding of how to convert, add, and subtract binary numbers. Knowledge of each of these concepts is reinforced through the use of the binary toy. In each case, the analysis indicates a statistically significant increase between the pre- and post-surveys. In addition, the ES11Ja and ES11Jb groups show a higher post-score on each of these three statements than the ES11Sa and ES11Sb groups. Since the ES11Sa and ES11Sb groups were not introduced to the toy, we believe this indicates that the binary toy had a positive impact on student understanding.

Structure of Outreach. These statements were designed to gauge whether students enjoyed the format of the program. Pre-survey data was not considered because students were unable to rate these statements until after the program was completed. As shown in Table 2.2, both ES groups, on average, agreed or moderately agreed that the instructors did an appropriate job. Also, on average, both ES groups enjoyed the format of the outreach module. However, the ES11Jb group was the only group that indicated they would like to participate in additional computer science outreach programs. This was not surprising since the ES11Jb group showed the highest interest in majoring in computer science. Lastly, three of the four ES groups indicated they enjoyed learning about binary. It is interesting to note that the ES11Ja and ES11Jb groups averaged a score of 5.1 for this statement, whereas the ES11Sa and ES11Sb groups scored 3.5 and 4.3, respectively. This suggests the binary toy had a positive impact.

2.3.2 Approach 2

The next pilot study engaged a group of students from a local high school. There were 34 participants, comprising 6 juniors and 28 seniors. This group is referred to as HS2011.

Lecture and Version 2 of the Binary Toy. We met with the HS2011 group for two days. The format of the first day was identical to the previous offerings. On the second day, we reviewed the material from the first day and introduced students to version 2 of the binary toy, shown in Figure 2.2b. Since this group was twice the size of the groups described in the previous section, we divided the students into two groups of 17. One group was introduced to the toy and asked to practice converting, adding, and subtracting binary numbers using the toy. While the first set of students were playing with the binary toy, the second set of students were learning about the development process for creating the toy. Students were given the opportunity to examine the breadboarded model of the toy and were instructed on how the more complete prototype was built. To allow students ample time to use the toy, the groups swapped learning areas after approximately 25 minutes. At the end of the period, a post-survey was administered.

Evaluation. The Likert-style survey discussed in the previous section was again used. Unfortunately, due to an unforeseen policy issue, pre-treatment survey data was unavailable for this pilot group. Hence, our analysis is based only on the post-treatment survey.

As before, in the evaluation of the final pilot, we group the survey statements into three categories measuring interest, content understanding, and module organization. For each statement group, we compute the average and standard deviation across the response data for all of the constituent statements. The results are summarized in Table 2.3:

Table 2.3: High School Survey Results

Statement Category	**Average**	**Std. Dev.**
Interest in CS (S1-2)	2.40	1.28
Content Understanding (S3-11)	3.71	1.66
Module Organization (S12-15)	3.29	1.37

Recall that a score of 3 denotes moderate disagreement, and a score of 4 denotes moderate agreement. Accordingly, the average scores in the content understanding category indicate that students completed the program with a generally positive impression of their content understanding. Unfortunately, they had a less positive view of the module's organization and their likelihood of pursuing a computer science degree. It is interesting to note that the high standard deviation values indicate significant variation in the response data. Indeed, an analysis of the individual statements reveals that approximately half resulted in bimodal response distributions, with frequency peaks on either side of 3. This suggests that the class was partitioned into two groups — those who "got it", and those who didn't.

We posit several potential explanations for the underwhelming response data. First, with regard to interest, this was the only pilot group that was not self-selected to participate in an outreach module. They elected to participate in a statistical analysis course, and were then required to participate in the binary arithmetic module. Their pre-existing interest in the displaced statistical analysis content may have biased their attitudes toward the outreach content. With regard to module organization, these results are not surprising. The classroom setup made it difficult to power all of the toys in a manner that supported small group participation. The devices were arranged on a central table, and students took turns participating in a large group. Our impression was that only half of the students interacted with the toys, which aligns with the bimodal response data noted above. Finally, it is impossible to tell whether these figures represent improvements over students' baseline impressions given the absence of pre-treatment data.

2.3.3 Approach 3

The next pilot study engaged two groups of Emerging Scholar juniors during the summer of 2012. The group was divided into two sections. The sections, referred to as ES12Ja and ES12Jb, had 16 and 12 participants, respectively.

Lecture and Version 3 of the Binary Toy. We met with ES12Ja and ES12Jb for two days. Pre-surveys were administered before the first meeting. The format of the remainder of the first day was identical to the previous offerings. On the second day, we reviewed the lecture material, specifically binary addition and subtraction, and introduced version 3 of the binary toy, as shown in Figure 2.2c. We also discussed the toy's development process, using the previous two versions as reference. In groups of 2 or 3, students were asked to practice converting, adding, and subtracting binary numbers, first on paper, then using the toy to verify their answers. After several practice problems, students began using only paper for the conversion process.

Evaluation. In addition to the Likert-style pre- and post-survey questions administered in the two previous pilot studies, we administered a pre- and post-quiz, shown in Listing 2.3, to measure content knowledge gained from the module. The statistical analysis performed on students' pre- and post-survey responses, as well as the pre- and post-quiz scores was identical to the analysis described in section 2.3.1. Table 2.4 summarizes the results of the analysis. The first column of the

```
1. Decimal number 34 can be written as ______ in binary.
   a) 11001   b) 10101   c) 100010   d) 11011
2. There is/are only ______ digit(s) in the binary number system.
   a) 1   b) 2   c) 10   d) infinite
3. 1001 in binary is the same as ______ in decimal.
   a) 7   b) 8   c) 10   d) 9
4. Perform the following binary addition.
     1100
   + 0110
    ---------
   a) 10100   b) 10010   c) 10000   d) 0100
5. Perform the following binary subtraction.
     1101
   - 1001
    ---------
   a) 1000   b) 0100  c) 1010  d) 0001
6. Binary numbers are important in Computer Science.  Why?
   a) They are easier to learn.
   b) They are best suited to represent the ON and OFF
      states of electronic switches.
   c) Decimal numbers are not that popular among computer
      scientists and engineers.
```

Listing 2.3: Binary Quiz

table represents the survey statements. The second and fourth columns list the average scores for each statement of the pre-survey and quiz. The third and fifth columns list the average scores for the post-survey and quiz. The average scores shown in bold in the third and fifth columns represent a statistically significant increase from the pre- and post-survey and quiz responses. A statistical paired t-test with a p-level less than 5% indicates a significant increase. As in previous offerings, we categorized the statements into three groups. Of the 15 total statements, two were related to student interest (statements 1-2), nine to student understanding (statements 3-11), and four to students' perception of the program (statements 12-15).

Interest in Computer Science. Statements 1 and 2 measure student attitudes toward CS. Responses for these questions indicate that there was not a statistically significant increase in attitudes toward the discipline. However, three out of the four responses increased, and both ES12Ja and ES12Jb indicated that CS seemed to be fun. Overall, the analysis indicated students showed a positive attitudinal shift toward CS.

Survey Statement	**ES12A pre**	**ES12A post**	**ES12B pre**	**ES12B post**
S1	4.06	4.25	3.83	4.08
S2	2.81	3.43	2.83	2.58
S3	2.69	**4.31**	2.42	**3.58**
S4	2.65	**4.06**	2.5	3.5
S5	1.88	**4.69**	1.58	**3.92**
S6	1.88	**4.5**	1.33	**4.08**
S7	1.94	**4.5**	1.67	**3.92**
S8	2.81	3.88	1.58	**3.92**
S9	2.44	**4.69**	1.67	**3.83**
S10	2.06	**4.47**	1.42	**4**
S11	2.06	**4.47**	1.42	**4**
S12	–	4.93	–	4.82
S13	–	4.73	–	4.58
S14	–	3.8	–	3.08
S15	–	4.73	–	4.08
Quiz Average	38	**71**	40	**61**

Table 2.4: Approach 3 - Binary Number System Pre-/Post-Survey Means

Content Understanding. Statements 3-11 assess students' self-efficacy. Overall, 89% of the post responses (16 out of 18, across groups) indicate a statistically significant increase in self-efficacy.

Structure of Outreach. The remaining statements, 12-15, assess whether students enjoyed this form of outreach. Pre-survey responses were not considered for these statements since they were only relevant after the presentation of the module. Overall, 75% of the responses (6 out of 8, across groups) averaged 4 (moderately agree) or higher, indicating that students enjoyed the outreach program.

Content Quiz. The quiz assesses content knowledge gained from the module. Not surprisingly, both ES12Ja and ES12Jb demonstrated a statistically significant increase in average test scores. This suggest student understood the material taught.

2.4 Conclusion

We began with the observation that the binary number system is central to a host of areas across computing. It is widely regarded as a fundamental topic, featured in a number of popular outreach programs. Unfortunately, there is evidence that existing approaches to teaching this topic inadequately engage and excite students. In response, we described a new approach to introducing binary arithmetic in the pre-collegiate curriculum using a supporting embedded platform that simultaneously engages visual and kinesthetic learners. The evaluation results are largely positive across the four pilot studies that have been conducted. Our hope is that this approach will serve as a model for introducing a fundamental topic that is often perceived as dull by young learners.

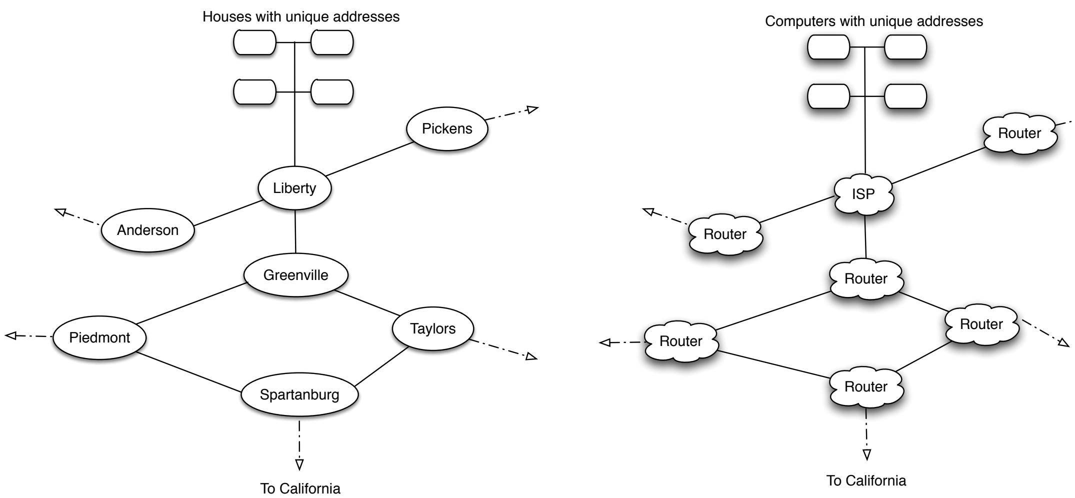
Chapter 3

Networks, Protocols, and Algorithms

Networks have been essential to society for thousands of years. The human body is a network of organs. The mail service, connecting people world-wide, has roots dating back to 2400 B.C., when the Egyptian Pharaohs deployed a network of couriers to deliver written communications [32]. Today, the mention of a network brings to mind computers, the Internet, and a range of social sharing services. Recognizing the importance of networks in our society, and more specifically, within the field of computing, we selected networks, protocols, and algorithms as the focus of our second module.

3.1 Networks, Protocols, and Algorithms Module

The curriculum module is designed to span two sessions of 60-minutes each. The first session consists of a lecture and a series of questions and discussion points designed to engage students. The second session consists of demonstrations and hands-on activities to reinforce the lecture concepts. For this module, the learning objectives call for students (i) to understand the definitions of network, protocol, and algorithm; (ii) to understand how networks, protocols, and algorithms are related; and (iii) to understand the relevance of networks, protocols, and algorithms in computer science. Appendix A provides a mapping between the learning objectives and the appropriate categories in Bloom's Taxonomy of Education Objectives.



(a) Postal Service

(b) Internet

Figure 3.1: Routing Illustrations

3.1.1 Lecture

We begin the lecture by defining a *network* as a group of entities connected and capable of communicating with each other, and then motivating what makes a network, *a network*. This is done by asking students to identify familiar networks. Students' responses range from radio networks to a pack of dogs. We next discuss the main components of several familiar networks: the postal service network, traditional computer networks, and the computer networks that constitute the Internet. Using an example students are familiar with —the postal service— we discuss the importance of assigning unique addresses in a standard format to each house, business, and school, to ensure reliable communication. Using the graphical illustration shown in Figure 3.1a, we discuss the process of delivering a letter from a home in South Carolina to a home in California. We point out that for some destinations, more than one path may be available, and then discuss why this is important. We discuss the factors that should be considered in choosing the best delivery path to use. Replacing the post offices with routers, we discuss the delivery of email using the same scenario, as shown in Figure 3.1b. Next, we define a *protocol* as a set of rules. We then steer the discussion toward an example scenario where the standard protocol is not followed. We ask students to consider the consequences if Alice, a mail carrier, decided she was tired of seeing the same houses, trees, and people each day. What if Alice wanted a change of scenery and decided, for one day, to deliver mail for her customers to the customers on Bob's route. We point out that this would result in chaos, confusion, and undelivered mail. As another example, suppose that the state of South Carolina decided to deviate from the mail addressing protocol used by the U.S. Postal Service. Diverging from the Postal Service addressing protocols would make it difficult for mail to be delivered from outside of South Carolina to destinations within South Carolina. We conclude by reiterating that it is essential for all entities in a network to abide by the same set of rules, or *protocols*.

Next we discuss the concept of an algorithm. First, we define an algorithm as a set of *step-by-step instructions to complete a task*, and then invite students to provide examples of common tasks they perform, where the steps might be considered an algorithm. The answers range from completing math problems to following a recipe to baking a cake. We then ask what happens when an algorithm is followed incorrectly, such as when completing a math problem. Using their responses, we explain that to complete a task, a correct algorithm must be followed, often using a specific ordering of steps. To illustrate when these steps must be completed in a specific order, we discuss the process a washing machine follows — add water, add soap, wash, rinse, and spin. We ask students to consider the outcome if a washing machine performed the steps in the following order: spin, add water, add soap, wash, rinse. To illustrate that task ordering is not always important, we use the example of adding milk and eggs to cake batter. We then discuss the relationship between networks, protocols, and algorithms and discuss why following standard protocols and algorithms is important within a network.

3.1.2 Activities

The second session begins with a demonstration of a sensor on our embedded toy. The architecture of the toy is described in Section 3.1.3. After the sensor demonstration, students are divided into groups and rotated through various additional activities. One activity introduces students to a bus-based hardware platform that allows sensing devices to be added to and removed from a network without interrupting communication among the devices. In another activity, students use the toy to create a network of devices that can communicate with each other. Finally, to demonstrate the importance of algorithms, student participate in an algorithm game.

Sensor Demonstration. The sensor demonstration begins with a description of the sensors and their capabilities. A desktop program provides a graphical view of real-time readings from the sensors.

A galvanic skin response (GSR) sensor was selected to allow students to actively participate in the demonstration by volunteering to come forward and place their fingers on the contact points (described in Section 3.1.3). We explain that one contact applies voltage to a finger, and the other contact measures the voltage on another finger. We explain that an emotional stimulus, such as fright, pain, nervousness, happiness, or anger, will stimulate the sympathetic nervous system, causing the skin to produce a tiny amount of sweat that will change the amount of natural resistance across the skin, reflected in a change in the graph being displayed. When the student volunteer has placed her two fingers on the contacts, she is instructed to stay calm for a moment to allow the graphical readings to level out. Next, she is instructed to pinch her ear, causing a change in the graph due to the physiological change caused by (very mild) pain. We then allow the class to ask the volunteer (reasonable) questions, or to make comments that might cause an emotional response from the volunteer.

This demonstration exposes students to the sensors on the network toy and the concept of measuring GSR. After a brief question and answer period about the demonstration, students are placed in groups and directed to the next activity.

Bus Network. The bus network activity begins with a demonstration of a microcontroller network composed of our toys, communicating through a (serial) bus. One of the devices is connected to a laptop computer. This device is responsible for identifying all of the devices in the network and collecting the data they send. The laptop computer uses a Java program to display the data.

We explain to the students that these devices are communicating through a serial bus, exposing them to the concepts of serial and parallel communication. We also discuss the associated communication protocol. We explain further that the serial bus is like a one-lane road, allowing traffic in both directions. We then ask how more than one car can use such a road. Using students' responses, we discuss the concept of a multi-drop bus and contention issues for bus access in a serial network.

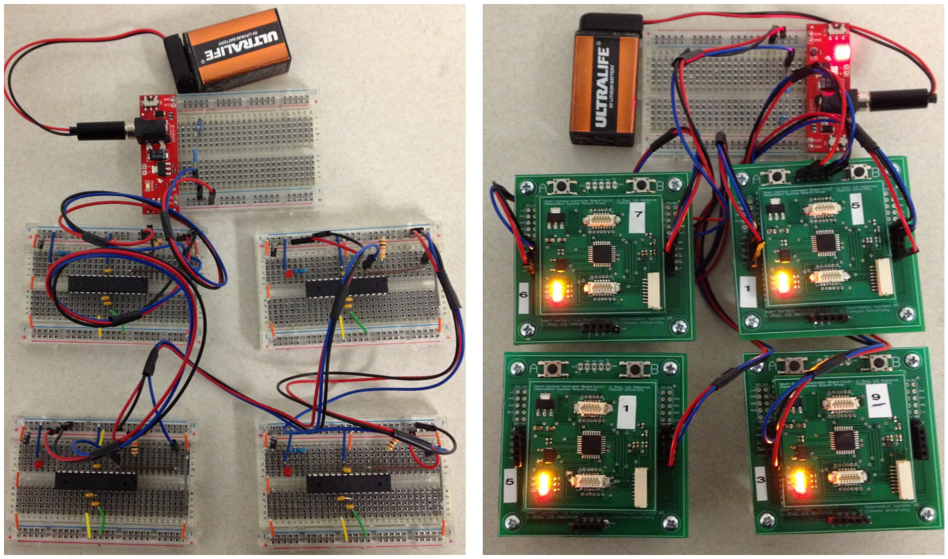
Next, we point out that each device connected to the network has a unique ID, which is displayed on the laptop screen. The ID of each device appears and disappears from the screen as devices are added to or removed from the network. As a demonstration of the sensors on our toy, students are shown readings on the screen characterizing normal noise levels in the room. Then, two student volunteers are selected to speak loudly or softly near a given sensor. All other students then observe the changing noise level on the computer screen.

We next remove all of the devices from the network, except for the one communicating with the computer. We then add one device at a time, each time asking students to identify the new ID on the computer screen, and to remember it. Next, we play a game, removing one device at a time, asking which device will disappear from the screen. This exercise builds the foundation for explaining how devices connected to a network learn about the arrival and departure of other devices. After a brief question and answer period, students are guided to the next activity.

Creating a Network. The network activity begins with a discussion on the process of creating the network hardware. We discuss how the network toy started as an idea, was prototyped using a breadboard, and then is transferred to a PCB, programmed, and tested. A quick demonstration is provided using a set of breadboarded devices connected and pre-programmed, as shown in Figure 3.2a. During the demonstration, basic definitions are revisited.

Next, each group is given a set of network devices pre-programmed with a *unique* peer-to-peer protocol. The differing protocols allow for a discussion of the communication problems created if one or more of the devices are interchanged between networks with different protocols. Students in each group are instructed on how to connect the devices using jumper wires, and are then tasked with connecting the devices to form a network. Once they are satisfied that all of the devices are connected properly, the network is powered up, as shown in Figure 3.2b. Each device is programmed to turn on or blink its LEDs, depending on the state of the device. Using the LED pattern, an explanation of the algorithm running on the network is provided. Students are given the chance to swap individual devices between networks, demonstrating communication problems that arise due to differing protocols. Throughout this activity, students are posed with questions designed to reinforce the concepts being taught. Just as important, students are given the opportunity to ask questions concerning the devices and the concepts covered.

Algorithms Game. We begin with a review of the importance of algorithms in computer science. We remind students that computers,“robots” in particular, complete tasks exactly as they are in-



(a) Breadboarded Network (b) Toy Network

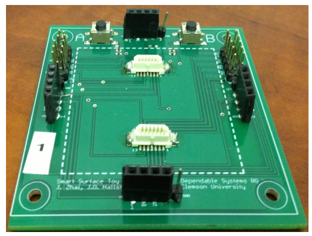
Figure 3.2: Creating a Network

structed. We explain to the students that they must navigate a robot through a maze using only three commands — "functions". We ask students to help define the three commands — step, turn left, turn right. Next, we asks a student volunteer to be blind-folded, and to serve as the robot. Once the student is blind-folded the remaining students create a maze using chairs and/or tables. When the maze is complete, the students take turns navigating the volunteer through the maze. As an incentive for students not to lead the volunteer into other objects, we offer a piece of candy to each student if the robot goes through the entire maze without touching any objects.

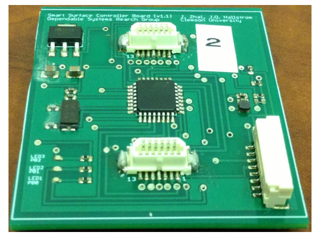
This activity provides an opportunity to discuss programming concepts, such as conditionals and loops. As an example, for the first few minutes of the activity we allow students to command the robot to step, step, step, etc. We then ask the students if they are tired of giving the individual step command over and over. We ask them to consider an easier and faster way of giving this command. Inevitably, they will suggest telling the robot to step multiple times. We then explain the concept of for loops to the students.

The activities provided in our module appeal to students of all learning styles. Students who are kinesthetic learners are able to create a network by connecting the devices with jumper wires. They learn about GSR through a sensor created using two pennies. Visual learners are able to visualize the change in sensor readings through a graphical user interface and are able to visualize the network communication through the pattern of blinking LEDs. Finally, students who prefer auditory learning are able to learn through the discussions, as well as the numerous question and answer segments throughout the activities.

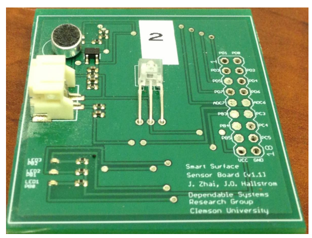
3.1.3 Network Toy Architecture



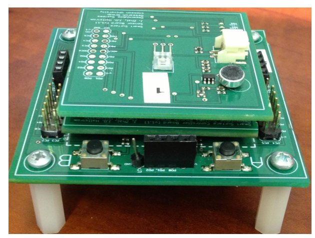
(a) Power Board



(b) Control Board



(c) Sensor Board



(d) Assembled Network Toy

Figure 3.3: Network Toy

Inspired by the MoteStack stackable sensing platform [29], we developed a second *"serious toys"*, a hardware platform consisting of three printed circuit board (PCB) layers, designed to engage students in hands-on networking activities. Each PCB layer is connected using two sets of Hirose connectors [41]. The assembled toy is shown in Figure 3.3d; Figures 3.3a, 3.3b, and 3.3c show the individual layers of the toy.

Power (bottom) Layer. As shown in Figure 3.3a, the bottom layer of the device is designed to enable device interconnection, exposing power, ground, and communication pins using male and female headers placed on each side of the board. To prevent accidental reverse polarization, the ground pin is connected to a male header, while the power pin is connected to a female header. To make the device extendable for future toys, this layer also has two sets of headers that expose the unused microcontroller pins, as well as two tactile switches connected to input pins on the microcontroller.

Control (middle) Layer. As shown in Figure 3.3b, the middle layer provides power regulation and computation support; it is the core of the device. It consists of a 5V regulator, an ATmega168 microcontroller [4], three LEDs —red, green, and yellow— as well as a header used for programming. (We omit supporting circuitry in our discussion).

Sensor (top) Layer. As shown in Figure 3.3c, the final layer provides three sensors used during hands-on demonstrations. Each sensor is connected to an analog-to-digital converter (ADC) pin on the microcontroller. One is a simple GSR sensor that consists of two pennies soldered to two wires. One wire is connected to the 5V power supply, while the second is connected to an ADC pin, grounded through a 10K resistor. A .1uf capacitor provides basic filtering. The second sensor is a broad spectrum photosensor, and the third is an electret condenser microphone. This layer also provides three LEDs — red, green, and yellow.

3.2 Pilot Groups

We piloted this module with six classes at a local middle school, and four groups from Emerging Scholars.

3.2.1 Local Middle School

The participating local middle school is the largest in the state, with approximately 1,350 students. During the 2012 school year, the middle school achieved an overall rank of "Good", indicating that its performance exceeds the standards for the state's "2020 Performance Vision". The vision states that by "2020, all students will graduate with the knowledge and skills necessary to compete successfully in the global economy, participate in a democratic society, and contribute positively as members of families and communities [64]." The pilot group consisted of two classes each from the sixth, seventh, and eighth grades. The pilot group was chosen from a STEM exploratory course. The two sixth grade pilot groups were participants in an "Intro to Careers" study, where students explore careers in STEM fields. The remaining seventh and eighth grade pilot groups were participants in the "Gateway to Technology" program, which offers an introduction to various topics in engineering, robotics, 3D modeling, and other topics.

3.2.2 Emerging Scholars

During the summer of 2013, the Emerging Scholars group, as described in Section 2.2.1, consisted of two groups of juniors, and two groups of seniors.

3.3 Pilot Studies

In this section, we discuss two studies conducted using the networking toy.

3.3.1 Local Middle School

During the winter of 2012, this pilot study engaged two classes each from the sixth, seventh, and eighth grades. For the remainder of this chapter, the individual classes are referred to by their grade and class section. As an example, 6A refers to sixth grade, section A. As shown in Table 3.1, these classes had a total of 160 students enrolled, with 136 participating in the first day of the program, and 118 completing both days of the program. Note that there were additional students who participated on the second day. Participants were only included in the analysis if they completed the evaluation for both days of the module. Table 3.1 also shows the number of students in each class identified by the school as needing special learning considerations. The evaluation process was anonymous; these students received no additional help completing their activities or evaluation instruments.

No. of Students	6A	6B	7A	7B	8A	8B	Total
Enrolled	24	20	29	25	30	32	160
Participants (day1)	21	18	23	22	26	26	136
Participants (day2)	15	17	23	20	23	20	118
Special Consideration	0	3	4	4	5	3	19

Table 3.1: Pilot Group Participation Information

Lecture and Network Toy. Recall that the module was designed to span two 60-minute class periods. The class period for the middle school was 40 minutes; therefore, we met with this group for four days. On the first day, we administered the pre-survey and quiz. On the second day, we presented the lecture on networks, protocols, and algorithms. On the third day, we introduced the network toy and other activities, as described in Section 3.1.2. We began the fourth day with a review of the lecture and administered the post-survey and quiz.

Evaluation. The survey consisted of 13 Likert-style statements, shown in Listing 3.1. For each statement, students were asked to rate their level of agreement by choosing from *strongly disagree, disagree, moderately disagree, moderately agree, agree, and strongly agree.* The statements were weighted from 1-6, with 1 being *strongly disagree*, and 6 being *strongly agree.* The statements were

```
1. Computer science seems like it would be fun.
2. I might be interested in majoring in computer science in
   college.
3. I think I know the definition of a network.
4. I think I understand the concept of a network.
5. I think I understand the value of networks in computer science.
6. I think I understand the relationship between networks and the
   Internet.
7. I think I know the definition of a protocol.
8. I think I understand the relationship between networks and
   protocols.
9. I think I understand the definition of an algorithm.
10. I think the teacher did an appropriate job explaining the
    material.
11. I like the format of this outreach program.
12. I would like to attend more outreach programs related to
    computer science.
13. I liked learning about networks, protocols, and algorithms.
```

Listing 3.1: Survey Statements

designed to evaluate students' level of interest in computer science, their perceived level of content understanding, and their perception of the outreach program.

With the exception of statements 10-13, a statistical analysis was completed to determine whether there was a significant change in the pre and post responses. Students were unable to rate statements 10-13 when completing the pre-survey; therefore only the post-survey data was considered. A two sample F-test was performed to determine if the variance was equal. Depending on the variance determination, the appropriate t-test was performed to determine if the changes between the pre-and-post data sets were significant (p-level was 5%). Table 3.2 summarizes the results of the statistical analysis. The first column represents the survey statements presented in Listing 3.1. The second, fourth, sixth, eighth, tenth, and twelfth columns list the average scores for each statement of the pre-survey across all pilots. The third, fifth, seventh, ninth, eleventh, and thirteenth columns list the average scores for each statement of the post-survey. The post-survey scores shown in bold denote statistically significant changes in the mean response. Of the 60 statistical analyses performed across all pilots 44 (73%) of the analyses indicated a statistically significant change in mean response. We again categorize the statements into three groups. Of the 13 statements, two were related to student interest (statements 1-2), seven to student understanding (statements 3-9), and four to student perception of the program (statements 10-13).

Survey Statement	**6A pre**	**6A post**	**6B pre**	**6B post**	**7A pre**	**7A post**	**7B pre**	**7B post**	**8A pre**	**8A post**	**8B pre**	**8B post**
S1	4.53	4.8	4.53	5	4.22	4.48	4	4.2	3.48	3.61	3.58	3.47
S2	3.2	**4.27**	3.56	**4.75**	2.87	3.26	3.35	3.35	2.74	3.22	3.1	2.95
S3	3.47	**4.93**	3.82	**4.82**	3.48	**4.61**	3.5	**4.6**	3.48	**4.44**	3.42	4.21
S4	3.13	**4.6**	3.77	**4.77**	3.44	**4.57**	3.15	**4.65**	3.30	**4.30**	3.53	4
S5	3.07	**4.53**	3.59	**5.06**	3.39	**3.78**	3.15	**4.15**	3	**4.17**	3.79	**4.74**
S6	3.6	**4.6**	3.94	**5**	3.91	4.13	3.6	4.1	3.44	**4.30**	3.68	**4.26**
S7	2.2	**4.4**	3.18	**5.35**	2.52	**4.13**	2.8	**4.15**	3	**4.04**	3.11	**4.53**
S8	2.13	**4.33**	2.63	**4.88**	2.48	**3.78**	2.45	**3.8**	2.57	**4.04**	2.63	**4.05**
S9	1.71	**3.93**	2.38	**4.81**	2	**3.76**	1.8	**3.93**	1.9	**3.55**	2	**4.61**
S10	—	5	—	5.29	—	4.71	—	4.82	—	4.4	—	4.79
S11	—	4.67	—	4.94	—	3.86	—	4.53	—	3.75	—	3.86
S12	—	3.93	—	4.71	—	3.36	—	3.65	—	3.21	—	3.64
S13	—	4.47	—	4.94	—	3.86	—	3.77	—	3.86	—	4
Quiz	66	72	54	**79**	56	**70**	48	**64**	56	**67**	53	**69**

Table 3.2: Local Middle School — Networks, Protocols, and Algorithms Pre/Post-survey Means

Interest in Computer Science. Statements 1 and 2 were designed to measure student attitudes toward computer science. Table 3.2 shows that the module had a positive impact on student attitudes for most groups. For group 8B, the average score for both statements decreased, though not significantly; statement 2 showed no change for group 7B. We believe group 8B's scores could have been influenced by the timing of the class period in which this group participated. The group participated during the last class of the day, and the post-evaluation documents were completed the last full day of classes before the start of the winter holiday. For statement 2, groups 6A and 6B both experienced a significant increase; however, the changes for these statements were not significant among the remaining groups. Overall, the analysis shows a positive attitudinal-shift with respect to interest in CS.

Content Understanding. Statements 3-9 were designed to evaluate students' self-efficacy. Overall, all of the post response scores increased, with 88% (37 out of 42, across groups) indicating a statistically significant increase in self-efficacy.

Structure of Outreach. Statements 10-13 were designed to measure whether students enjoyed the format of the outreach program. The survey data summarized in Table 3.2 indicates that all six groups felt the instructor did an appropriate job explaining the material. Three of the six groups moderately agreed that they liked the format of the program, with the remaining groups 7A, 8A, and 8B, showing average scores of 3.86, 3.75, and 3.86, respectively. Recall that a score of 3 represents moderate disagreement, and a score of 4 represents moderate agreement. However, further analysis of group 7A, 8A, and 8B's responses to statement 11 revealed that the median score was 4 for each group. The percent of students, per group, scoring 4 or above was 62%, 70%, and 71%, respectively, indicating that the majority of the students did, in fact, like the format of the program. Statement 12 gauged students' interest in attending other CS outreach programs. It was not surprising that groups 6A and 6B scored higher than the other groups given that they were the only two groups that showed an interest in majoring in CS in college. Finally, three of the six groups indicated that they liked learning about networks, protocols, and algorithms, with the remaining groups, 7A, 7B, and 8A, showing average scores of 3.86, 3.76, and 3.86, respectively. Again, further evaluation revealed groups 7A and 7B both had a median score of 4, and 8A's median was 3.5. The percentage of students for groups 7A, 7B, and 8A scoring 4 or above was 67%, 65%, and 50%, respectively. The above indicates the evaluation results are largely positive across all groups.

Content Quiz. The content quiz, shown in Listing 3.2, consisted of 10 multiple choice and true-false questions designed to gauge students' understanding of the material taught. Students were asked to complete the quiz at the beginning and end of the program. As shown in Table 3.2, all of the groups showed an increase in post-quiz scores. With the exception of group 6A, all increases were statistically significant. However, we noticed group 6A scored highest on the pre-quiz, possibly explaining why the increases were not significant. Overall, evaluation results for this pilot group suggest students understood the material presented.

3.3.2 Emerging Scholars

During the summer of 2013, this pilot study engaged two groups of junior and senior Emerging Scholars. Each group of juniors and seniors were divided into two sections. The senior sections, referred to as ES13Sa and ES13Sb, had participants totaling 14 and 11, respectively. The junior sections, referred to as ES13Ja and ES13Jb, had 17 participating students.

Lecture and Network Toy. We met with each section for two days, in 75-minute sessions. On the first day, we administered the pre-survey and quiz. We then presented the lecture on networks, protocols, and algorithms, as described in Section 3.1. We began the second day with a review of the material covered in the lecture. Next, we introduced the network toy and other activities. The post-survey and quiz were administered at the end of the class period.

```
1. A group of entities connected together using a communication
   channel is called a ______.
   a) Forest b) Post Office c) Network d) Station
2. A(n) ______ is a set of planned actions to complete a given
   task.
   a) Algorithm b) Program c) Job d) Instruction
3. A set of rules for exchanging messages between two entities
   is called a(n) ______.
   a) Algorithm b) Program c) Network d) Protocol
4. If an unexpected event happens, a computer is smart and can
   determine by itself what action it needs to take.
   a) True b) False
5. Each computer in a network can use a different protocol and
   still be able to communicate with all other computers on
   the network.
   a) True b) False
6. You must have a computer to form a network.
   a) True b) False
7. It is important for each device/computer in a network to have a
   unique identity.
   a) True b) False
8. There are no problems associated with allowing one computer
   in a network to control the actions of all other computers.
   a) True b) False
9. A cable TV network is an example of a centrally managed network.
   a) True b) False
10. The Internet is an example of a distributed computer system.
    a) True b) False
```

Listing 3.2: Quiz Questions

Evaluation. The evaluation instruments and statistical analysis were the same as the Local Middle School pilot described in Section 3.3.1.

Table 3.3 summarizes the results of the statistical analysis of students' pre- and post-survey and quiz scores. The first column of the table lists the survey statements. The average pre- scores for each statement and the quiz are listed in columns two, four, six, and eight. The post-survey and quiz averages are listed in columns three, five, seven, and nine. The post-survey and quiz averages shown in bold indicate a statistically significant increase based on a p-value less than 5%. Across the groups, 44 statistical analyses, were performed. Of the 44 analyses 23 (52%) indicated a significant change in mean response. We again categorized the statements into three groups. Of the 13 total statements, two were related to student interest (statements 1-2), seven to student understanding (statements 3-9), and four to student perception of the program (statements 10-13).

Survey Statement	ES13Sa pre	ES13Sa post	ES13Sb pre	ES13Sb post	ES13Ja pre	ES13Ja post	ES13Jb pre	ES13Jb post
S1	4.62	5.08	4.27	4.64	3.88	3.88	4.24	4.65
S2	3.69	4	3.27	3.36	3.88	3.35	2.41	**3.82**
S3	4.39	5	3.91	**5.27**	5.88	**5.06**	4.24	4.82
S4	4.08	4.77	3.55	**4.55**	3	**4.29**	4.18	**5**
S5	4.23	4.85	3.18	**4.73**	2.53	**4.47**	3.70	**4.65**
S6	4	**4.62**	4	**4.82**	3.7	**4.65**	3.14	**4.62**
S7	3.85	**5.08**	2.82	**5**	1.94	**5.12**	2.71	**4.94**
S8	3.85	**4.92**	2.36	**4.64**	1.59	**4.71**	2.71	**5**
S9	3.69	**4.85**	2.82	**4.82**	1.82	**4.82**	3.18	**4.71**
S10	–	4.85	–	5.34	–	5.19	–	5.29
S11	–	4.77	–	4.55	–	5.28	–	4.82
S12	–	5	–	4	–	3.8	–	3.941
S13	–	4.77	–	4.91	–	4.2	–	4.59
Quiz Averages	59	68	54	64	52	**73**	61	71

Table 3.3: Emerging Scholars – Networks, Protocols, and Algorithms Pre/Post-survey Means

Interest in Computer Science. Statements 1 and 2 assess students' attitudes toward CS as a discipline and career choice. As shown in Table 3.3, statements 1 and 2 indicate that the module had a mostly positive impact on student attitudes. Overall, 75% of respondent scores increased (6 out of 8 across groups), with one statistically significant increase; ES13Ja did not change. With the exception of ES13Ja, for statement 1, three out of four respondent scores were 4 (moderately agree) or higher. A closer review of ES13Ja reveals that the median score was 4, with 59% of respondent scores at 4 or higher. Three of the four responses for statement 2 indicate a positive attitudinal shift, with ES13Jb showing a statistically significant increase. ES13Ja showed a decrease, though not statistically significant. Again, a closer look at ES13Ja indicates that although there was a decrease in the average post score, the median was 4, with 59% of post scores at 4 or higher. The evaluation results suggest the program had a positive impact on student interests in CS.

Content Understanding. Statements 3-9 assess students' self-efficacy with respect to the material taught. All post responses showed an increase, with 85.7% indicating a statistically significant increase in self-efficacy (24 out of 28, across groups). All of the 28 post responses averaged 4 or higher, indicating the students were confident in their understanding of the material presented.

Structure of Outreach. Statements 10-13 assess students' overall level of program enjoyment. Pre-survey responses were not considered. Overall, 87.5% averaged a score of 4 or higher (16 out of 18, across groups). In response to statement 10, students indicated that they thought the teacher did an appropriate job, with scores ranging from 4.85 to 5.34. Responses to statement 11 showed that students liked the format of the outreach program, with scores ranging from 4.55 to 5.29. Statement 12 assessed whether students would like to attend more outreach in CS. Analysis results indicated that both ES13Sa and ES13Sb preferred to attend more CS outreach, with average scores of 5 and 4, respectively. Although average scores for ES13Ja and ES13Jb were 3.8 and 3.94, respectively, further analysis indicated the median score for both groups was 4, with 53% and 54%, respectively, of students in each group scoring 4 or higher. Average responses to statement 13 indicated students from all four of the Emerging Scholars pilot groups enjoyed learning about networks, protocols, and algorithms; averages ranged between 4.2 to 4.91. Overall, the analysis indicated the students in general enjoyed the outreach program.

Content Quiz. The quiz responses measured content knowledge gained from the module. Post-quiz scores increased 10 or more points, with the exception of ES13Sa, which had a 9 point increase. Although all pilot group quiz scores increased, only ES13Ja's average indicated a significant increase, at 22 points. The evaluation indicates the program had a positive impact on the students' content understanding.

3.4 Conclusion

Networks are pervasive in our lives. We use networks in our homes to connect our phones, our computers, and our entertainment devices. They enable communication by phone, by email, and by postal service. Most workplaces cannot operate without computer networks. Motivated by the importance of networks in our daily lives and the crucial role networks play in computing, we have developed a curriculum module designed to introduce pre-collegiate students to basic concepts of networks, protocols, and algorithms. Our module includes lecture, demonstration, and hands-on activities supported by an embedded *toy*. The curriculum is designed to engage students of all learning styles – visual, auditory, and kinesthetic. We piloted the program with six groups of middle school students, two each from sixth, seventh, and eighth grades, and four groups of high school students. The evaluation results are largely positive across all groups.

Chapter 4

Sensors and Sensor Networks

Sensors have become pervasive, used throughout our daily activities. Smartphones, for example, are equipped with multiple sensors, including i) a proximity sensor to determine the location of the phone in relation to the user's ear, ii) an accelerometer to determine the orientation of the screen, iii) a light sensor to support automatic adjustment of screen brightness, and iv) a gyroscope to improve the user's gaming experience. Virtually all appliances include one or more sensors. Networks of sensors are also common, covering home and medical monitoring [8, 43], wildlife behavioral monitoring [38], military applications [49], and environmental management [29]. Observing the value of sensors in computing and building on the Networks, Protocols, and Algorithms module, we selected the area of sensors and sensor networks for the topic of the third module.

4.1 Sensors and Sensor Networks Module

The third curriculum module spans two 60-minute sessions. The first begins with a lecture and ends with a hands-on activity. The second consists of additional hands-on activities and a sensor demonstration, both designed to reinforce the lecture concepts. We focus on four learning objectives. The objectives call for students (i) to have a basic understanding of how sensors work; (ii) to understand the construction of a basic sensor; (iii) to be familiar with various types of sensors; and (iv) to understand the relevance of sensors and sensor networks in computer science. Appendix A provides a mapping between the learning objectives and the appropriate categories in Bloom's Taxonomy of Education Objectives.

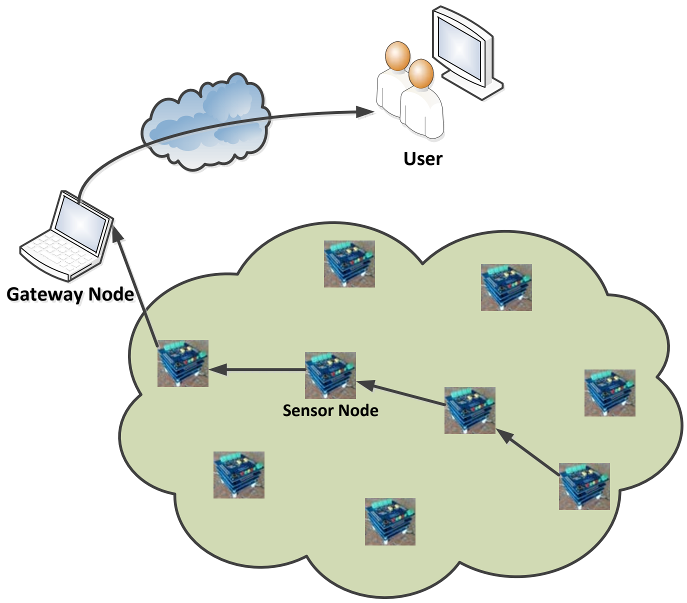


Figure 4.1: Wireless Sensor Network Example

4.1.1 Lecture

We begin by asking students to identify familiar sensors, and then discuss their responses. We ask students to consider simple sensors in the human body – skin, eyes, ears, nose. After a brief discussion related to "human sensors", we discuss what sensors are used for – namely, to measure a physical or environmental condition. Using a passive infrared (PIR) motion sensor as an example, we discuss the basics of how sensors work. We begin by asking students if they understand why the doors at the local grocery store open automatically when a person is near. Using student responses, we discuss in detail how PIR motion sensors work. Next, we provide a demonstration of a PIR motion sensor connected to a microcontroller. The microcontroller increments a counter displayed on an LCD screen, activates a piezo buzzer, and turns on an LED each time motion is detected. Next, using Figure 4.1, we discuss a network of wireless sensor nodes and the process of passing data from one sensor node to another, eventually reaching a main computer server. This discussion provides an opportunity to explain common issues in sensor networks, including routing and reliability. We explain the value of sensors to society by discussing several applications in agriculture, water conservation, and automotives. Finally, we conclude by assisting students in making a simple soil-moisture sensor, shown in Figure 4.2a, from plaster of paris, nails, and a straw, as adapted from [14].

4.1.2 Activities

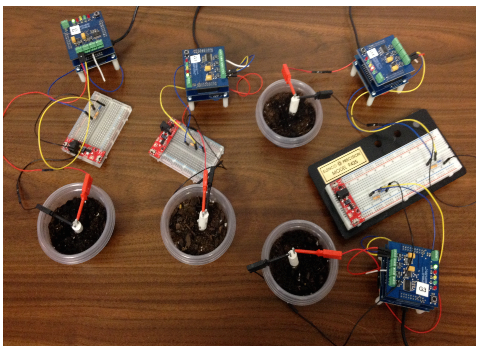
Day two includes two activities. The first involves testing the soil moisture sensors made during the previous session. The second includes assisting students in making a homemade pressure sensor, similar to [12].

Sensor Testing. We begin day two with a series of questions designed to help students understand how the homemade soil-moisture sensor detects moisture. Topics covered during this discussion include principles of electrical conductance, insulators, voltage, current, and resistance. We first test each of the homemade soil moisture sensors by connecting the probes of a multimeter to the nails in the sensor, inserting the sensor in a cup of dry soil, slowly adding water, and observing the change in resistance. The next step involves connecting each sensor to a MoteStack [29], a stackable sensing platform. As shown in Figure 4.2c, students insert the sensor probes in various degrees of moist soil. Next, they observe the readings using a web-based application, shown in Figure 4.2b, developed as part of the Intelligent River® program [67]. This exercise provides an opportunity to discuss i) how the MoteStacks collect data; ii) how data is transmitted to the data center; and iii) how data is retrieved and displayed on the website. Weather permitting, this exercise takes place outdoors; otherwise, containers of soil are provided for use in the classroom. This activity provides a natural opportunity to review the information taught in the Networks, Protocols, and Algorithms module.

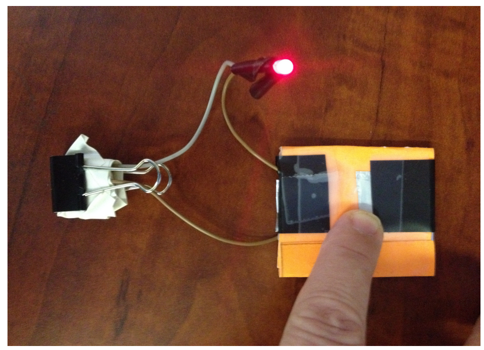
Making a Pressure Sensor. This activity was designed to reinforce student understanding of sensor construction. The sensor, pictured in Figure 4.2d, is made from poster board, household aluminum foil, tape, Velostat® by 3M (a resistive material [52]), insulated solid wire, an LED, and a 3.3V coin battery. To conserve time, individual components of the sensor are pre-cut and individually packaged. Supplied with a pressure sensor packet, each student is given a detailed explanation of the purpose of each item, and then guided through the construction of the pressure sensor. This activity provides an opportunity to discuss concepts of basic circuits, how sensors work, and their relevance to computer science.

(a) Homemade Moisture Sensor

(b) Example of Website Display



(c) Motestack Sensor Network



(d) Homemade Pressure Sensor

Figure 4.2: Sensor Module Activity Examples

4.2 Sensor Network Architecture

Figure 4.3 depicts the architecture of a prototypical sensor network system. Our sensor system is designed to collect, transmit, and display data. Sensors made by the students are connected to a set of MoteStacks, which read voltage changes corresponding to changes in environmental parameters. To transmit data, each MoteStack is equipped with an XBee wireless radio [44]. The collected data is transmitted to our data center using a laptop connected to an embedded device also equipped with an XBee wireless radio, a *gateway node*. The gateway node acts as an interface between the sensor network and the data center. The data center processes and saves the received data in a database. Sensor observations collected from different sensors are identified by unique IDs assigned to the sensors. A customized website is used to retrieve the saved data from the database and display the data in the form of graphs. Each group of students is able to track the data collected from their sensors by selecting their sensor ID on the website.

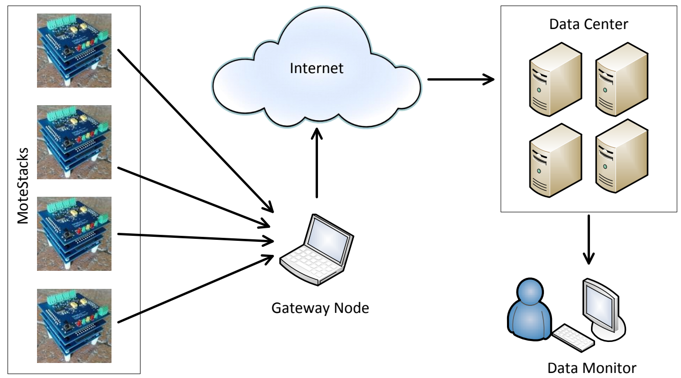


Figure 4.3: Sensor Network Architecture

4.3 Pilot Groups

We piloted this module with five classes from a local middle school, and four groups from Emerging Scholars.

4.3.1 Local Middle School

The local middle school is a charter middle school, a nonprofit public school system operating within the local public school district, with a mission to teach students "personal responsibility and a compassion for their community through single gender classes and innovative teacher, parent, and community collaborative learning [19]." The school provides challenging academic opportunities through core curriculum, including English, Math, Science, History, and Foreign Language. To increase self-esteem and help students realize the value of their potential contributions to society, students participate in various service opportunities within their communities. This school enjoys a large volunteer group comprising parent and community leaders. The pilot groups consisted of two groups of seventh grade boys, two groups of seventh grade girls, and one group of eighth grade boys. Each group was enrolled in a science course.

4.3.2 Emerging Scholars

The Emerging Scholars group consisted of two groups of juniors, and two groups of seniors. A description of the Emerging Scholars program is provided in 2.2.1.

4.4 Pilot Studies

In this section, we discuss two pilot studies conducted using the Sensors and Sensor Networks module. We begin by describing three modifications to the evaluation instruments and methods used in previous pilot studies. We then detail the evaluation of each of the two pilot studies.

4.4.1 Evaluation Modifications

During the analysis of the pilot studies discussed in Chapters 2 and 3, we identified three evaluation instrument improvements, which were incorporated prior to piloting the Sensors and Sensor Networks module. We briefly summarize each improvement.

First, previous pilot studies suggest the surveys were too long. The surveys for the Binary Number System module and the Networks, Protocols, and Algorithms module consist of 15 and 13 Likert-style statements, respectively. An analysis of the survey responses revealed that students often failed to complete the second page of the survey. In previous pilot studies, we attempted to correct this problem by reminding students to complete both pages of the survey document. However, student interest in completing the entire survey often appeared to wain, resulting in incomplete surveys. Thus, it was determined that we needed to condense the pre- and post-survey statements to one page [45]. We reduced the number of statements included on the pre- and post-survey documents to eight statements.

The previous evaluation process also lacked a mechanism to ensure student accountability. To assess content understanding during each pilot study, we administered a pre- and post-quiz. However, in accordance with our approved Research Compliance Review Application with Clemson University's Institutional Review Board (IRB), students were told that their participation in our program would not affect their course grade. We also told students that the pre- and post-quizzes would not be seen by their course instructor. As an unintended result, there was no incentive for students to participate in the learning process. From class observations, we found that most students found the program engaging, evidenced by their willingness to participate. Inevitably, however, there were several students that needed an incentive to participate. Therefore, we changed the current IRB application by including a statement allowing the class instructor to assign homework, and to test the students on the content taught in our pilot. To assist the course instructor, we provided sample homework assignments, as well as test questions.

Finally, we observed from previous pilot studies that the pilot groups often consist of students with various degrees of academic abilities. As shown in Section 3.3.1, at least 15% of participants in five out of the six pilot groups were identified by the school as needing special learning considerations. Further, participating pilot groups are typically chosen by school administrators, rather than being self-selected. It has been our observation that although the modules are engaging for most students, there are often a few students who will not participate. To determine if these participants are having a negative effect on the evaluation results, it is useful to identify the data collected from these students. To provide a means of identifying these students, we developed two changes, one to the evaluation instruments, and one to the collection process. First, we included questions on the pre-quiz involving basic math principles, targeting at least two grades below the pilot group. Inability or unwillingness to answer the questions assists in identifying students in the above categories. As a change to the collection process, we modified the current IRB documents to allow students to put their names on the evaluation instruments. We also asked the course instructor to provide a map of seating assignments. Using observations and a map of student seating assignments, we identified students who did not participate in the learning process, as well as students who appeared to have special learning needs. This allowed us to perform the statistical analysis with and without these students' data to examine the impact they are having on the data.

4.4.2 Evaluation Instruments

We used two evaluation instruments for this module, administered both pre- and post-treatment, consisting of a survey and a quiz. The survey consisted of 5 Likert-style statements, shown in Listing 4.1, and 3 demographic questions on age, race, and gender. The statements' Likert responses and weights are described in Section 2.3. Statements 1 and 2 are designed to evaluate student interest in computer science, and statements 3-5 measure students' self-efficacy with respect to content understanding. The quiz consists of short answer, true/false, and multiple choice questions, as shown in Listing 4.2, designed to assess students' content understanding.

4.4.3 Local Middle School

During the spring of 2014, this pilot study engaged student participants from five single-gender classes at a local middle school – three male and two female. For the remainder of this

```
1. I would like to attend more outreach programs related to computer
   science.
2. I might be interested in majoring in Computer Science in college.
3. I think I understand the importance of sensors and sensor networks
   in computer science.
4. I think I understand how to make a sensor from common household
   products.
5. I think I understand the principle function of a sensor.
```

Listing 4.1: Survey Statements

```
1. The principle function of a sensor is to ______ a physical or
   environmental condition.
   a) create b) ignore c) measure d) none of the above
2. When a sensor node in a wireless sensor network collects data
   it sends the data to each sensor node in the network.
   a) true b) false
3. An example of a sensor on the human body is a nose.
   a) true b) false
4. A ___________  consists of a group of sensors that monitor
   physical or environmental conditions and pass the data through
   a network to a main location.
   a) gateway network
   b) sensor network
   c) network of laptop computers
   d) none of the above
5. A sensor can be constructed using common household products.
   a) true b) false
6. Name two applications of a network of sensors.
```

Listing 4.2: Quiz Questions

chapter, the classes will be identified by their grade, section, and gender. As an example, 7AM refers to seventh grade, section A, all male.

Lecture and Sensor Toy. As discussed in Section 4.1, the module was designed to span two 60-minute class periods. The class period for the middle school was 50 minutes. We consequently met with the groups for three days each. On the first day, we administered the pre-survey and quiz, presented the lecture, and assisted students in making a homemade moisture sensor, as described in Section 4.1.1. On the second day, we presented the *Sensor Testing* and *Making a Pressure Sensor* activities described in Section 4.1.2. On the third day, we began with a review of various concepts presented during the lecture. Next, we discussed careers in CS, finishing with administering the post-survey and quiz.

```
1. Suppose you invited 12 friends to a pizza party.  Assuming
   each person will eat 3/8 of a pizza, what is the MINIMUM number
   of pizzas you will need to purchase.  Please show your work.
2. Add 3/10 + 4/100. Please show your work.
3. Multiply 147 X 32. Please show your work.
```

Listing 4.3: Local Middle School - Age Appropriate Math Questions

Evaluation. The evaluation instruments are described in Section 4.4.2. A statistical analysis was completed to determine whether there was a statistically significant change in response data for statements 2-5. Students were unable to rate statement 1 when completing the pre-survey; only the post-survey was considered. A two-sample F-test was performed to determine if the variance was equal. Based on the determination of the F-test for variance, the appropriate t-test was performed to determine if there was a significant change in mean between the pre and post data sets (p-level was 5%). The statistical analysis was performed on three subsets of the data. First, an analysis was performed on all of the student data, shown in Table 4.3, referred to as the *aggregate* table. Next, recall we included three basic math questions on the pre-quiz, shown in Listing 4.3 . Only data from students that answered at least two of the three math questions correctly were used in the analysis summarized in Table 4.4, referred to as the *high* table. Finally, we performed an analysis on data from students who answered at most one question correctly, shown in Table 4.5, referred to as the *low* table. For each of the tables, the first column denotes the survey statements, shown in Listing 4.1. The second, fourth, sixth, eighth, and tenth columns list the average scores for each statement of the pre-survey, across all pilots. The third, fifth, seventh, ninth, and eleventh columns list the average scores for each of the post-survey statements. The post-survey scores shown in bold denote a statistically significant change in the mean response. With respect to the aggregate data, Table 4.3 shows that of the 25 statistical analyse performed, 19 (76%) indicate statistically significant changes of mean. The 5 statements are categorized into two groups, two statements were related to interest in computer science (statements 1-2), and three statements to student self-efficacy (statements 3-5). In the following subsections, we discuss the aggregate survey analysis, and the pre- and post-quiz score analysis. Finally, we consider the impact students identified to have special learning considerations or students who were unwilling to participate in the learning process are having on the data.

Interest in Computer Science. Statement 1 was designed to gauge student interest in computer science and Statement 2 was designed to gauge student attitudes toward computer science as a career choice. Table 4.3 shows that overall, the program had a positive impact on student attitudes toward computer science. Students could not answer Statement 1 until completion of the program; only post-survey data was considered. Post-survey means for Statement 1 ranged from 3 to 4.2. Recall that a score of 3 represents moderate disagreement, and 4 represents moderate agreement. With the exception of 8AM, all groups had a median score of 4 or above. The analysis for Statement 2 showed an increase in post-survey means across all pilots ranging from 3.05 to 4.4, with 2 out of 5 (40%) indicating a statistically significant increase. With the exception of 7BF, all groups showed a median score of 4 or above.

Content Understanding. Statements 3-5 gauge student self-efficacy with respect to the content taught. Overall, 100% of post-survey means (across all pilots) showed a significant increase. The range for all statements was 3.9 to 5.

Content Quiz. The content quiz consisted of six multiple choice, true-false, and short answer questions designed to gauge students' understanding of the material taught. Students were asked to complete the quiz prior to the lecture, and again after the completion of the program. As shown in Table 4.3, all of the pilot groups showed an increase in post-quiz means, of which 2 (40%) were statistically significant increases. The post-quiz scores fell between 70% to 83%.

Impact of Low Performers. To determine the impact students with special learning considerations may have had on the aggregate evaluation results, we performed a secondary analysis focusing on data collected from students who correctly answered at least two of the three math questions included on the pre-quiz. The results are shown in Table 4.4. A review of this *high* table indicates that there was little difference from the results shown in Table 4.3, *aggregate*. This suggests that the impact of the *low* performers on the aggregate data was minimal. To confirm this assessment, we performed a statistical analysis on the data from students that did not answer at most one of the three math questions correctly on the pre-quiz. The results are shown in Table 4.5. Next, we took the difference between the *high* and *low* results, shown in Table 4.6; parentheses indicates a negative difference. Finally, we calculated the average change of post scores for each statement and quiz, across all pilot groups. The results are shown in Table 4.1. The average change was in the

Survey Statement	Average High-Low Change
S1	.03
S2	.15
S3	.04
S4	.02
S5	.13
Quiz	7

Table 4.1: Langston Middle School - Average Change of High/Low Difference (across all groups)

range of .02 to .15 on a six point scale. This suggests that students who answered fewer than two of the three questions correctly had a minimal impact on the evaluation results. In fact, this indicates that the teaching approach, relying on serious toys, is an effective approach for both high and low performers.

Table 4.6 shows that, with the exception of the quiz, all 8AM post differences were negative. Table 4.7 shows there were 4 students in the *low* group, and 12 in the *high* group. A review of the original post-survey statements for students in the *low* group reveals that one student scored strongly agree (6) on all of the survey statements, and two students scored agree (5) on statements 3-5. A review of original post-surveys for the 12 students in the *high* group did not reveal evidence of similar scoring patterns. We speculate that this is an indication that these students showed a lack of engagement when completing the post-survey. In conclusion, the analyses indicate the program is having a positive impact. Largely, the groups show a positive attitudinal shift in interest toward CS as a career. All groups show an increase in self-efficacy with respect to content understanding. The post-quiz and module-specific quizzes indicate that students understood the material taught.

4.4.4 Emerging Scholars

During the summer of 2014, this pilot study engaged student participants from four groups, two each of junior and senior Emerging Scholars. The junior sections will be referred to as ES14Ja and ES14Jb, and the seniors as ES14Sa and ES14Sb.

Lecture and Sensor Toy. Recall that the module was designed to span two 60-minute class periods. The Emerging Scholars' class periods were 75-minutes each. We consequently met with these groups over two-day periods. On the first day, we administered the pre-survey and quiz, presented the lecture, and assisted students in making a homemade moisture sensor, as described in Section 4.1.1. On the second day, we presented the *Sensor Testing* and *Making a Pressure Sensor*

```
1. Suppose you invited 12 friends to a pizza party.  Assuming
   each person will eat 3/8 of a pizza, what is the MINIMUM number
   of pizzas you will need to purchase.  Please show your work.
2. Multiply 15.3 X 23.21. Please show your work.
3. Suppose you work for ABC Corporation making $12.50 an hour.
   You have just been told you are getting a 10 percent raise.
   What is your new hourly pay?  Please show your work.
```

Listing 4.4: Emerging Scholars - Age Appropriate Math Questions

activities described in Section 4.1.2. We discussed careers in CS and concluded with administering the post-survey and quiz.

Evaluation. The evaluation instruments used in this pilot study were the same as those used in the Local Middle School pilot described in Section 4.4.2, with the exception of the three math questions included on the pre-quiz. The Emerging Scholars math questions are shown in Listing 4.4. The corresponding statistical analysis is described in Section 4.4.3. Tables 4.8, 4.9, and 4.10 detail the results of the three statistical analyses performed on the Emerging Scholars' data. For each of the three tables, the first column represents the survey statements found in Listing 4.1. The second, fourth, sixth, and eighth columns list the average scores for each statement of the pre-survey, across all pilots. The third, fifth, seventh, and ninth columns list the average scores for each statement of the post-survey. The post-survey scores shown in bold denote statistically significant changes in the mean response. With respect to the aggregate data, Table 4.8 shows that of the 20 statistical analyses performed across all pilots, 12 (60%) indicate statistically significant changes. The 5 statements fall into two groups, assessing interest in computer science (Statements 1-2) and content understanding (Statements 3-5), respectively.

Interest in Computer Science. The range of means for Statement 1 was 3.27 to 3.91. The median response for Statement 1 was 3 (moderately disagree) for ES14SB. The median response for each of the remaining groups – ES14SA, ES14JA, ES14JB – was 4 (moderately agree). The post-survey means for Statement 2 ranged from 2.73 to 3.68. Although none of the post-survey means were statistically significant, all groups showed an increase in interest in computer science. Further, both ES14Ja and ES14Jb had a post-survey median score of 4.

Survey Statement	**Average High-Low Change**
S1	.32
S2	.07
S3	-.02
S4	-.48
S5	.26
Quiz	6

Table 4.2: Emerging Scholars - Average Change of High/Low Difference (across all groups)

Content Understanding. Of the fifteen post-survey means across all groups, 100% showed a statistically significant increase in self-efficacy for Statements 3-5. The post-survey scores ranged from 4.18 to 5.

Content Quiz. The content quiz consisted of six true/false, multiple choice, and short answer questions. Each question was designed to gauge students' understanding of the material taught. Post-quiz means ranged from 70 to 78 percent. Across all groups, the post-quiz means showed a statistically significant increase, as one would expect.

Impact of Low Performers. A review of Tables 4.8 and 4.9 indicates there were minimal changes between the *aggregate* and *high* post-survey means. We took three additional steps to this assessment. First, we performed a statistical analysis on the data from students who did not answer at least two of the math questions correctly on the pre-quiz, as shown in Table 4.10. Next, we took the difference between the *high* and *low* results, shown in Table 4.11; parentheses indicate a negative difference. Finally, we calculated the average change of post scores for each statement and quiz, across all groups, shown in Table 4.2. The average change was in the range of -.48 to .32 on a 6 point scale. This indicates that students who answered no more than one math questions correctly had a minimal impact on the aggregate results.

In summary, the evaluation results suggest that the program had a positive impact. Students showed positive shifts in their attitudes toward CS. Post-means indicate that students understood the material taught. Further, the results suggest that the teaching approach applies equally well to both high and low performers.

4.5 Conclusion

We are surrounded by new technologies designed to enhance our standard of life – cell phones, household appliances, medical monitoring systems, security systems, and other innovations. Each relies fundamentally on sensors. Research and development in sensors and sensors networks is increasingly common, evidenced by more than 200 journals and conferences in the area [90]. Recognizing the importance of sensors in society and the importance of sensors and sensor networks in computer science, we chose the topics of sensors and sensor networks as the focus of the final curriculum module. The module includes a lecture component, demonstrations, hands-on activities, and a serious toy to support the concepts taught. We piloted the program with five middle school groups and four groups of high school students. The evaluations are largely positive. One particularly exciting outcome is that the teaching approach, focused on serious toys, applies equally well both to high and low performers.

Survey Statement	7AM pre	7AM post	7BF pre	7BF post	7CF pre	7CF post	7DM pre	7DM post	8AM pre	8AM post
S1	–	4.2	–	3.4	–	4	–	3.9	–	3
S2	3.24	4	2.25	3.05	2.55	**3.75**	3.25	**4.4**	2.63	3.5
S3	3.45	**5**	2.9	**4.65**	3.15	**5**	3.55	**4.55**	2.63	**4.94**
S4	2.48	**4.29**	1.9	**4.35**	2.05	**4.7**	2.5	**3.9**	1.63	**4.63**
S5	3.5	**4.84**	2.55	**4.6**	3.05	**4.9**	3.7	**4.6**	3.19	**4.81**
Quiz	61	**82**	63	73	52	**83**	65	70	67	77

Table 4.3: Langston Middle School - Aggregate Pre/Post-Survey and Quiz Means

Survey Statement	7AM pre	7AM post	7BF pre	7BF post	7CF pre	7CF post	7DM pre	7DM post	8AM pre	8AM post
S1	–	4.24	–	3.33	–	4.12	–	3.94	–	2.75
S2	3.47	4.18	1.83	**3**	2.59	**3.76**	3.29	**4.53**	2.42	3.25
S3	3.65	**5**	2.92	**4.92**	3.18	**5.05**	3.47	**4.47**	2.58	**4.83**
S4	2.47	**4.29**	1.5	**4.42**	2	**4.76**	2.53	**3.94**	1.33	**4.42**
S5	3.94	**5**	2.33	**5**	3.24	**5**	3.76	4.47	3.08	**4.58**
Quiz	63	**84**	67	76	51	**82**	66	71	72	79

Table 4.4: Langston Middle School - High Pre/Post-Survey and Quiz Means

Survey Statement	7AM pre	7AM post	7BF pre	7BF post	7CF pre	7CF post	7DM pre	7DM post	8AM pre	8AM post
S1	–	4	–	3.5	–	3.33	–	3.67	–	3.75
S2	2.25	3.25	2.88	3.13	2.33	**3.67**	3	3.67	3.25	4.25
S3	2.33	5	2.88	4.25	3	4.67	4	5	2.75	**5.25**
S4	2.5	4.25	2.5	**4.25**	2.33	**4.33**	2.33	3.67	2.5	**5.25**
S5	1.75	**4.25**	2.88	4	2	**4.33**	3.33	5.33	3.5	**5.5**
Quiz	54	75	52	66	56	87	61	67	52	71

Table 4.5: Langston Middle School - Low Pre/Post-Survey and Quiz Means

Survey Statement	7AM pre	7AM post	7BF pre	7BF post	7CF pre	7CF post	7DM pre	7DM post	8AM pre	8AM post
S1	–	.24	–	(.17)	–	.79	–	.27	–	(1.0)
S2	1.22	.93	(1.05)	(.13)	.26	.09	.29	.86	(.83)	(1.0)
S3	1.32	0	.04	.67	.18	.48	(.53)	(.53)	(.17)	(.42)
S4	(.03)	.04	(1.0)	.17	(.33)	.43	.20	.27	(1.17)	(.83)
S5	2.19	.75	(.55)	1.0	1.24	.67	.43	(.86)	(.42)	(.92)
Quiz	9	9	15	10	(5)	(5)	5	4	20	8

Table 4.6: Langston Middle School - High/Low Difference

	7AM	7BF	7CF	7DM	8AM
Number of participants	21	20	20	20	16
Students in *low* group	4	8	3	3	4
Percentage	19%	40%	15%	15%	25%

Table 4.7: Langston Middel School - Percentage: Low Performers/Participants

Survey Statement	ES14SA pre	ES14SA post	ES14SB pre	ES14SB post	ES14JA pre	ES14JA post	ES14JB pre	ES14JB post
S1	–	3.83	–	3.27	–	3.91	–	3.78
S2	3.17	3.58	2.64	2.73	3.23	3.68	3.33	3.67
S3	3.42	**4.92**	2.73	**4.18**	2.64	**4.73**	3.19	**4.76**
S4	2.08	**4.92**	1.82	**4.18**	1.77	**4.45**	2.24	**4.48**
S5	3.17	**5**	3.27	**4.45**	2.2	**4.5**	2.86	**4.81**
Quiz	50	**72**	67	**70**	43	**76**	54	**78**

Table 4.8: Emerging Scholars - Aggregate Pre/Post-Survey and Quiz Means

Survey Statement	ES14SA pre	ES14SA post	ES14SB pre	ES14SB post	ES14JA pre	ES14JA post	ES14JB pre	ES14JB post
S1	–	3.71	–	4.04	–	3.71	–	4.04
S2	4	4	2.6	2.2	3.43	3.71	3.54	3.92
S3	3.67	4.67	2.6	4.2	2.29	**5**	3.08	**4.62**
S4	1.67	**5.33**	2.2	**4**	1.29	**4.14**	2.38	**4.46**
S5	4.3	5	3.8	4.6	2	**4.71**	**3**	**4.85**
Quiz	50	78	53	**80**	48	**75**	58	**78**

Table 4.9: Emerging Scholars - High Pre/Post-Survey and Quiz Means

Survey Statement	ES14SA pre	ES14SA post	ES14SB pre	ES14SB post	ES14JA pre	ES14JA post	ES14JB pre	ES14JB post
S1	–	3.89	–	3.17	–	4.15	–	3.38
S2	2.89	3.44	2.67	3.17	3.23	3.69	3	3.25
S3	3.33	**5**	2.83	4.17	3	**4.62**	3.56	**4.89**
S4	2.22	**4.78**	1.5	**4.33**	2.08	**4.62**	2	**4.5**
S5	2.78	**5**	2.83	**4.33**	2.36	**4.45**	2.63	**4.75**
Quiz	50	**69**	58	64	39	**83**	48	**73**

Table 4.10: Emerging Scholars - Low Pre/Post-Survey and Quiz Means

Survey Statement	ES14SA pre	ES14SA post	ES14SB pre	ES14SB post	ES14JA pre	ES14JA post	ES14JB pre	ES14JB post
S1	–	.18	–	.87	–	(.44)	–	.66
S2	1.11	.56	(.07)	(.97)	.20	.02	(.54)	.67
S3	.34	(.33)	(.23)	.03	(.71)	.38	(.48)	(.27)
S4	(.55)	(.55)	.70	(.33)	(.79)	(.48)	(.38)	(.04)
S5	1.52	0	.97	.27	(.36)	.26	.37	.10
Quiz	0	9	(5)	16	9	(8)	10	5

Table 4.11: Emerging Scholars - High/Low Difference

Chapter 5

Combined Program

While the modules described in Chapters 2, 3, and 4 are designed to enable incremental, independent adoption, together they form a coherent thread of instruction. To evaluate the effectiveness of the combined program in positively impacting students' content understanding and attitudes toward computer science, we piloted the program using all three modules.

Depending on participants' class schedules, the combined program spans 8-10 days. Local middle school class periods are shorter than local high school periods – 40 minutes for middle schoolers, and 70 minutes for high schoolers. On the first day of the program, we introduce ourselves and administer the pre-evaluation instruments described in Sections 5.2.1.1 and 5.2.2.1. The next six to eight days consist of lecture, and activities, beginning with the Binary Number System module, followed by the Networks, Protocols, and Algorithms module, and finally, the Sensors and Sensor Networks module. The last day of the program begins with a short review of each module and a discussion on careers in computing. Students are then given a hands-on opportunity to play with, explore, and ask questions about the *serious toys*. The program concludes with the administration of the post-evaluation instruments described in Sections 5.2.1.1 and 5.2.2.1.

Program Modifications. The GSR activity described in Section 3.1.3 is typically a component of the (independent) Networks, Protocols, and Algorithms module. Combined with the other modules, however, the GSR sensor demonstration is a better fit for the Sensors and Sensor Networks module. Hence, in the combined program, we move the activity to this module.

5.1 Pilot Groups

We piloted the combined program with two local middle schools and a local high school – six groups from the first middle school, five from the second, and one from the high school.

5.1.1 Local Middle Schools

The first participating middle school is described in Section 3.2.1. The pilot groups consisted of two classes each from the sixth, seventh, and eighth grades. The groups were chosen from a technology exploratory course. This course consists of two tracks – Introduction to Careers and Gateway to Technology. The two sixth grade classes participated in Introduction to Careers, where students explore careers in STEM fields. The seventh and eighth grade groups participated in Gateway to Technology, which provides an introduction to various topics in engineering, including robotics, 3D modeling, and other topics. The second participating middle school is a charter middle school described in Section 4.3.1. The pilot groups consisted of five single-gendered science classes.

5.1.2 Local High School

The participating high school is one school within a K-12 charter system. The school qualifies as a Title 1 institution, with a high percentage of at-risk students. The school system has approximately 800 students, with 95% living in poverty. The pilot group consisted of eighth, ninth, and tenth grade students enrolled in a geometry course.

5.2 Pilot Studies

In this section, we discuss three pilot studies. The first two were conducted during the fall of 2013. After minor modifications to the evaluation procedure and instruments, discussed in Section 4.4.1, the third pilot was conducted in the spring of 2014.

5.2.1 Fall 2013

During the fall of 2013, the combined modules were piloted with the first local middle school described in Section 5.1.1, and the local high school described in Section 5.1.2.

```
1. I would like to attend more outreach programs related to CS.
2. I might be interested in majoring in CS in college.
3. I think I understand the importance of binary numbers in CS.
4. I think I understand the concept of a binary numbering system.
5. I think I know the definitions of a network, a protocol, and
   an algorithm.
6. I think I understand the importance of networks, protocols,
   and algorithms in CS.
7. I Think I understand the importance of sensors and sensor
   networks in CS.
8. I think I understand what sensors are used for.
```

Listing 5.1: Survey Statements

5.2.1.1 Evaluation Instruments

The evaluation instruments consisted of a survey and multiple quizzes. The survey and a quiz was administered on the first and last day of the program. After completion of each of the three modules, a module-specific quiz was administered. The survey was designed to determine student attitudes toward CS, and students' self-efficacy with respect to their understanding of the material covered. It consisted of eight Likert-style statements, shown in Listing 5.1. The response weights are described in Section 2.3. The quizzes consisted of true/false, multiple choice, and free response questions designed to evaluate student understanding of the material covered. The pre- and post-quiz, shown in Listing 5.2, consisted of 8 questions, 3 pertaining to The Binary Number System module, 3 pertaining to the Networks, Protocols, and Algorithms module, and 2 pertaining to the Sensors and Sensor Networks module. The additional quizzes were administered after the completion of each module and consisted of 6-8 questions specific to the module.

5.2.1.2 Local Middle School

The middle school classes are referred to by their grade and class section; e.g., 6A refers to sixth grade, section A. For this pilot, each class period was approximately 40 minutes long; the study spanned ten days. Table 5.1 summarizes the total student participation on the first and last day of the program and the number of students identified by the school as needing special learning considerations. In total, 155 participants completed the pre-survey and quiz, and 126 participants completed the post-survey and quiz. Survey and quiz results were only considered if participants completed both pre and post instruments. The table also shows that there were 20 participants with special learning considerations. The evaluation process was anonymous; therefore,

```
1. A group of entities connected together using a communication
   channel is called a ______.
   a) Forest  b) Post Office c) Network d) Station
2. A(n) ______ is a set of planned actions to complete a
   given task.
   a) Algorithm b) Program c) Job d) Instruction
3. A set of rules for exchanging messages between two entities is
   is called a(n) ______.
   a) Algorithm b) Program c) Network d) Protocol
4. Decimal number 34 can be written as ______ in binary.
   a) 11001 b) 10101 c) 100010 d) 11001
5. Perform the following binary addition.
   1100
  +0110
  ------
   a) 1000 b) 0100 c) 1010 d) 0001
6. Perform the following binary subtraction.
   1101
  -1001
  ------
   a) 1000 b) 0100 c) 1010 d) 0001
7. The Principle function of a sensors to create and measure an
   environmental change.
   a) True b) False
8. A Wireless Sensor Network is a group of sensor nodes that
   collect data and send the data directly to a main computer.
   a) True b) False
```

Listing 5.2: Pre-/Post-Quiz Fall 2013

these students received no additional help completing the pre- and post-survey, the pre- and post-quiz, or supplemental content quizzes.

Evaluation. Table 5.2 summarizes the results of a statistical analysis of pre- and post-survey results and pre- and post-quiz results for the middle school pilot groups. The first column of the table refers to survey statement. The even-numbered columns list the average scores for the pre-survey and quiz. The remaining columns list the average scores for the post-survey and the post-quiz. The post averages shown in bold represent statistically significant increases. Statistical significance was determined in the same manner described in Section 2.3. Statements 1 and 2 measure student attitudes toward CS. The remaining six statements measure students' self-efficacy with respect to their understanding of the material covered. Table 5.3 shows the module-specific quiz means for

Participants	**6A**	**6B**	**7A**	**7B**	**8A**	**8B**	**Total**
First Day	23	22	26	29	29	26	155
Last Day	21	17	24	24	20	20	126
Special Considerations	3	2	3	4	6	2	20

Table 5.1: Local Middle - Fall 2013 - School Participation Information

Survey Statements	**6A pre**	**6A post**	**6B pre**	**6B post**	**7A pre**	**7A post**	**7B pre**	**7B post**	**8A pre**	**8A post**	**8B pre**	**8B post**
S1	—	3.14	—	3.65	—	4.46	—	3.83	—	3.8	—	3.45
S2	3	2.81	2.94	3.06	4.21	4.29	3.67	3.71	2.85	3.65	2.95	3.85
S3	3.14	**4.76**	2.88	**4**	3.67	4.17	1.91	**4.30**	2.3	**4.45**	2.8	**4.6**
S4	2.41	**4.52**	2.41	**4.06**	3	**4.3**	1.96	**4.5**	2.05	**4.5**	2.5	**4.5**
S5	2.91	**4.52**	3.29	**4.35**	3.417	**4.67**	3	**4.46**	2.6	**4.45**	3.1	**4.65**
S6	3.33	**4.48**	3.83	4.5	3.83	**4.58**	3.58	4.04	2.52	**4.8**	2.9	**4.95**
S7	4	4.38	3.77	4.65	4.04	4.67	3.25	**4.46**	2.95	**4.7**	3.19	**4.75**
S8	3.62	**4.95**	3.81	**4.94**	4.04	4.67	3.71	4.13	3	**4.3**	3.	**4.75**
Combined Quiz	39	**60**	39	**58**	40	48	40	**56**	42	**67**	38	**62**
Rev. Combined Quiz	43	**67**	38	**83**	43	54	43	**64**	43	**76**	41	**75**

Table 5.2: Local Middle School - Fall 2013 - Pre-/Post-Survey Results

Module Specific Quiz	**6A**	**6B**	**7A**	**7B**	**8A**	**8B**
Binary	57	63	60	64	74	62
Network	83	62	68	66	76	73
Sensors	53	51	48	63	56	55
Revised Sensors	67	57	59	80	70	61

Table 5.3: Local Middle School - Fall 2013 - Module - Specific Quiz Means

each group. The following subsections discuss the analysis of the pre- and post-survey, pre- and post-quiz, and the supplemental quizzes.

Interest in Computer Science. Statement 1 concerns the outreach program and could not be answered until completion of the program; only post-survey data was considered. As shown in Table 5.2, the average scores were mostly positive, ranging from 3.14 to 4.46, with a median of 4 (moderately agree) or above for all groups, except 6A. Statement 2, designed to gauge student interest in CS as a career choice, saw increases for all groups, except 6A; none of the changes were significant, however.

Content Understanding. Statements 3-8 were designed to gauge students' self-efficacy with respect to the material taught. Table 5.2 shows that across all groups, all response means increased, with 78% (28 out of 36) of the means showing a statistically significant increase.

Content Quiz. Pre- and post-quizzes and module specific quizzes were used to measure content understanding. The pre- and post-quiz means, shown in Table 5.2, indicate the quiz averages

```
1. Decimal number 34 can be written as ______ in binary.
   a) 11001   b) 10101   c) 100010   d) 11011
2. There is/are only ______ digit(s) in the binary number system.
   a) 1   b) 2   c) 10   d) infinite
3. 1001 in binary is the same as ______ in decimal.
   a) 7   b) 8   c) 10   d) 9
4. Perform the following binary addition.
     1100
   + 0110
    ---------
    a) 10100   b) 10010   c) 10000   d) 0100
5. Perform the following binary subtraction.
     1101
   - 1001
    ---------
   a) 1000   b) 0100  c) 1010  d) 0001
6. Binary numbers are important in Computer Science.  Why?
   a) They are easier to learn.
   b) They are best suited to represent the ON and OFF
      states of electronic switches.
   c) Decimal numbers are not that popular among computer
      scientists and engineers.
```

Listing 5.3: Local Middle School - Fall 2013 - Supplemental Binary Module Quiz

increased across all groups, with 83% (5 out of 6) of the increases being statistically significant. After the completion of each module, a supplemental quiz was administered. The module-specific quizzes are shown in Listings 5.3, 5.4, and 5.5. The results shown in Table 5.3 indicate that the supplemental quiz averages for the Sensors and Sensor Networks module were lower than the averages for the Binary Number System and Networks, Protocols, and Algorithms modules. Further investigation revealed that there were two questions on the Sensors and Sensor Networks module-specific quiz that most students answered incorrectly. These questions were also included on the pre- and post-quiz. It was determined that these questions were ambiguous; therefore, the averages were re-calculated excluding these two questions. As indicated in Tables 5.2 and 5.3, the revised scores increased, across the six groups by 11 points, on average. Considering that approximately 13% of the students participating on the first day of the pilot had special learning needs, the average scores, across all groups, are mostly positive.

Overall, the data suggests that the combined program had a positive impact. Students exhibited positive shifts in their attitudes toward CS as a discipline and as a career option. With

```
1. A group of entities connected together using a communication
   channel is called a ______.
   a) Forest b) Post Office c) Network d) Station
2. A(n) ______ is a set of planned actions to complete a given task.
   a) Algorithm b) Program c) Job d) Instruction
3. A set of rules for exchanging messages between two entities
   is called a(n) ______.
   a) Algorithm b) Program c) Network d) Protocol
4. If an unexpected event happens, a computer is smart and can
   determine by itself what action it needs to take.
   a) True b) False
5. Each computer in a network can use a different protocol and
   still be able to communicate with all other computers on
   the network.
   a) True b) False
6. You must have a computer to form a network.
   a) True b) False
7. It is important for each device/computer in a network to have a
   unique identity.
   a) True b) False
8. There are no problems associated with allowing one computer
   in a network to control the actions of all other computers.
   a) True b) False
9. A cable TV network is an example of a centrally managed network.
   a) True b) False
10. The Internet is an example of a distributed computer system.
   a) True b) False
```

Listing 5.4: Local Middle/High School - Fall 2013 - Supplemental Network Module Quiz

respect to content understanding, student post-means indicate that they were confident in their understanding of the material taught.

5.2.1.3 Local High School

The high school pilot group, referred to as HS, consisted of 17 students, with 16 completing both the pre- and post-surveys and pre- and post-quiz. Participants consist of 2 eighth, 11 ninth, and 3 tenth grade students. The class period was approximately 70 minutes, the pilot spanned 8 days.

Evaluation. Table 5.4 depicts the means for pre- and post-survey and pre- and post-quiz for the high school pilot group. Column one refers to the survey statements and the quiz. Columns two and three list the average scores for the pre- and post-survey and pre- and post-quiz. As in the

```
1. The principle function of a sensor is to create and measure and
   environmental change.
   a) True b) False
2. A Wireless Sensor Network is a group of sensor nodes that
   collect data and send the data directly to a main computer.
   a) True b) False
3. Your nose is a sensor.
   a) True b) False
4. A/An ______ consist of spatially distributed sensors that monitor
   physical or environmental conditions and pass the data through
   a network to a main location.
   a) Gateway Sensor Nodes
   b) Wireless Sensor Network
   c) Internet
   d) Post Office
5. If a sensor node in a wireless sensor network breaks, any sensor
   node that passes data to the broken sensor node must wait until
   the broken sensor node is fixed before it can send data again.
   a) True b) False
6. A homemade ______ can be made using plaster-of-paris, a straw,
   and two nails.
   a) Motion sensor
   b) Galvanic Skin Response Sensor
   c) Soil Moisture Sensor
   d) Pressure Sensor
```

Listing 5.5: Local Middle School/High School - Fall 2013 - Supplemental Sensor Module Quiz

earlier pilots, the post averages shown in bold represent statistically significant increases. Statistical significance was determined in the same manner as discussed in Section 2.3. Survey Statements 1 and 2 measure student attitudes toward CS. The remaining six statements measure students' self-efficacy with respect to their understanding of the material covered. Table 5.5 show the results of the module-specific quiz averages.

Interest in Computer Science. Only post-survey data was considered for Statement 1. Table 5.4 shows that, on average, students would like to participate in more CS outreach. Statement 2 results show that student interest in CS appears to have increased, though the change is not statistically significant. The post average for Statement 2 responses was 3.9 (median score of 4). A closer examination of the data indicates that 10 out of 16 students' post responses were 4 or higher, compared to 8 out of 16 of the pre responses. This indicates an increase in student interest in majoring in CS in college.

```
1. Decimal number 34 can be written as ______ in binary.
    a) 11001   b) 10101   c) 100010   d) 11011
2. There is/are only ______ digit(s) in the binary number system.
    a) 1   b) 2   c) 10   d) infinite
3. 1001 in binary is the same as ______ in decimal.
    a) 7   b) 8   c) 10   d) 9
4. What is the largest decimal number that can be represented
   using 4 binary digits?
   a) 15  b) 17 c) 14 d) 33
5. Can you represent decimal number 18 using 4 bits?
   (Please provide a reason for you answer.)
   a) True b) False  Reason: ______
6. Perform the following binary addition.
      1100
    + 0110
    ---------
   a) 10100   b) 10010   c) 10000   d) 0100
7. Perform the following binary subtraction.
      1101
    - 1001
    ---------
   a) 1000   b) 0100  c) 1010  d) 0001
8. Binary numbers are important in Computer Science.  Why?
   a) They are easier to learn.
   b) They are best suited to represent the ON and OFF
      states of electronic switches.
   c) Decimal numbers are not that popular among computer
       scientists and engineers.
```

Listing 5.6: Local High School - Fall 2013 - Supplemental Binary Quiz

Content Understanding. Statements 3-8 were designed to gauge student content understanding. Table 5.4 shows that 100% of post-survey results indicate a statistically significant increase in students' self-efficacy with respect to their understanding of the material taught. The post-survey means range from 5.19 to 5.56, where 5 represents *agree.*

Content Quizzes. Table 5.4 shows a statistically significant increase in the post-quiz average. Table 5.5, shows the average score for the supplemental quizzes, administered after the completion of each module. The supplemental quizzes are shown in Listings 5.4, 5.5 and 5.6. The results, from both tables, show that students understood the material taught. Similar to the middle school pilot groups, many of the high school students incorrectly answered one or both of the two ambiguous questions on the sensor quiz. The questions were again discarded, increasing the post-quiz score by 22 points, and the supplemental sensor quiz average by 32 points.

Survey Statement	Local High School pre	Local High School post
S1	—	4.47
S2	3.31	3.94
S3	2.5	**5.31**
S4	2.31	**5.31**
S5	2.88	**5.44**
S6	3	**5.19**
S7	3.13	**5.19**
S8	3.5	**5.56**
Combined Quiz	48	**70**
Revised Combined Quiz	55	**92**

Table 5.4: Local High School - Fall 2013 - Pre-/Post-Survey and Quiz Results

Module-Specific Quiz	Local High School Average Scores
Binary	79
Network	79
Sensors	58
Revised Sensors	90

Table 5.5: Local High School - Fall 2013 - Module-Specific Quiz Means

In conclusion, the analysis suggests that the pilot had a positive impact on student attitudes toward CS. On average, students exhibited high self-efficacy with respect to the content taught. Quiz averages also indicate that students understood the material taught.

5.2.2 Spring 2014

During the spring of 2014, the combined program was piloted with five single-gendered classes at a local middle school – three male and two female. The groups are identified by their grade, section, and gender; e.g., 6AF refers to sixth grade, section A, female. The class periods were approximately 50 minutes; the program spanned 10 days.

5.2.2.1 Evaluation Instruments

The evaluation instruments were the same as previous pilots. The pre- and -post-survey includes eight Likert-style statements, shown in Listing 5.1. The response weights were the same as in previous pilot studies. The pre- and post-quiz and supplemental quizzes consisted of the same types of questions as in previous pilots. However, prior to beginning of this pilot, all quizzes were

```
1. Decimal number 42 (base 10) can be written as ______ in binary
   (base 2). Please show your work.
   a) 110010 b) 101010 c) 100010 d) 110110
2. Perform the following binary addition.
   1110
  +0110
  ------
   a) 1000 b) 0100 c) 1010 d) 0001
3. Each computer in a network can use a different protocol and still
   be able to communicate with all other computers in the network.
   a) True b) False
4. An algorithm consists of a set of step-by-step instructions.
   These instructions must always be completed in a specific order.
   a) True b) False
5. The principle function of a sensor is to ______ change.
   a) create b) detect c) correct d) none of the above
6. A sensor can be constructed using common household products.
   a) True b) False
```

Listing 5.7: Local Middle School - Spring 2014 - Pre-/Post-Quiz

```
1. Suppose you invited 12 friends to a pizza party.  Assuming
   each person will eat 3/8 of a pizza, what is the MINIMUM number
   of pizzas you will need to purchase.  Please show your work.
2. Add 3/10 + 4/100. Please show your work.
3. Multiply 147 X 32. Please show your work.
```

Listing 5.8: Local Middle School - Spring 2014 - Age Appropriate Math Questions

modified to correct the ambiguous questions described in Section 5.2.1. The pre- and post-quiz, shown in Listing 5.7, consisted of 6 questions, 2 pertaining to The Binary Number System module, 2 pertaining to the Network, Protocols, and Algorithms module, and 2 pertaining to the Sensors and Sensor Networks module. As described in Subsection 4.4.1, three age appropriate math questions, shown in Listing 5.8, were added to the pre-quiz. The supplemental quizzes administered after the completion of each module consisted of 6-8 questions specific to the module, as shown in Listings 5.9, 5.10, and 5.11.

Evaluation. A statistical analysis of the pre- and post-survey and pre- and post-quiz was performed across three subsets of data. First, an analysis was performed on all student data. The results are shown in Table 5.6, referred to as the *aggregate* table. Next, only data from students who answered at least two of the three math questions correctly were used in the analysis summarized in Table 5.7, referred to as the *high* table. Finally, an analysis was performed on data from students who did not answer at least two of the three math questions correctly. The results are shown in Table 5.8, referred to as the *low* table. For each of the tables, the first column denotes the survey statements, shown in Listing 5.1 and the quiz, shown in Listing 5.7. The even-numbered columns show the average scores for each statement of the pre-survey. The post-survey average scores are listed in the remaining columns. Statistical significance was determined in the same manner as discussed in Section 2.3. The *aggregate* table shows that of the 40 statistical analyses performed, 35 (88%, across all groups) indicate a statistically significant increase of the mean. The eight statements are categorized in two groups, concerning interest in CS and student self-efficacy, respectively. Tables 5.10, 5.11, 5.12 and 5.13 show the means for the module-specific quizzes for each group.

Interest in Computer Science. As shown in Table 5.6, post-survey means for Statement 1 ranged from 2.35 to 3.76 (moderately disagree). Three of the five groups had a median score of 4 (moderately agree). Analysis of Statement 2 showed that 3 of the 5 (60%) groups' survey response means increased, though not significantly. Statement 2 had a median score of 4 for two of the groups.

Content Understanding. Statements 3-8 gauge student self-efficacy with respect to the content taught. All post-survey means across all pilots showed a significant increase. The range for all statements was 4.52 to 5.30.

Content Quizzes. Table 5.6 shows 100% of post-quiz means increased significantly. Post-quiz scores ranged between 66% and 75%. At the conclusion of each module, students completed a supplemental quiz, shown in Listings 5.9, 5.10, and 5.11. The means shown in Table 5.10, ranged between 73% to 83% for the Binary Number System module, 60% to 76% for the Networks, Protocols, and Algorithms module, and 72% to 87% for the Sensors and Sensor Networks module.

Impact of Low Performers. An analysis of the data from students who correctly answered at least two of the three math questions included on the pre-quiz shown in Table 5.7 revealed there was little difference between the *high* and *aggregate* tables. This suggests that the impact of *low* performers was minimal. A statistical analysis was performed on data from students who did not correctly answer at least two of the supplemental math questions; results are shown in Table 5.8. Next, we calculated the difference between the *high* and *low* results; the results are shown in Table 5.9.

As in previous pilots, parentheses indicate a negative difference. Finally, we calculated the average change in post scores for each statement and quiz, shown in Table 5.14. The range of change for post-survey statements falls between -.50 and .04 on a six point scale. Note that seven of the eight changes are negative. In fact, 75% of the post-survey means shown in Table 5.9 are negative. This indicates that students in the low category scored higher than those in the high category, suggesting that students in the low category exhibited high self-efficacy. However, the post-quiz scores indicate students in the high category had a better understanding of the material taught. Table 5.9 shows only 20% of the post-quiz scores were negative. This indicates students in the high group scored higher on the post-quiz. Tables 5.10, 5.11, and 5.12 show the average module-specific quiz scores for the aggregate, high, and low data sets, respectively. Table 5.13 shows the difference between high and low results on the module-specific quizzes. Note only 43% of the responses were negative. The average change in module-specific quiz results is summarized in Table 5.15, with a range of -2 to 13 on a 100 point scale. The results again show that the combined program provides an effective teaching approach for both high and low performers.

In summary, the analyses indicate that the program is having a positive impact. The majority of groups showed a positive attitudinal shift in interest towards CS as a career. All groups showed an increase in self-efficacy with respect to content understanding. Overall, post quiz averages and module-specific quiz averages suggest that student, in both high and low performing groups, understood the material taught.

5.3 Conclusion

Each module described in Chapters 2, 3, and 4 is designed to facilitate independent adoption. However, when combined they form a coherent thread of instruction. We developed a combined program to evaluate the effectiveness of the program in positively impacting students' content understanding and attitudes toward computer science. We piloted the modules with two local middle schools, and a local high school. The evaluations are largely positive across all groups.

Survey Statements	6AF pre	6AF post	6BM pre	6BM post	6CM pre	6CM post	7AM pre	7AM post	7BF pre	7BF post
S1	–	2.35	–	3.76	–	3.7	–	3.62	–	3.05
S2	2.05	2.10	3.24	3.38	3.5	3.25	2.90	3.24	3.14	2.67
S3	2.6	**4.55**	2.52	**5.10**	2.1	**4.8**	3.19	**4.62**	2.10	**4.67**
S4	2.05	**5.05**	2.19	**5.10**	1.9	**4.8**	2.57	**4.48**	1.67	**4.52**
S5	2.6	**4.95**	2.67	**5**	2.7	**5**	3.48	**5.19**	2.62	**4.95**
S6	2.7	**4.8**	3.19	**5.24**	2.9	**5.15**	3.81	**4.95**	2.52	**4.90**
S7	2.9	**4.95**	3.57	**5**	3.4	**5.2**	3.48	**5.14**	2.62	**4.86**
S8	3.25	**4.9**	4.29	**5.14**	4.2	**5.3**	3.3	**5.2**	2.95	**5.19**
Quiz	32	**68**	36	**74**	37	**66**	25	**75**	19	**70**

Table 5.6: Local Middle School - Spring 2014 - Aggregate - Survey and Quiz Means

Survey Statements	6AF pre	6AF post	6BM pre	6BM post	6CM pre	6CM post	7AM pre	7AM post	7BF pre	7BF post
S1	–	2.14	–	3.76	–	3.63	–	3.4	–	2.92
S2	1.78	1.92	3.18	3.29	3.44	3.06	2.67	3.07	3.15	2.69
S3	2.43	**4.43**	2.53	**5.06**	2.19	**4.56**	3.2	**4.53**	2	**4.62**
S4	1.86	**4.93**	2.35	**5.06**	1.76	**4.63**	2.6	**4.47**	1.46	**4.54**
S5	2.57	**4.93**	2.53	**5.06**	2.47	**5**	3.33	**5.2**	2.31	**4.92**
S6	2.57	**4.57**	3	**5.12**	2.76	**5.13**	3.67	**4.93**	2.62	**4.62**
S7	2.43	**4.86**	3.35	**5**	3.24	**5.19**	3.33	**5.13**	2.39	**4.69**
S8	2.86	**4.93**	4.12	**5.12**	3.88	**5.31**	3.47	**5.27**	3.15	**5.15**
Quiz	30	**71**	38	**78**	33	**66**	27	**77**	21	**72**

Table 5.7: Local Middle School - Spring 2014 - High - Survey and Quiz Means

Survey Statements	6AF pre	6AF post	6BM pre	6BM post	6CM pre	6CM post	7AM pre	7AM post	7BF pre	7BF post
S1	–	2.83	–	3.75	–	4	–	4.17	–	3.25
S2	2.69	2.5	3.5	3.75	3.75	4.0	3.5	3.67	3.13	2.63
S3	3	4.83	2.5	**5.25**	1.75	**5.75**	3.17	4.83	2.25	**4.75**
S4	2.5	**5.33**	1.5	**5.25**	2.25	**5.5**	2.5	**4.5**	2	**4.5**
S5	2.67	**5**	3.25	**4.75**	3.75	5	3.83	**5.17**	3.13	**5**
S6	3	**5.33**	4	5.75	2.75	**5.75**	4.17	5	2.63	**5.38**
S7	4	**5.17**	4.5	5	3.5	**5.25**	3.83	5.17	3	**5.13**
S8	4.17	4.83	5	5.25	4.75	5.25	2.8	**5**	2.63	**5.25**
Quiz	36	**61**	29	58	46	**67**	19	**69**	17	**67**

Table 5.8: Local Middle School - Spring 2014 - Low - Survey and Quiz Means

Survey Statements	6AF pre	6AF post	6BM pre	6BM post	6CM pre	6CM post	7AM pre	7AM post	7BF pre	7BF post
S1	–	(.69)	–	.01	–	(.37)	–	(.77)	–	(.33)
S2	(.91)	(.58)	(.32)	(.46)	(.31)	(.94)	(.83)	(.6)	.02	.06
S3	(.57)	(.4)	.03	(.19)	.44	(1.19)	.03	(.3)	(.25)	(.13)
S4	(.64)	(.4)	.85	(.19)	(.49)	(.87)	.1	(.03)	(.54)	.04
S5	(.1)	(.07)	(.72)	.31	(1.28)	0	(.5)	.03	(.82)	(.08)
S6	(.43)	(.76)	(1)	(.63)	.01	(.12)	(.5)	(.07)	(.01)	(.76)
S7	(1.57)	(.31)	(1.15)	0	(.26)	(.06)	(.5)	(.04)	(.61)	(.44)
S8	(1.31)	.1	(.88)	(.13)	(.87)	.06	.67	.27	.52	(.1)
Quiz	(6)	10	9	20	(12)	(1)	8	8	4	5

Table 5.9: Local Middle School - Spring 2014 - High/Low - Difference - Survey and Quiz Means

Module-Specific Quiz	6AF	6BM	6CM	7AM	7BF
Binary	77	78	74	83	73
Network	76	75	60	70	73
Sensor	81	80	83	88	67

Table 5.10: Local Middle School - Spring 2014 - Aggregrate - Module-Specific Quiz Means

Module-Specific Quiz	6AF	6BM	6CM	7AM	7BF
Binary	82	78	77	85	79
Network	75	73	59	71	73
Sensor	83	78	84	87	72

Table 5.11: Local Middle School - Spring 2014 - High - Module-Specific Quiz Means

Module-Specific Quiz	6AF	6BM	6CM	7AM	7BF
Binary	61	78	58	75	63
Network	81	79	63	67	71
Sensor	75	92	77	90	62

Table 5.12: Local Middle School - Spring 2014 - Low - Module-Specific Quiz Means

Module-Specific Quiz	6AF	6BM	6CM	7AM	7BF
Binary	21	0	19	10	16
Network	(6)	(6)	(4)	4	2
Sensor	(8)	(14)	7	(3)	10

Table 5.13: Local Middle School - Spring 2014 - High/Low - Module-Specific Quiz Means

Survey Statement	Average High-Low Change
S1	-.43
S2	-.50
S3	-.44
S4	-.29
S5	-.04
S6	-.47
S7	-.17
S8	.04
Quiz	8

Table 5.14: Local Middle School - Spring 2014 - Average Change of High-Low Means

Module-Specific Quiz	Average High-Low Change
Binary	13
Network	-2
Sensors	-2

Table 5.15: Local Middle School-Spring 2014 - Module-specific - Average Change of High-Low Means

```
1. Decimal number 34 (base 10) can be written as ______ in binary.
   a) 11001   b) 10101   c) 100010   d) 11011
2. 1001 in binary is the same as ______ in decimal.
   a) 7   b) 8   c) 10   d) 9
3. Perform the following binary addition.
     1100
   + 0110
   ---------
4. Perform the following binary subtraction.
     1101
   - 1001
   ---------
5. Binary numbers are important in Computer Science.  Why?
   a) They are easier to learn.
   b) They are best suited to represent the ON and OFF
      states of electronic switches.
   c) Decimal numbers are not that popular among computer
      scientists and engineers.
6. Decimal number 34 (base 10) can be written as ______ in base 3.
   Please show your work.
   a) 1021 b) 1200 c) 1112 d) 1002
```

Listing 5.9: Local Middle School - Spring 2014 - Supplemental Binary Quiz

```
1. A group of entities connected together using a communication
   channel is called a ______.
   a) Forest b) Post Office c) Network d) Station
2. A(n) ______ is a set of planned actions to complete a given task.
   a) Algorithm b) Program c) Job d) Instruction
3. A set of rules for exchanging messages between two entities
   is called a(n) ______.
   a) Algorithm b) Program c) Network d) Protocol
4. Each computer in a network can use a different protocol and
   still be able to communicate with all other computers on
   the network.
   a) True b) False
5. There are no problems associated with allowing one computer
   in a network to control the actions of all other computers.
   a) True b) False
6. An algorithm consists of a set of step-by-step instructions.
   These instructions must always be completed in a specific order.
   a) True b) False
```

Listing 5.10: Local Middle School - Spring 14 - Supplemental Network Quiz

```
1. The principle function of a sensor is to ______ a physical or
   environmental condition.
   a) Create b) Ignore c) Measure d) None of the above
2. When a sensor node in a wireless sensor network collects data
   it sends the data to each sensor node in the network.
   a) True b) False
3. An example of a sensor on the human body is a nose.
   a) True b) False
4. A ______ consists of a group of sensors that monitor physical
   or environmental conditions and pass the data through a network
   to a main location.
   a) Gateway Network
   b) Sensor Network
   c) Network of Laptop Computers
   d) None of the above
5. A sensor can be constructed using common household products.
   a) True b) False
6. Name two applications of a network of sensors.
```

Listing 5.11: Local Middle School - Spring 2014 - Supplemental Sensor Quiz

Chapter 6

Related Work

In this chapter, we summarize three categories of related work. Section 6.1 discusses K-12 outreach efforts that focus on teaching computer science concepts using CS Unplugged activities, and basic programming skills using the Scratch and Alice programming environments. Section 6.2, first, discusses the use of digital manipulatives in K-12 outreach. Next, the use of virtual and physical manipulatives in education is discussed. Section 6.3 discusses efforts to teach K-12 students the specific computer science concepts incorporated in the curriculum modules being developed. Section 6.4 discusses the pedagogical philosophy of Active Learning.

6.1 Outreach Efforts

For more than a decade, significant energy has been expended on computing programs targeting K-12 students. In this section, we highlight programs that teach students computer science concepts using the popular CS Unplugged [24] activities, and teach basic programming skills using popular programming environments such as Scratch [54] and Alice [17], which are designed to make programming accessible to K-12 students.

6.1.1 CS Unplugged

Similar to our curriculum modules, CS Unplugged modules are designed to teach specific computer science concepts to K-12 students. CS Unplugged teaches these concepts using game and puzzle-like activities. Our modules rely on a digital manipulative and are designed to appeal to students of all learning styles.

Mano *et al.* [60] introduce an outreach program designed to increase student interest in computer science. They piloted their approach at a local middle school. Volunteers visited 4 classes for approximately 45 minutes each month. Activities used in this program included disassembling and reassembling a computer, learning how to program with Alice, and using CS Unplugged activities. Survey results and volunteers' observations suggest that the program had a positive impact on student interest in computer science. Similar to their work, we evaluate the impact of our modules on student interest in computer science. We are also interested in measuring students' understanding of the concepts being taught, as well as students' interest in participating in additional computer science outreach programs.

Lambert *et al.* [50] similarly report their experiences in introducing a set of CS Unplugged activities to a group of fourth graders. The authors visited three classes once a week for 5 weeks; each session was approximately 30 minutes. They conducted pre- and post-evaluations to assess student interest in computer science and mathematics, as well as anxieties related to mathematics. Survey results indicated that students' interest in computer science increased, and their confidence in their math skills increased, but their interest in math did not. Similarly, we evaluate the impact of our modules on student interest in computer science.

Taub *et al.* [86] also focus on analyzing the impact of CS Unplugged activities on participants' views toward computer science. Their pilot program consisted of 13 seventh and eighth grade students. Eighteen CS Unplugged activities were presented. The evaluation consisted of a pre-survey, with no mention of a post-survey. Six students volunteered to participate in a structured post-interview. The interview involved viewing images and discussing students' thoughts on what the images represented with respect to computer science. The authors concluded that CS Unplugged activities had a positive effect on students' views toward computer science, but that their effectiveness could be improved. One suggested improvement was to strengthen students' understanding of the connections between the activities and computer science concepts being taught. The authors also note that the activities do not adequately represent career opportunities in computer science. In addition to measuring students' attitudes toward computer science, we evaluate our modules' effectiveness in improving content understanding.

We previously presented a year-long outreach program based on CS Unplugged activities to groups of high school students [31]. The program consisted of 10 one-hour sessions repeated for two semesters in an introductory programming course. The evaluation objectives were to evaluate the impact of the program on student attitudes toward computer science, and to evaluate the impact of the program on students' self-efficacy with respect to the content being taught. Evaluation instruments consisted of pre- and post-surveys. Using a quasi-experimental, nonequivalent control group design, the surveys were used to evaluate the impact of a semester-long outreach program, repeated for two consecutive semesters. With the exception of a self-efficacy question pertaining to the definition of an algorithm –"I think I could explain what an "algorithm" is."– the evaluation indicated that the program had no statistically significant impact on student attitudes toward computer science, nor students' perceived content understanding (*i.e.*, self-efficacy). This program relied on the CS Unplugged game activities, without the use of computers or digital manipulatives. Our proposed modules teach specific computer science concepts using digital manipulatives, as well as using lectures and demonstrations.

6.1.2 Programming Concepts

Recognizing the potential of computing in the early 1960's, researchers began developing programming languages and environments designed to increase interest in computer programming. In 2005, Kelleher and Pausch [47] provided a taxonomy of available programming environments for novice programmers of all ages. Based on the intended use of each system, the environments are grouped into two categories. The first set is designed to teach programming. The second set of systems use programming to empower people to build things that are tailored to their needs. Since then, many other environments have been developed. In this section, we explore outreach and training efforts that use specialized programming languages and environments created to teach early learners basic programming skills, and to develop computational thinking skills. We discuss research efforts that use Scratch [54], a 2D drag-and-drop block style programming language, as well as Alice [17], a 3D drag-and-drop block style programming language. Although these outreach and training efforts are designed to teach computer programming skills rather than specific computer science concepts, as our modules are designed to do, we include them for completeness.

Scratch

Scratch is a 2D visual programming platform that allows the user to drag-and-drop blocks of code that snap together to form a set of instructions. The environment allows students as young as age 6 to explore computer programming through the creation of animated games or stories. Our curriculum modules focus on introducing specific computer science concepts, relying on digital manipulatives rather than focusing exclusively on programming.

Lewis [53] describes an effort designed to evaluate student attitudes toward programming, as well as their ability to interpret loops and conditional statements after using Scratch and Logo [56]. A total of fifty 10-12 year old students were enrolled in a 12-day summer enrichment program. For the first 6 days of the program, approximately half of the students were taught programming using Scratch, while the remaining students were taught using Logo. At the end of the sixth day, the groups switched languages. Although this program spanned 12 class periods, the evaluation focused only on the first 6 days of the program, comparing the effectiveness of Scratch and Logo. Since Scratch is a drag-and-drop block style programming language with a strong visual component, and Logo is text-based with no visual component, the author hypothesized that the students who learned Scratch would indicate a more positive attitude toward programming and would demonstrate an increase in self-efficacy in interpreting loops and conditional statements [53]. However, the evaluation indicated that on average, students using Logo had higher confidence in their programming abilities, but students using Scratch showed a better understanding of conditional constructs. With respect to learning loops and the likelihood of participating in further programming outreach efforts, there were no significant differences between the groups.

Maloney *et al.* [59] describe a study focused on Scratch, spanning eighteen months at "Computer Clubhouse", an after-school program located in South Central, Los Angeles. The Clubhouse provides a variety of gaming and design activities, including board games, Microsoft's Xbox games [63], an on-site recording studio, RPG Maker [30], and Bryce 5 [26]. The authors introduced a set of activities involving Scratch to encourage students to learn programming skills. Clubhouse mentors were enlisted to provide assistance to participants; however, none of the mentors had prior experience in programming. Even without substantial help, Clubhouse members were able to teach themselves how to create games, stories, illustrations, and other projects. Over eighteen months, the authors collected over 500 projects to analyze. The analysis indicated that concepts such as conditional statements, loops, and user interaction patterns were easy for Clubhouse members to learn. However, boolean logic, use of variables, and random number generation took more effort. The authors believe participants were drawn to Scratch because they could drag-and-drop blocks of code that snap together, thereby eliminating the possibility of syntax errors.

Sivilotti *et al.* [84] discuss a three-hour module piloted during a week-long workshop consisting of eighth grade girls. The module was designed to teach advanced software engineering concepts using the Scratch programming language. The concepts that were taught included specifications, refinement, and composition. The module consisted of a twenty-minute lecture and two programming labs. They first required each student to solve a series of increasingly difficult programming assignments using Scratch, based on specifications and refinement. The students were divided into groups for the second lab, each tasked with creating a game sprite that met a precise set of specifications. After a specified time, the teams compiled the sprites into one interactive Scratch project. To evaluate the effectiveness of the module, the control group was not introduced to the software engineering concepts. Instead, they were introduced only to the Scratch programming environment. Upon completion of the module, each group completed a survey used to evaluate student impressions of the module and computer science in general. Mostly, the survey responses did not indicate substantial differences across the two groups. They did indicate, however, that using Scratch was successful in engaging students in programming, as well as in teaching advanced programming topics.

Alice

Alice is a 3D programming environment designed to teach computational thinking, problem solving, and programming skills. Our curriculum modules also teach computational thinking and problem solving skills, but do not focus on programming skills. Further, our program focuses on teaching specific concepts relying on hands-on activities using digital manipulatives.

To encourage the use of 3D virtual environments in middle schools, Rodger *et al.* [75] describe a summer program piloted with a group of middle school teachers and students. During the first portion of the summer program, teachers attended a three-week workshop focused on developing curriculum modules for math, science, language arts, social studies, art, and technology, all based on Alice. The second portion of the summer program consisted of one-week summer camps for two groups of middle school students. Each day, students were provided with four hours of instruction in Alice and three hours of free time to create projects of their choice. To assess students' use of basic constructs provided by Alice, and students' use of computer science concepts while creating their projects, the authors examined copies of the student projects. The examination indicated that all of the students used built-in methods. Although they were taught the importance of comments on the first day of the workshop, only 8.5% of students used comments in their projects. Over 40% of students created their own methods, used if statements, loops, and lists, and created events.

Kelleher *et al.* [46] describe a one-day workshop that used two versions of Alice –Storytelling Alice, and a generic version of Alice that does not support storytelling– to introduce middle school girls to computer programming. To appeal to middle school girls, the authors chose a project involving the creation of an animated 3D movie. Participants consisted of 88 fifth through ninth grade girl scouts. Forty-five of the participating girls were assigned to the control group, which used generic Alice; the remaining 43 participants were assigned to the experimental group, which used Storytelling Alice. Initially, each group was given two hours and fifteen minutes to complete a tutorial, and to develop a program using their respective versions of Alice. After the initial phase of the workshop, each participant completed a programming quiz and an attitudinal survey to measure interest in programming and attitudes toward computer science as a career choice. Once the evaluation was complete, participants were given thirty minutes to work using the alternate version of Alice, and then to decide which version of Alice they preferred. Finally, participants chose one of the two programs they created to showcase to all other participants. The evaluation suggested both the control group and the experimental group were equally successful in learning programming concepts. However, participants that used Storytelling Alice were more motivated to program, than participants that used generic Alice.

6.2 Manipulatives

During the early 19th century, Johann Heinrich Pestalozzi was a pioneer of learning through "hands-on" activities. Inspired by Pestalozzi, Fredrich Froebel, known as the father of kindergarten, developed a set of "gifts" –balls, blocks, and sticks– for children to use for playing and learning [74, 92]. Manipulatives continue to be useful in today's classroom to engage kinesthetic learners, whether they are as simple as the gifts offered by Froebel, or as advanced as digital and virtual manipulatives. In Section 6.2.1, we begin by defining physical, virtual, and digital manipulatives. Next, Section 6.2.1.1 describes a sampling of outreach programs that employ the use of digital manipulatives. Lastly, Section 6.2.1.2 describes research pertaining to the use of virtual and physical manipulatives.

6.2.1 Physical, Virtual, and Digital Manipulatives

In literature, the terms virtual and digital manipulative are often mischaracterized. As an example, Bennet [1] refers to a web-based application as a digital manipulative. In this section, we present literature examples that most closely represents our understanding of the definitions of physical, virtual and digital manipulatives.

Moyer *et al.* [65] define physical (concrete) manipulatives as objects that can be touched, turned over, slid, etc. They define virtual manipulatives across two categories: static visual representations and dynamic visual representations. Static visual representations consist of images, such as those "found in books, drawings for overhead projectors, sketches on a chalkboard, etc." Static visual representations are limiting in that they can not be used in the same manner as a concrete manipulative. On the other hand, dynamic visual representations are images that can be manipulated as if they were three-dimensional objects. According to Moyer, the dynamic category should be considered a true virtual manipulative. Resnick [74] defines digital manipulatives as "computationally-enhanaced versions of traditional children's toys." Similarly, Raffle [73] states that, "digital manipulatives embed computation in familiar children's toys." Similar to Resnick and Raffle, we consider a digital manipulative to be a small embedded toy designed to engage kinesthetic learners though hands-on activities.

6.2.1.1 Digital Manipulatives

Zuckerman *et al.* [95] propose that manipulatives, both physical and digital, fall into two categories: "Froebel-inspired Manipulatives" (FiMs), which consist of materials that cultivate modeling of real-world structures, and "Montessori-inspired Manipulatives" (MiMs), which encourage modeling of abstract structures. The authors describe two types of digital MiMs, small digital devices that snap together, whack provide an engaging, interactive experience for students. The first are FlowBlocks, designed to demonstrate computer science concepts such as loops and variables, and mathematical concepts such as counting and probability. The second are SystemBlocks, designed to support simulation of dynamic behavior, including the flow of water through a bathtub, or the spread of a virus. To evaluate the FlowBlocks and SystemBlocks, the authors observed a total of 25 four to eleven-year old children interacting with the systems for approximately 40 hours. The SystemBlocks were piloted with a group of fifth grade (10-year old) and preschool (4-year old) students. The authors found that all of the 10-year old students and 60% of the 4-year old students were able to associate the directional flow of the LED's on the SystemBlocks with the flow of, say, water. The FlowBlocks were piloted with students ranging in age from 6-11 years old. Given a set of

FlowBlocks, students were observed forming loops, using the probability block to direct sequences, counting, and measuring.

Resnick *et al.* [74] discuss four digital manipulatives used in K-12 outreach. The first are LEGO Programmable Bricks (PBricks). Students learn to write programs using Logo, a programming language for LEGO PBricks. The PBricks have controllable motors and lights and are able to acquire data through sensors and communicate via infrared. The second are Programmable Beads (PB), small embedded devices that have a microprocessor and LEDs that can be strung together to form a necklace. The PBs communicate using inductive coupling. Depending on the program provided, and the order the PBs are strung on the necklace, they create various colorful light patterns. The third manipulative is BitBall, a rubber ball the size of a baseball. It is embedded with a LEGO PBrick, an accelerometer, and colored LEDs. Using infrared communication and a modified version of Logo, students can program the ball to turn on its LEDs based on readings from the accelerometer. The last manipulative is the Thinking Tag. These devices were originally developed as conference name tags that contained information about the wearer and transmitted that information to other Thinking Tags. Later, Thinking Tags were incorporated in schools as part of "participatory simulations". As an example, students can wear the Thinking Tags to simulate the spread of a virus. In the scenarios described, students were given the devices, shown how to use or program them, and left to explore and create. Our goal is to teach specific computer science concepts, while at the same time allowing students to play, explore, and ask questions.

Raffle *et al.* [72] introduce Topobo, a 3D modeling system "embedded with kinetic memory, the ability to record and playback physical motion." Topobo consists of various components that snap together to form geometric or abstract shapes. Each component is categorized as *active* or *passive*. Active components are networkable and/or motorized, each connected with small cables used to provide power and communication. To interact with the manipulatives, students begin by creating an animal shape such as a horse, a dog, or a spider. To start recording a motion, students press a button on an active component, then twist, pull, and move the shape. To stop the recording, a button is again pressed. The recorded motion then begins to replay, continuing until the recording process is restarted. Topobo was designed to help students learn topics such as balance, center of gravity, coordination, relative motion, and movement. As an extension to Topobo, Raffle *et al.* [73] introduce Backpacks, a device that can be attached to an active component of Topobo. Backpacks allow students to change the phase, amplitude, frequency, or orientation of the creation's motion recording and playback. Both Topobo and Backpacks are designed to teach concepts related to physics. Neither provide an associated curriculum module with the toy; students learn through experimentation.

Prachanronarong [71] describes Blockuits, a series of interactive toys designed to cultivate young learners' creativity, as well as introduce basic principles of circuitry. Blockuits come in three forms: small wooden blocks, plush toys shaped like monsters, and large foam blocks. Each contains embedded circuit components. Similar to Serious Toys, Blockuits provide users with a tactile experience. What differentiates our program from Blockuits is that they are not designed to teach specific concepts. Blockuits are not associated with a specific curriculum module; the approach relies on students learning through play.

Goh *et al.* [36], describe i-Cube, a cube-shaped digital manipulative with spacial awareness of neighboring cubes. The i-Cube was designed to provide educators with a generic and flexible tool to support learning through play. As an example, to demonstrate the use of i-Cube, the authors present MusiCube Arranger, an exploratory music application that allows students to create short repetitive musical sequences by simply aligning the cubes in various orders. Our program provides a curriculum module that focuses on teaching specific CS concepts, relying on a digital manipulative to increase content understanding of the topic.

Magloire *et al.* [57] present a series of workshops designed to introduce young learners to basic electronics and technology through a variety of hands-on activities. Magloire describes three activities used in the workshops. First, participants use conductive paint to create interactive electronic projects. Next, participants are introduced to LittleBits®– a group of digital manipulatives in the form of electronic modules that snap together to create electronic circuits. Lastly, participants learn to design and build 3D projects using 3D printers. This program allows students to learn various STEM concepts through active playing. Our program teaches specific CS concepts relying on digital manipulatives created specifically for the curriculum module being taught.

Beelight [83] is a digital manipulative created to improve student recognition of colors. Shen *et al.* explains the device consists of two parts, a color collector, a sharing platform. The color collector consists of a color sensor, an Arduino, and RGB LED's. The sharing platform consists of an Arduino and RGB LEDs. The Beelight has two modes of operation, color sharing and color collecting. In color sharing mode, students hold the color collector near an object, the color sensor detects and stores the color. When the color collector is paired with the color sharing platform, the platform turns the color that was collected. Students are tasked with naming the color. The color collecting mode is the opposite. The color sharing platform displays a color and students must find and collect the color that matches the platform. To appeal to young learners, the color collector was designed in the shape of a bee, while the sharing platform is honeycomb-shaped. This program provides a tool designed to improve color recognition, a concept students are familiar with. Our program teaches new concepts and provides digital manipulatives to reinforce the material taught.

In addition to the digital manipulatives described above, there is a host of proprietary digital manipulatives available for young learners, including Lego Mindstorms® [51], Makey Makey® [58], LittleBits® [55], and Arduino [2]. Each can be used to teach a range of STEM concepts through experimentation. Our program relies on learning through hands-on activities, but we focus on teaching specific CS concepts using a digital manipulative designed specifically for the concept being taught.

6.2.1.2 Virtual and Physical Manipulatives

Westenskow [89] describes meta-analysis reports concerning the effectiveness of using physical manipulatives in pre-collegiate mathematics education. Across twenty-three research programs, eleven show significant increases in student achievement when using physical manipulatives. Two programs indicated that students did not favor using physical manipulatives, and ten indicated that there were no significant differences in achievement between students who did use physical manipulatives and those who did not.

De Jong *et al.* [27] discuss the relative advantages of physical and virtual laboratories. Both physical and virtual laboratories help students develop team-work skills, cultivate interest in science, and promote conceptual understanding. de Jong notes that physical laboratories allow students to develop practical laboratory skills, such as setting up equipment, time management and planning skills, and equipment troubleshooting skills. One advantage of a virtual laboratory is that designers of an experiment can simplify learning by highlighting important information and removing confusing details. The decrease in set-up time also allows instantaneous results, enabling students to perform more experiments. Further, computer logs of student actions during virtual laboratories, allow instructors to flag concepts that need to be discussed further, as well as identify groups of students who need extra tutoring. de Jong notes that virtual laboratories show advantages with respect to conceptual skills. However, physical laboratories show advantages with respect to practical skills. Physical and virtual laboratories are individually effective, but a combination of both can offer benefits that neither can achieve individually.

Klahr *et al.* [48] describe an empirical study to explore the use of physical versus virtual materials in an engineering design project with middle school students. Students were tasked with assembling and testing various model cars, with a simple challenge to make them travel as far as possible. This study indicates that regardless of the type of materials used –virtual or physical– students showed substantial gains in content understanding and confidence. Klahr notes, however, that due to the cost, virtual materials may be the preferred instructional medium in many cases.

Olympiou *et al.* [68] describe a study on the effects of using three approaches to teaching a physics lesson involving light and color. The approaches involved the use of i) physical manipulatives only; ii) virtual manipulatives only; and iii) a combination of physical and virtual manipulatives. The results show that the combination of physical and virtual manipulatives enhances students' conceptual understanding of light and color more than the use of physical manipulatives or virtual manipulatives alone.

Similiar to Olympiou, Gire *et al.* [35] found using a combination of physical and virtual manipulatives effective when teaching the physics of pulleys. However, Olympiou found that students using the physical manipulative prior to the virtual manipulative scored higher with respect to concept understanding of effort force, distance pulled, and mechanical advantage.

The above research indicates that both physical and virtual manipulatives can be effective educational tools, individually. It has also been shown that the use of a combination of physical and virtual manipulatives often provide even greater benefit. As technology continues to advance, we are experiencing an increase in the development of digital manipulatives. *Serious Toys* are physical manipulatives that are computationally-enhanced. We consider serious toys to be a combination of physical and virtual manipulatives, offering combined benefits.

6.3 Computer Science Concepts

This section discusses programs designed to teach K-12 students i) the binary number system; ii) concepts of networks, protocols, and algorithms; and iii) sensors and sensor networks.

6.3.1 Binary

Sarkar *et al.* [80,81] describe a variety of kinesthetic activities designed to introduce students to computer organization based on the PIC microcontroller. In one activity, students are taught the basic concepts of bits, bytes, and binary numbers. They are then shown a PIC-based device that demonstrates binary numbers through the use of eight LEDs. Subsequent lessons cover computer memory, LED matrix manipulation, displays, and speech generation. Each lesson emphasizes the importance of the binary number system in computing. Evaluation results indicate that 70% of students were satisfied with the hands-on experience and would like more hands-on projects. In addition, 75% of students indicated that the PIC project gave them a better understanding of hardware fundamentals, including binary. The activities described in this work differ from ours in that student interaction involved writing a desktop program to turn on the display. Further, their work does not cover arithmetic.

Sakala *et al.* [79] also describe a software tool to teach the binary number system, with a focus on conversion from decimal to binary. The interface prompts the user to enter a number and demonstrates each step of the conversion process using division by 2, with the remainder representing the binary digit. To evaluate the interface, a lesson on binary conversion was given to a control group using a traditional lecture method; the experimental group was taught using the interface. The results indicate that the tool-based approach was more effective. The primary difference between this work and ours is that this group relies on a software tool to teach binary conversion. We rely on a hardware tool to teach students binary conversion, in addition to binary arithmetic.

Goldschmidt *et al.* [37] describe how several fundamental computer science concepts can be incorporated in the K-12 curriculum, including alternative number systems. They describe how to count in binary using rhythm in physical education. In addition, they explain how the Mayan base-20 system, the Enigma Code, and encryption/decryption can be used as discussion points in a social studies class. As with this group, we use lectures and activities to teach alternative number systems. Our activity consists of a hardware-based tool rather than group games.

Waraich [88] describes a multimedia learning environment used for lessons focused on computer architecture, including binary arithmetic and logic gates. The target audience includes first-year computer science undergraduates. Students that used the instructional environment received higher test scores than those who did not. As with the previous work, this approach uses software to teach binary and logic gates.

Chun *et al.* [20] describe a web-based system designed to help teach K-12 students and educators basic programming skills. An 8x8 LED matrix driven by serial communication through a web-based application is used to display images, as well as binary numbers. The web application provides a virtual LED display that allows students to turn LEDs on and off on the LED display. This hardware and software system can be used to teach binary number systems; however, the primary goal is to teach basic programming skills.

6.3.2 Networks, Protocols, and Algorithms

CS Unplugged [24] offers several activity modules that demonstrate the need for developing step-by-step instructions for computers to follow. For example, *Marching Orders* is an activity in which a student is given an image and instructed to give the rest of the class step-by-step instructions on how to draw the image. Similarly, *Harold the Robot* involves a volunteer playing the part of a robot, while the class is tasked with giving step-by-step instructions on how to build a structure from blocks. These activities demonstrate the difficulty of creating an algorithm for a computer to follow. *Muddy City—Minimal Spanning Trees* introduces familiar networks and some basic networking concepts. Each of these activities demonstrates either the importance of networks or the necessity to create correct step-by-step instructions for a computer to follow. However, none of these activities are reinforced with hands-on activities, as is the case with our module.

Computing Science Inside [23] also offers modules designed to teach basic networking and algorithms concepts. *Tablets of Stone* focuses on introducing students to networks and protocols by discussing examples of networks familiar to students (e.g., Internet, Postal Service, cell phone). They also discuss the importance of having a common set of rules within each network –protocols– governing communication. This program introduces a communication scenario and requires students to develop a protocol for the scenario. As an activity, students are given a set of ambiguous instructions explaining how to draw a particular object. Students are asked to carefully follow the instructions and compare drawings with other students after they have completed each step. As expected, the objects that are drawn typically vary. As an exercise, students are tasked with rewriting and testing their drawing algorithm. Although many of the concepts covered in this module are similar to ours, our approach relies on hands-on activities with an embedded device, which is engaging to kinesthetic learners, as compared to paper and pencil exercises.

6.3.3 Sensors

Saad *et al.* [39, 78] discuss a three-year project designed to integrate advanced information technologies into sixth through twelfth grade classes, with the goal of motivating students to pursue careers in STEM-related fields. The efforts included a series of workshops, summer camps, and follow-up activities for 90 participating teachers and 120 participating students, with two possible paths of study. The first was a sensor network path comprising curriculum modules that introduce students to i) electronic transducers; ii) data acquisition; iii) data transmission; iv) wireless communication; v) TCP/IP protocols; vi) database concepts; and vii) data analysis and presentation. The second path of study focused on multimedia, introducing students to i) web development tools; ii) wiki tools; iii) multimedia concepts and tools; and iv) database concepts. There was no mention of a formal evaluation of this program. The program differs from ours in that their curriculum is web-based, whereas ours is taught in the classroom and includes hands-on activities involving manipulatives.

Baker *et al.* [6] discuss a series of curriculum modules to teach middle school students engineering and technology concepts. One of the modules uses LEGO Mindstorms® NXT robots to teach students how to program the robots to use touch sensors, light sensors, and RFID sensors for various applications. In addition to applications of sensors, our module focuses on teaching students how sensors work, the construction of a basic sensor, and the process of passing data through a network of wireless sensor nodes.

Dabney *et al.* [25] report on their use of sensors –an accelerometer, gyroscope, magnetometer, and photosensor– on a Samsung Nexus S smartphone to teach Java programming during a week-long summer camp for high school students. The camp was composed of 22 students, of which six were female, one was African-American, and one was of Hispanic descent. To make the transition to programming in Java easier, during the first three days of the camp, participants created mobile apps using MIT's App Inventor [18], a web-based drag and drop platform that allows users to snap code blocks together to create a program. During the last two days of the camp, students explored the sensors on the smartphone by creating Android apps that used each of the on-board sensors.

Using pre- and post-surveys, the authors found that there was a significant increase in students' interest in studying computer science in college and developing educational apps and games. They also reported a significant increase in students' self-efficacy in regard to their understanding of conditional blocks, loops, mathematical comparisons, variables, and functions. This program focused on creating applications that use sensors, whereas our program teaches how sensors work and how to create networks of sensors.

6.4 Active Learning

The teaching approach described in this dissertation has roots in the pedagogical philosophy of *active learning* [13], which involves activities and techniques designed to ensure student engagement. The philosophy is perhaps best summarized by Martinez and Eisenhart [62] as part of a case study in pedagogical techniques for teaching Physics: "The alternative instructional methods that show promise are various activities that allow students to be active class participants, procedures or devices that give instructors quick ways to assess their students' understanding during class, in-class opportunities to discuss students' understandings and difficulties, and activities specially designed to be fun, challenging, and relevant." It is interesting to note that active learning approaches may have differential impacts for women and other underrepresented groups in computing. In a pedagogical study led by Yoder et al. [94], for example, evaluation results showed that female students tend to perform better when taught using active learning techniques.

Chapter 7

Conclusion and Future Directions

Like many computer science educators, we are looking for new ways to excite the next generation of undergraduates about pursuing a degree in computer science. Although the number of awarded bachelor's degrees in computer science has increased over the past three years, work is still needed to meet the projected growth in computer-related jobs by 2020 [15]. Structured education begins in K-12. In this dissertation, we set out to investigate a new approach to teaching CS concepts with tailored curriculum modules and embedded computing manipulatives. The approach is designed to engage visual, auditory, and kinesthetic learners through lectures, visual demonstrations, and hands-on activities. The results are expected to be broadly applicable across topic areas within computing. We focus on three in this initial exploration, selected because they are among the most important computing concepts today.

7.1 Curriculum Modules

In this section, we summarize our work on the development and evaluation of three curriculum modules focused on binary number systems; networks, protocols, and algorithms; sensors and sensor networks, respectively. We combined the three modules to form an integrated program and we also summarize our evaluation of this combined program of study.

7.1.1 The Binary Number System

Binary is fundamental to many computer science topics and is typically introduced early in the undergraduate computer science curriculum. It is often viewed as onerous, both to teach and learn. Relying on an embedded computing manipulative, or serious toy, we have developed a curriculum module to teach the binary number system in a manner that is both engaging and informative. The module spans two 60-minute class periods. The class consists of a lecture facilitated by a series of questions. Students learn to convert decimal numbers to binary numbers and vice-versa. We explain that the strategies used to convert decimal numbers to binary are applicable to other bases — e.g., decimal to base three, or base eight. The second class begins with a review, and then an introduction to the serious toy. Students are assigned several binary conversion, addition, and subtraction problems and are instructed to perform the operations on paper before validating their work using the toy. We piloted and evaluated the module with four groups of high school students. The evaluation results were largely positive across the four pilots.

7.1.2 Networks, Protocols, and Algorithms

Networks are woven throughout many aspects of our lives. We are inherently members of a network consisting of family, friends, and co-workers. Networks connect our phones, our computers, and our entertainment devices. Recognizing the importance of networks in our lives, and more importantly, within computer science, we developed a curriculum module designed to introduce young learners to basic concepts of networks, protocols, and algorithms. Similar to the Binary Number System design, the module spans two 60-minute class periods. The first session includes an engaging lecture designed to teach students to recognize various networks and their associated communication protocols. Students learn the importance of networks, protocols, and algorithms, how the concepts are related to one another, and their relevance to computer science. The second session begins with an introduction to the serious toy and a demonstration of various sensors on the toy. Next, students observe a network of these toys communicating through a serial bus. Finally, students use a set of toys and jumper wires to create a network capable of communication. We piloted this module with six classes at a local middle school, and four groups of high school students. The evaluation results were largely positive across the pilots.

7.1.3 Sensors and Sensor Networks

In modern life, sensors and sensor networks are increasingly common. Most home appliances rely on one or more sensors, and modern vehicles include hundreds of sensors. Networks of sensors are used in a variety of applications, including medical and wildlife monitoring [38, 43], military applications [49], and environmental management [29]. Motivated by the ubiquity of sensors in our daily lives, the value of sensors in computing, and building on the Networks, Protocols, and Algorithms module, we developed Sensors and Sensor Networks as our third curriculum module. The module spans two 60-minute class periods. The first begins with a lecture designed to give students a basic understanding of how sensors work. We discuss various sensors and the relevance of sensors and sensor networks in computer science. The first session concludes with an activity where students make a simple soil-moisture sensor from plaster of paris, nails, and a straw, adapted from [14]. During the second session, students test the soil-moisture sensors by connecting the sensors to a network of MoteStacks [29] and observing changes in the data readings as water is added to the surrounding soil. Students observe the changes in data through a custom website. We piloted this module with five classes at a local middle school, and four groups of high school students. The evaluation results were largely positive across the pilots.

7.2 Combined Module

Although each module is designed to enable independent adoption, we piloted the three modules as an integrated program. The combined program spans eight 60-minute class periods. The first and last involve administration of pre- and post-surveys, and pre- and post-quizzes. The second and third sessions cover the Binary Number System module. The third and fourth sessions cover the Network, Protocols, and Algorithms module. The remaining sessions cover the Sensors and Sensors Networks module. We piloted the combined program with eleven classes from two local middle schools. We also piloted the program with one class from a local high school. The evaluation results were largely positive across the pilots.

Similar to the results for the individual curriculum modules, the evaluation results for the combined module were largely positive.

This suite of *serious toys* and accompanying curriculum modules has the potential to reenergize the pedagogical landscape around fundamental topics that are often difficult to teach in a manner that is both informative and fun.

7.3 Future Directions

While there are many future directions to explore, we will concentrate on three: (i) The module topics selected for this dissertation were chosen, in part, due to their importance to CS. Given the large number of equally important CS topics not yet considered, developing additional curriculum modules with accompanying serious toys would be a natural next step. (ii) The next direction involves expanding the existing curriculum modules by developing sub-modules enhanced by existing serious toys. As an example, developing a module focused on data representation concepts and the ASCII table would be a logical progression of the Binary Number System module and accompanying serious toy. (iii) Finally, a particularly interesting study related to this work would be to conduct a comparison between the effectiveness of CS Unplugged activities and the teaching approach described in this dissertation.

Appendices

Appendix A Module Learning Objectives

	Knowledge	Comprehension	Application	Analysis	Synthesis	Evaluation
Binary Number System						
Learning Objective 1			X			
Learning Objective 2			X			
Learning Objective 3			X			
Learning Objective 4				X		
Learning Objective 5					X	
Networks, Protocols, and Algorithms						
Learning Objective 1		X				
Learning Objective 2				X		
Learning Objective 3				X		
Sensors and Sensor Networks						
Learning Objective 1			X			
Learning Objective 2					X	
Learning Objective 3		X				
Learning Objective 4				X		

Table 1: Mapping of Module Learning Objectives to Bloom's Taxonomy

Here we present Table 1, a visualization that depicts a mapping of the learning objects listed in Chapters 2, 3, and 4, with the category in Bloom's Taxonomy of Education Objectives that most closely represents the thinking skills associated with the learning objectives. In the following sections the categories of Bloom's Taxonomy of Education Objectives and each of our curriculum modules learning objectives are described.

A.1 Bloom's Taxonomy of Education Objectives

The six categories from lower-order to higher-order thinking skills, are as follows: i) knowledge, ii) comprehension, iii) application, iv) analysis, v) synthesis, and vi) evaluation. According to [69], the description of each category is as follows:

Knowledge "involves the recall of specifics and universals, the recall of methods and processes, or the recall of patterns, structure, or setting."

Comprehension "refers to a type of understanding or apprehension such that the individual knows what is being communicated and can make use of the material or idea being communicated without necessarily relating it to other material or seeing its fullest implication."

Application refers to the "use of abstractions in particular and concrete situations."

Analysis represents the "breakdown of a communication into its constituent elements or parts such that the relative hierarchy of ideas is made clear and/or the relations between ideas expressed are made explicit."

Synthesis involves the "putting together of elements and parts so as to form a whole."

Evaluation engenders "judgments about the value of material and methods for given purposes.

A.2 Binary Number System Module

The learning objectives call for students to:

i learn to convert decimal numbers to binary numbers and vice-versa;

ii learn to add two binary numbers of arbitrary length;

iii learn to subtracttwo binary numbers of arbitrary length;

iv understand the relevance of binary numbers to computer science; and

v earn to apply the concepts learned to number systems of arbitrary bases.

A.3 Networks, Protocols, and Algorithms

The learning objectives call for students to:

i understand the definitions of network, protocol, and algorithm;

ii understand how networks, protocols, and algorithms are related; and

iii understand the relevance of networks, protocols, and algorithms in computer science.

A.4 Sensors and Sensor Networks

The learning objectives call for students to:

i have a basic understanding of how sensors work;

ii understand the construction of a basic sensor;

iii be familiar with various types of sensors; and

iv understand the relevance of sensors and sensor networks in computer science.

Bibliography

[1] Jr. A.B Bennett and L.T. Nelson. Mathematics for Ealimentary Teachers, 6th edition, Oline Learning Center. http://highered.mheducation.com/sites/0072532947/student_view0/digital_manipulative_kit.html.

[2] Arduino. Arduino. http://www.arduino.cc.

[3] J. Arlegui et al. A PBL Approach Using Virtual and Real Robots (with BYOB and LEGO NXT) to Teaching Learning Key Competences and Standard Curricula in Primary Level. In *Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality*, TEEM '13, pages 323–328, New York, NY, USA, 2013. ACM.

[4] Atmel Corporation. Atmega168. http://www.atmel.com/devices/atmega168.aspx.

[5] Atmel Corporation. Atmega168. http://www.atmel.com/devices/atmega8515.aspx.

[6] J. Baker et al. Application of Wireless Technology in K-12 STEM Outreach Programs in Middle Schools. In *Antennas and Propagation Society International Symposium APSURSI, 2010 IEEE*, pages 1–4, 2010.

[7] V. Barr and C. Stephenson. Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*, 2(1):48–54, February 2011.

[8] D. Basu et al. Wireless sensor network based smart home: Sensor selection, deployment and monitoring. In *Sensors Applications Symposium (SAS), IEEE*, pages 49–54, Feb 2013.

[9] A. Bdeir. Electronics As Material: LittleBits. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, pages 397–400, New York, NY, USA, 2009. ACM.

[10] L. Benotti et al. Engaging High School Students Using Chatbots. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, pages 63–68, New York, NY, USA, 2014. ACM.

[11] B.S. Bloom et al. *Taxonomy of Educational Objectives, the classification of education goals Handbook I: Cognitive Domain*. McKay.

[12] GoGo Board. Knowing how sesors work on the GoGo Board. http://www.gogoboard.org/node/12.

[13] C.C. Bonwell and J.A. Eison. Achive Learning: Cresting Eexcitement in the Classroom. *ASHE-ERIC Higher Education Report No.1*, 1(1):4–121, 1991.

[14] G. Branwyn. No-cost soil moisture sensor. makezine.com/2009/03/04/nocost-soil-moisture-sensor/.

[15] Bureau of Labor Statistics, U.S. Department of Labor. Occupational Outlook Handbook, 2010-11 Edition, Computer Scientists. www.bls.gov/oco/ocos304.htm.

[16] Gail Carmichael. Girls, Computer Science, and Games. *SIGCSE Bull.*, 40(4):107–110, November 2008.

[17] Carnegie Mellon University. Alice. http://alice.org/.

[18] MIT Center for Mobile Learning. App inventor. http://appinventor.mit.edu/, 2012. online.

[19] Langston Charter Middle School. Mission Statement. `http://www.langstoncharter.org/about-langston/mission-statement.html`.

[20] S. Chun and J. Ryoo. Development and Application of a Web-based Programming Learning System with LED Display Kits. In *Proceedings of the 41^{st} ACM Technical Symposium on Computer Science Education*, SIGCSE '10, pages 310–314, New York, NY, USA, March 2010. ACM.

[21] Clemson University Emerging Scholars. http://www.clemson.edu/academics/programs/emerging-scholars/.

[22] U.S. Congress Joint Economic Commitee. *STEM Education: Preparing for the Jobs of the Future*. Government Printing Office, 2012.

[23] CS Inside. http://csi.dcs.gla.ac.uk/index.php.

[24] CS Unplugged. http://www.csunplugged.org.

[25] M.H. Dabney et al. No Sensor Left Behind: Enriching Computing Education with Mobile Devices. In *Proceeding of the 44th ACM technical symposium on Computer science education*, SIGCSE '13, pages 627–632, New York, NY, USA, 2013. ACM.

[26] Daz3D. Bryce5. www.daz3d.com.

[27] T. de Jong et al. Physical and Virtual Laboratories in Science and Engineering Eduration. *Science*, 340:305 – 308, 2013.

[28] P. Doerschuk et al. Pilot summer camps in computing for middle school girls: from organization through assessment. In *Proceedings of*, ITiCSE '07, pages 4–8, New York, NY, USA, 2007. ACM.

[29] G. W. Eidson et al. The South Carolina Digital Watershed: End to End Support for Real Time Management of Water Resources. *IJDSN*, 2010, 2010.

[30] Inc Enterbrain. RPG Maker. www.rpgmakerweb.com.

[31] Y. Feaster et al. Teaching CS Unplugged in the High School (with limited success). In *Proceedings of*, ITiCSE '11, pages 248–252, New York, NY, USA, June 2011. ACM.

[32] Cybertelecom Federal Internet Law and Policy An Educational Project. Postal Service. http://www.cybertelecom.org/notes/usps.htm, 2013.

[33] E.P. Freitas et al. Supporting platform for heterogeneous sensor network operation based on unmanned vehicles systems and wireless sensor nodes. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 786–791, 2009.

[34] R. Garlikov. The Socratic Method Teaching by Asking Instead of by Telling. http://www.garlikov.com.

[35] E. Gire et al. The Effects of Physical and Virtual Manipulatives on Students' Conceptual Learning About Pulleys. In *Proceedings of the 9th International Conference of the Learning Sciences - Volume 1*, ICLS '10, pages 937–943. International Society of the Learning Sciences, 2010.

[36] W.B. Goh et al. The i-Cube: Design Considerations for Block-based Digital Manipulatives and Their Applications. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, pages 398–407, New York, NY, USA, 2012. ACM.

[37] D. Goldschmidt et al. An Interdisciplinary Approach to Injecting Computer Science into the K-12 Classroom. *Journal of Computing Sciences in Colleges*, 26(6):78–85, 2011.

[38] R.N. Handcock et al. Monitoring Animal Behaviour and Environmental Interactions Using Wireless Sensor Networks, GPS Collars and Satellite Remote Sensing. *Sensors*, 9(5):3586–3603, 2009.

[39] L. He et al. Information Technology Education for K-12 Students and Teachers: From Sensor Network to Comprehensive and Customized Web Interaction. In *Proceedings of the 9th ACM SIGITE conference on Information technology education*, SIGITE '08, pages 65–70, New York, NY, USA, 2008. ACM.

[40] D.J. Hernandez et al. Double Jeopardy How Third-Grade Reading Skills and Poverty Influence High School Graduation. www.aecf.org.

[41] Hirose Connectors. Hirose connectors. `http://www.hirose.co.jp/cataloge_hp/e54000041.pdf/`.

[42] N. Hollinworth et al. Making Electronics More Accessible to People with Learning Disabilities. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 1255–1260, New York, NY, USA, 2014. ACM.

[43] N.F. Ince. Integration of Wearable Wireless Sensors and Non-Intrusive Wireless In-Home Monitoring System to Collect and Label the Data from Activities of Daily Living. In *3rd IEEE/EMBS International Summer School on Medical Devices and Biosensors*, pages 28–31, September 2006.

[44] Digi International. XBee®. `www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module#overview`.

[45] Josheph Hollingsworth. Joe Hollingsworth. http://homepages.ius.edu/JHOLLY/.

[46] C. Kelleher et al. Storytelling Alice Motivates Middle School Girls to Learn Computer Programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 1455–1464, New York, NY, USA, 2007. ACM.

[47] C. Kelleher and R. Pausch. Lowering The Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. *ACM Comput. Surv.*, 37(2):83–137, June 2005.

[48] D. Klahr et al. Hands on what? The relative effectiveness of physical versus virtual materials in an engineering design project by middle school children. *Journal of Research in Science Teaching*, 44(1):183–203, 2007.

[49] K.A. Kumar. IMCC protocol in heterogeneous wireless sensor network for high quality data transmission in military applications. In *1st International Conference on Parallel Distributed and Grid Computing (PDGC)*, pages 339–343, October 2010.

[50] L. Lambert and H. Guiffre. Computer Science Outreach in an Elementary School. *Journal of Computing Sciences in Colleges*, 24(3):118–124, 2009.

[51] Lego. Lego education. http://education.lego.com/en-us.

[52] LessEMF. LessEMF Conductive Plastic and Glass EMF Shielding Devices and Materials. http://www.lessemf.com/plastic.html.

[53] C.M. Lewis. How Programming Environment Shapes Perception, Learning and Goals: Logo vs. Scratch. In *Proceedings of the 41^{st} ACM Technical Symposium on Computer Science Education*, SIGCSE '10, pages 346–350, New York, NY, USA, March 2010. ACM.

[54] MIT Media Lab Lifelong Kindergarten Group. Scratch. http://scratch.mit.edu/.

[55] LittleBits. Littlebits electronics. http://littlebits.cc/?gclid=CKTe8qaytMECFSEV7AodHhEAtw.

[56] Logo. BFOIT Introducation to Computer Programming. http://www.bfoit.org/itp/install.html. date of last access.

[57] K. Magloire and N. Aly. Scitech Kids Electronic Arts: Using STEAM to Engage Children all Ages and Gender. In *Integrated STEM Education Conference (ISEC), 2013 IEEE*, pages 1–4, March 2013.

[58] Makey Makey. A beginners kit for everyone. http://www.makeymakey.com/.

[59] J.H. Maloney et al. Programming by choice: urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, SIGCSE '08, pages 367–371, New York, NY, USA, 2008. ACM.

[60] C. Mano et al. Effective in-class activities for middle school outreach programs. In *Frontiers in Education Conference (FIE), '10 IEEE*, pages F2E–1 –F2E–6, oct. 2010.

[61] J. Martin. Enhancing Computer and IT Skills of Rural South Carolina High School Students. http://www.cs.clemson.edu/ jmarty/scholars/scholars.html.

[62] K. Martinez and M. Eisenhart. Literature Review of Best Parctics in College Pphysics and Best Practices for Women in College Phusics. http://advance.colorado.edu/research_best_practices.html.

[63] Inc. Microsoft. XBOX. www.xbox.com/en-US.

[64] Gettys Middle School. 2011 report card. http://gms.pickens.k12.sc.us/.

[65] P.S. Moyer et al. What Are Virtual Manipulatives. *Teaching Children Ma thematics*, pages 372 – 377, 2002.

[66] National Council of Teachers of Mathematics. Principles and Standards for School Mathematics. http://www.nctm.org.

[67] Institute of Computational Ecology. Intelligent River®Research Enterprise. www.clemson.edu/public/ecology/program_irre.html.

[68] G. Olympiou and Z.C Zacharia. Blending Physical and Virtual Manipulatives: An Effort to Improve Students' Conceptual Understanding Through Science Laboratory Experimentation. *Science Education*, 96(1):21–47, 2012.

[69] Patricia Armstrong. Center for Teaching, Bloom's Taxonomy. http://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/.

[70] A.M. Paul. Why Third Grade Is So Important: The Matthew Effect. ideas.time.com/2012/09/26/why-third-grade-is-so-important-the-matthew-effect/.

[71] P. Prachanronarong. Blockuits: Innovating Building Blocks. In *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, pages 519–522, New York, NY, USA, 2013. ACM.

[72] H. Raffle et al. Topobo: A Constructive Assembly System with Kinetic Memory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 647–654, New York, NY, USA, 2004. ACM.

[73] H. Raffle et al. Beyond Record and Play: Backpacks: Tangible Modulators for Kinetic Behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 681–690, New York, NY, USA, 2006. ACM.

[74] M. Resnick et al. Digital manipulatives: New toys to think with. In *CHI*, pages 281–287, 1998.

[75] S. Rodger et al. Engaging Middle School Teachers and Students with Alice in a Diverse Set of Subjects. In *Proceedings of the* 40^{th} *ACM Technical Symposium on Computer Science Education*, pages 271–275, New York, NY, USA, March 2009. ACM.

[76] S.H. Rodger et al. Weaving Computing into All Middle School Disciplines. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, pages 207–212, New York, NY, USA, 2014. ACM.

[77] G. Rusak and D. Lim. Come Code with Codester: An Educational APP That Teaches Computer Science to K-6 Students. *J. Comput. Sci. Coll.*, 29(6):135–143, June 2014.

[78] A. Saad et al. Ossabest: A Comprehensive ITEST Project for Middle and High School Teachers and Students. In *Proceedings of the 9th ACM SIGITE conference on Information technology education*, SIGITE '08, pages 71–76, New York, NY, USA, 2008. ACM.

[79] L.C. Sakala et al. The Use of Expert Systems Has Improved Students Learning in Zimbabwe. *Journal of Sustainable Development in Africa*, 12(3):1–13, 2010.

[80] N.I. Sarka and T.M. Craig. Teaching Computer Hardware and Organisation Using PIC-Based Projects. *International Journal of Electrical Engineering Education*, 43(2):150–312, 2006.

[81] N.I. Sarka and T.M. Craig. A Low-Cost PIC Unit for Teaching Computer Hardware Fundamentals to Undergraduates. *Inroads - The SIGCSE Bulletin*, 39(2):88–91, 2007.

[82] D. Shaw and Beginner's Mind Collective. Makey Makey: Improvising Tangible and Nature-based User Interfaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, pages 367–370, New York, NY, USA, 2012. ACM.

[83] Y. Shen et al. BeeLight: Helping Children Discover Colors. In *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, pages 301–304, New York, NY, USA, 2013. ACM.

[84] P.A.G. Sivilotti and S.A. Laugel. Scratching the Surface of Advanced Topics in Software Engineering: A Workshop Module for Middle School Students. In *Proceedings of the* 39^{th} *ACM Technical Symposium on Computer Science Education*, SIGCSE '08, New York, NY, USA, March 2008. ACM.

[85] E. Sun and S. Han. Fun with Bananas: Novel Inputs on Enjoyment and Task Performance. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 1275–1280, New York, NY, USA, 2013. ACM.

[86] R. Taub et al. The effect of CS Unplugged on middle-school students' view of CS. In *Proceedings of the 14th ACM Annual Conference on Innovation and Technology in Computer Science Education*, pages 99–103, New York, NY, USA, July 2009. ACM.

[87] The CSTA Standards Task Force. CSTA K-12 Computer Science Standards. http://www.csta.org.

[88] A. Waraich. Using Narrative as a Motivating Device to Teach Binary Arithmetic and Logic Gates. In *Proceedings of*, ITiCSE '04, pages 97–101, New York, NY, USA, June 2004. ACM.

[89] A. Westenskow. Comparing the Use of Virtual Manipulatives and Physical Manipulatives in Equivalent Fraction Intervention Instruction. In *Proceedings of the 11th International Conference of Transformations and Paradigm Shifts in Mathematics Education*. Quality Content of Saxony, 2011.

[90] WikiCFP. Sensor Nnetworks. www.wikicfp.com.

[91] C. Wilson et al. Running on Eempty: The Fallure to Teach k-12 Computer Science in the Ddigital Age. www.acm.org.

[92] S. Wilson. The Gifts of Friedrich Froebel. *Journal of the Society of Architectural Historians*, 26(4):238–241, 1967.

[93] Jeannette M. Wing. Computational thinking. *Commun. ACM*, 49(3):33–35, March 2006.

[94] J.D. Yoder and C.M. Hochevar. Encouraging Achive Learning Can Improve Students' Performance on Examinations. *Teaching of Psychology*, 32(2):91–95, 2005.

[95] O Zuckerman et al. Extending Tangible Interfaces for Education: Digital Montessori-Inspired Manipulatives. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 859–868, New York, NY, USA, 2005. ACM.

[96] Iris Zur Bargury. A new curriculum for junior-high in computer science. In *Proceedings of*, ITiCSE '12, pages 204–208, New York, NY, USA, 2012. ACM.

[97] S. Zweben. Computing Degree and Enrollment Trends from the 2011-2012 CRA Taulbee Survey. www.cra.org/uploads/documents/resources/taulbee/CRATaulbeeCSDegreesandEnrollment2011-12.pdf, 2010.