

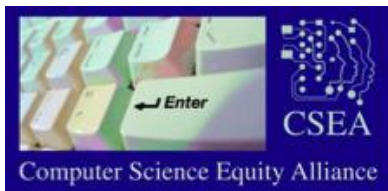


Joanna Goode
University of Oregon

Gail Chapman
University of California, Los Angeles

Sponsors & Supporters

This curriculum was created under the auspices of the Broadening the Participation in Computing National Science Foundation grant, "Into the Loop: An University K-12 Alliance to Increase and Enhance the Computer Science Learning Opportunities for African-American, Latino/a, and Female Students in the Second Largest School District in the Country". Principal Investigator: Jane Margolis (UCLA); Co-Principal Investigators Joanna Goode (University of Oregon), Todd Ullah (LAUSD), Deborah Estrin (UCLA). The Computing and Data Analysis Unit was created under the auspices of the National Science Foundation Math/Science Partnership grant, "MOBILIZE: Mobilizing for Innovative Computer Science Teaching and Learning." Co-principal Investigators: Deborah Estrin (UCLA, CENS), Mark Hansen (UCLA, CENS), Joanna Goode (University of Oregon, College of Education), Jane Margolis (UCLA, Center X), Thomas Philip (UCLA, Center X), Jody Priselac (UCLA, Center X), and Todd Ullah (LAUSD).



Acknowledgments

George Benainous, David Bernier, Robb Cutler, Judy Hromcik, Michelle Hutton, John Landa, Clifford Lee, Cueponcaxochitl Moreno, Jean Ryoo, Suzanne Schaefer, Chris Stephenson, Diane Watkins

For additional information related to the Exploring Computer Science Partnership visit: www.exploringcs.org

CONTENTS

Course Overview	5
Goals	5
Standards	5
Hardware.....	5
Software	5
Prerequisites.....	5
The Instructional Philosophy of <i>Exploring Computer Science</i>.....	6
Introduction to Curricular Approach	6
Concrete Instructional Strategies	10
Assessment.....	11
Overview of the Instructional Materials	12
Unifying Themes and Practices	13
Scope and Sequence.....	14
Overview Chart.....	16
Topic Descriptions and Objectives	20
Unit 1: Human Computer Interaction (4 weeks)	20
Unit 2: Problem Solving (4 weeks).....	21
Unit 3: Web Design (5 weeks)	22
Unit 4: Introduction to Programming (6 weeks)	23
Unit 5: Computing and Data Analysis (6 weeks).....	24
Unit 6: Robotics (7 weeks).....	25
Unit 1: Human Computer Interaction.....	27
Introduction.....	28
Daily Overview Chart	29
Daily Lesson Plans.....	30
Unit 2: Problem Solving.....	74
Introduction.....	75
Daily Overview Chart	76
Daily Lesson Plans.....	77
Final Project.....	100

Unit 3: Web Design	102
Introduction.....	103
Daily Overview Chart	104
Daily Lesson Plans.....	105
Final Project.....	125
Flash Animation Supplement	128
Javascript Supplement	133
Unit 4: Introduction to Programming	135
Introduction.....	136
Daily Overview Chart	137
Daily Lesson Plans.....	138
Final Project.....	189
Unit 5: Computing and Data Analysis	193
Introduction.....	194
Daily Overview Chart	195
Daily Lesson Plans.....	196
Final Project.....	251
Unit 6: Robotics	253
Introduction.....	254
Daily Overview Chart	255
Daily Lesson Plans.....	256
Final Project.....	290

Course Overview

Goals

Exploring Computer Science is designed to introduce students to the breadth of the field of computer science through an exploration of engaging and accessible topics. Rather than focusing the entire course on learning particular software tools or programming languages, the course is designed to focus the conceptual ideas of computing and help students understand why certain tools or languages might be utilized to solve particular problems. The goal of *Exploring Computer Science* is to develop in students the computational thinking practices of algorithm development, problem solving and programming within the context of problems that are relevant to the lives of today's students. Students will also be introduced to topics such as interface design, limits of computers and societal and ethical issues.

This curriculum has been developed for a culturally, linguistically, and socially diverse group of students in Los Angeles Unified School District. District-wide, student ethnicities include .3% American Indian, 3.7% Asian, .4% Pacific Islander, 2.3% Filipino, 73.0% Latino, 10.9% African American, 8.8% White, and .6% Other or multiple responses. Over 38% of students are English-language learners, with most English language learners' students speaking Spanish as their primary language. Furthermore, 74% of students qualify for free or reduced lunches.

Standards

The standards used for the *Exploring Computer Science* curriculum are based on the topics and goals outlined in *A Model Curriculum for K-12 Computer Science* (2006) developed by the ACM K-12 task force curriculum committee. Most of the objectives in the course align with the Level III course, *Computer Science as Analysis and Design*, while some objectives are necessarily aligned with the Level II course, *Computer Science in the Modern World*, in order to provide appropriate background knowledge for the more advanced topics.

Hardware

An ideal laboratory environment for this course would include one computer for each student in the class. These computers can be either Macintosh or PC depending on availability. A networked system would make installation of software easier for the teacher.

Software

Each computer in the classroom should have a web browser installed that allows students to perform searches and make use of a variety of websites and internet tools. Teachers will need to download and install the Scratch programming language available at <http://www.scratch.mit.edu>. All website url's were updated at the time of this writing, but may change over time.

Prerequisites

This course will be considered a college preparatory elective for California students. It is recommended that students have completed an Algebra course prior to enrolling. Thus, the course should provide a rigorous, but accessible, introduction to computer science. No previous computer science course is required to take this course.

The Instructional Philosophy of *Exploring Computer Science*

Introduction to Curricular Approach

Exploring Computer Science teaches the creative, collaborative, interdisciplinary, and problem-solving nature of computing with instructional materials which feature an inquiry-based approach to learning and teaching. As part of this curriculum, students will delve into real-world computing problems that are culturally-relevant and address social and ethical issues while delivering foundational computer science knowledge to students. Students will engage in several in-depth projects to demonstrate the real-world applications of computing.

This curriculum builds off of learning theories that view learning as a social and cultural process that does not only occur in a vacuum at school; that is, students bring to school bodies of knowledge from their lives, culture, and communities. Building from students' prior knowledge, the collection of problem solving skills, everyday "algorithmic thinking", and social and ethical knowledge of computer-related problems will result in a more student-centered curriculum. Each unit connects students' informal knowledge, technology skills, and beliefs about computing to the theoretical and foundational tenets of computer science. Students will become members of a "computing community of practice" in the classroom where they will be introduced to the behavior, language, and skills of computer scientists. Furthermore, the interdisciplinary nature of computing allows for the incorporation of subject-matter topics across disciplines into the computing curriculum.

The Nine Principles of Learning from the Institute for Learning provide the theoretical foundation of research-based instructional practices that provide the foundation for the Secondary Redesign Comprehensive Plan developed by the Los Angeles Unified School District. These nine principles underscore the beliefs of the Los Angeles Unified School District; they are integrated throughout and explain the pedagogy used in the course.

1. Organizing for Effort

An effort-based school replaces the assumption that aptitude determines what and how much students learn with the assumption that sustained and directed effort can yield high achievement for all students. Everything is organized to evoke and support this effort, to send the message that effort is expected and that tough problems yield to sustained work. High minimum standards are set and assessments are geared to the standards. All students are taught a rigorous curriculum aligned to the standards, along with as much time and expert instruction as they need to meet or exceed expectations. This principle is one of the guiding beliefs common in every school in the Los Angeles Unified School District.

2. Clear Expectations

If we expect all students to achieve at high levels, then we need to define explicitly what we expect students to learn. These expectations need to be communicated clearly in ways that get them "into the heads" of school professionals, parents, school communities and, above all, students themselves. Descriptive criteria and models of work that meets standards should be publicly displayed, and students should refer to these displays to help them analyze and discuss their work. With visible accomplishment

targets to aim toward at each stage of learning, students can participate in evaluating their own work and setting goals for their own efforts.

3. Fair and Credible Evaluations

If we expect students to put forth sustained effort over time, we need to use assessments that students find fair, and that parents, community, and employers find credible. Fair evaluations are ones that students can prepare for: therefore, tests, exams and classroom assessments as well as the curriculum must be aligned to the standards. Fair assessment also means grading against absolute standards rather than on a curve, so students clearly see the results of their learning efforts. Assessments that meet these criteria provide parents, colleges, and employers with credible evaluations of what individual students know and can do.

4. Recognition of Accomplishment

If we expect students to put forth and sustain high levels of effort, we need to motivate them by regularly recognizing their accomplishments. Clear recognition of authentic accomplishment is the hallmark of an effort-based school. This recognition can take the form of celebrations of work that meets standards or intermediate progress benchmarks en route to the standards. Progress points should be articulated so that, regardless of entering performance level, every student can meet real accomplishment criteria often enough to be recognized frequently. Recognition of accomplishment can be tied to an opportunity to participate in events that matter to students and their families. Student accomplishment is also recognized when student performance on standards-based assessments is related to opportunities at work and in higher education.

5. Academic Rigor in a Thinking Curriculum

Thinking and problem solving will be the "new basics" of the 21st century, but the common idea that we can teach thinking without a solid foundation of knowledge must be abandoned, so must the idea that we can teach knowledge without engaging students in thinking. Knowledge and thinking are intimately joined. This implies a curriculum organized around major concepts that students are expected to know deeply. Teaching must engage students in active reasoning about these concepts. In every subject, at every grade level, instruction and learning must include commitment to a knowledge core, high thinking demand, and active use of knowledge.

6. Accountable Talk

Talking with others about ideas and work is fundamental to learning but not all talk sustains learning. For classroom talk to promote learning it must be accountable to the learning community, to accurate and appropriate knowledge, and to rigorous thinking. Accountable talk seriously responds to and further develops what others in the group have said. It puts forth and demands knowledge that is accurate and relevant to the issue under discussion. Accountable talk uses evidence appropriate to the discipline (e.g., proofs in mathematics, data from investigations in science, textual details in literature, and documentary sources in history) and follows established norms of good reasoning. Teachers should intentionally create the norms and skills of accountable talk in their classrooms.

7. Socializing Intelligence

Intelligence is much more than an innate ability to think quickly and stockpile bits of knowledge. Intelligence is a set of problem solving and reasoning capabilities along with the habits of mind that lead one to use those capabilities regularly. Intelligence is equally a set of beliefs about one's right and obligation to understand and make sense of the world, and one's capacity to figure things out over time. Intelligent habits of mind are learned through the daily expectations placed on the learner by calling on students to use the skills of intelligent thinking, and by holding them responsible for doing so, educators can "teach" intelligence. This is what teachers normally do with students from whom they expect achievement; it should be standard practice with all students.

8. Self-management of Learning

If students are going to be responsible for the quality of their thinking and learning, they need to develop and regularly use an array of self-monitoring and self-management strategies. These meta-cognitive skills include noticing when one doesn't understand something and taking steps to remedy the situation, as well as formulating questions and inquiries that let one explore deep levels of meaning. Students also manage their own learning by evaluating the feedback they get from others; bringing their background knowledge to bear on new learning; anticipating learning difficulties and apportioning their time accordingly and judging their progress toward a learning goal. These are strategies that good learners use spontaneously and that all students can learn through appropriate instruction and socialization. Learning environments should be designed to model and encourage the regular use of self-management strategies.

9. Learning as Apprenticeship

For many centuries most people learned by working alongside an expert who modeled skilled practice and guided novices as they created authentic products or performances for interested and critical audiences. This kind of apprenticeship allowed learners to acquire complex interdisciplinary knowledge, practical abilities, and appropriate forms of social behavior. Much of the power of apprenticeship learning can be brought into schooling by organizing learning environments so that complex thinking is modeled and analyzed, and by providing mentoring and coaching as students undertake extended projects and develop presentations of finished work, both in and beyond the classroom.

The units in *Exploring Computer Science* contain individual lessons that taken together as a unit fit the construct for inquiry-based learning outlined in the following chart adapted from the "5 E Model".

The Inquiry-Based Learning Cycle

(Adapted from the 5 E Model", R. Bybee)

Stage of Inquiry in an Inquiry-Based Science Program	Possible Student Behavior	Possible Teacher Strategy
Engage	Asks questions such as, Why did this happen? What do I already know about this? What can I find out about this? How can I solve this problem? Shows interest in the topic.	Creates interest. Generates curiosity. Raises questions and problems. Elicits responses that uncover student knowledge about the concept/topic.
Explore	Thinks creatively within the limits of the activity. Tests predictions and hypotheses. Forms new predictions and hypotheses. Tries alternatives to solve a problem and discusses them with others. Records observations and ideas. Suspends judgment. Tests ideas.	Encourages students to work together without direct instruction from the teacher. Observes and listens to students as they interact. Asks probing questions to redirect students' investigations when necessary. Provides time for students to puzzle through problems. Acts as a consultant for students.
Explain	Explains their thinking, ideas and possible solutions or answers to other students. Listens critically to other students' explanations. Questions other students' explanations. Listens to and tries to comprehend explanations offered by the teacher. Refers to previous activities. Uses recorded data in explanations.	Encourages students to explain concepts and definitions in their own words. Asks for justification (evidence) and clarification from students. Formally provides definitions, explanations, and new vocabulary. Uses students' previous experiences as the basis for explaining concepts.
Elaborate	Applies scientific concepts, labels, definitions, explanations, and skills in new, but similar situations. Uses previous information to ask questions, propose solutions, make decisions, design experiments. Draws reasonable conclusions from evidence. Records observations and explanations	Expects students to use vocabulary, definitions, and explanations provided previously in new context. Encourages students to apply the concepts and skills in new situations. Reminds students of alternative explanations. Refers students to alternative explanations.
	Checks for understanding among peers. Answers open-ended questions by using observations, evidence, and previously	Refers students to existing data and evidence and asks, What do you know? Why do you think...? Observes

Stage of Inquiry in an Inquiry-Based Science Program	Possible Student Behavior	Possible Teacher Strategy
Evaluate	accepted explanations. Demonstrates an understanding or knowledge of the concept or skill. Evaluates his or her own progress and knowledge. Asks related questions that would encourage future investigations.	students as they apply new concepts and skills. Assesses students' knowledge and/or skills. Looks for evidence that students have changed their thinking. Allows students to assess their learning and group process skills. Asks open-ended questions such as, Why do you think...? What evidence do you have? What do you know about the problem? How would you answer the question?

Concrete Instructional Strategies

There are several concrete instructional strategies that are included in each unit to implement this culturally relevant, student-centered, and inquiry-based vision.

- Each unit begins with a description of the topic, an explanation of the importance of this topic, possible social applications of this topic, and objectives/standards for the unit.
- Whenever possible, units begin with kinesthetic activity to get students involved in the unit topic. Students are more engaged when they go beyond seatwork to gain familiarity with the scope of a topic. Acting out computing concepts is one way to have students actively engaged in the curriculum.
- Whenever possible, units present the final unit project at the beginning of the unit so students understand what type of project they will engage in at the end of the unit. Daily assignments help scaffold their knowledge towards gaining the knowledge needed to complete a particular project. The final project represents a culmination of their new knowledge and provides an opportunity to expand their understandings to a particular socially-relevant problem.
- Computing terms and definitions are explicit and part of the instruction. The curriculum avoids unnecessary jargon which might distract from learning of the critical content. Students have opportunities to use writing to reinforce the literacy component behind these computing terms and definitions.
- Foundational computing topics are connected to the 'pop-technology' students have likely encountered: cellular phones, iPods, MySpace / Facebook, blogs, Internet browsing, etc.
- Whenever possible, real-world problems are presented in the context of socially-relevant issues impacting urban communities (housing, safety, poverty, health care, access to equal rights, educational opportunities, improving social services, translation services, transportation, etc.)

- Students have opportunities to work on problems that they help define and can individualize—i.e. selecting their own content for Web sites; creating original, not pre-scripted, problem-solving strategies, etc.
- Activities are designed to encourage students to work in a variety of collaborative settings: peer-programming, group research projects, etc. which encourage conversations around computing topics.
- Students will experience a variety of ways to communicate their answers—academic writing, writing a letter to a friend or companion, using presentation software, developing graphics or animation, listing algorithms, drawing illustrations, oral presentations, etc.
- Units incorporate examples of careers in computing as they arise in the curriculum. Students will be given hypothetical opportunities to act as a professional to take on the behavior and skills to solve a given problem.
- Although using technology is a core component of this curriculum, using computers is not necessarily embedded in the curriculum on a daily basis.

All of these strategies contribute to developing the problem-solving skills and algorithmic thinking processes that are emphasized throughout the course.

It is important to note that each unit focuses on different instructional strategies; this is purposeful. In some cases, it is because the particular subject matter lends itself more successfully to a particular set of strategies, but this was also done to highlight the wide variety of possible strategies that can be used effectively in teaching this course. We encourage teachers to experiment by trying strategies that work well for them in a variety of different places in the curriculum. Journal responses and blog entries can be used by students to communicate about their work in any of the units. Peer reviews, gallery walks, jigsawing, role plays and collaborative groups of varying sizes can be used for activities throughout the course. There are many other possibilities to consider.

Assessment

With the exception of the final projects, there are no specific assessments listed in the lesson plans. There are also very few specific “homework” assignments. Differences in grading policies, types of assessments required, and student schedules make it difficult to gauge the best combination of assessment tools to use in a particular environment. Teachers are encouraged to determine which class activities might lend themselves to some research outside of class and which might make useful assessments. Additional assessment instruments can be developed by individual teachers or teacher teams. All forms of assessment should meet the criteria outlined in the Nine Principles of Learning.

Overview of the Instructional Materials

The pages that follow contain the core of the materials teachers will need in order to plan and deliver *Exploring Computer Science*. The materials begin with the unifying themes and practices that are woven throughout the course followed by a Scope and Sequence chart that details the various topics included in the course, along with the unit in the course where each is introduced and reinforced. Teachers should continue to refer back to previous units where appropriate. For example, Unit 3 builds on many of the Unit 1 concepts by taking students from discussing and viewing websites to actually using and developing them. The approximate time allotment noted in the chart includes all activities from introduction through application.

Following the Scope and Sequence is an overview of each unit that includes the unit description and overall objectives of the unit. There is also a table that indicates the topics for each instructional day of the course.

Finally, are the daily lesson plans with detailed student activities and teaching strategies for each day. Each lesson has been built on a 55 minute class period. In schools where class periods are shorter or longer (or on varying block schedules) adjustments will need to be made; such adjustments may include combining lessons (for longer class periods) or assigning parts of the lesson for homework (for shorter class periods).

An attempt was made to provide enough detail to the teaching strategies sections to give teachers clear guidance as to the activities involved and the types of questions that might need to be asked to prompt discussion. At the same time, an effort was made not to be prescriptive.

Each unit includes supplementary materials, a final project, and a sample rubric for the final project.

Unifying Themes and Practices

The individual lessons in this course were developed to reinforce the unifying themes and support the use of the computational practices that we expect students to employ.

The three themes are

- The creative nature of computing
- Technology as a tool for solving problems
- The relevance of computer science and its impact on society

There are many technological tools that enable people to explore concepts and create exciting and personally relevant artifacts that impact society. In this course, programming is used as one of the tools, but not the only tool. Students are asked to be creative in designing and implementing solutions as they translate ideas into tangible forms. As students actively create, they will also discuss the broader implications of computing technologies.

Throughout the course students will gain experience in employing the following computational practices.

- Analyze the effects of developments in computing.
- Design and implement creative solutions and artifacts.
- Apply abstractions and models.
- Analyze their computational work and the work of others.
- Connect computation with other disciplines.
- Communicate thought processes and results.
- Work effectively in teams.

As students design and implement solutions using abstractions and models, they will analyze the processes they and their peers use to arrive at solutions, study the effects of their creations and learn how computing concepts connect explicitly and implicitly to other disciplines. Students will learn about the collaborative nature of computer science by working in teams and communicate the results of their work in writing and orally supported by graphs, visualizations and computational analysis.

Scope and Sequence

Topic	Focus	HCI	PS	WEB	PR	DA	ROB
1. Computers and the internet (~2 weeks)	1. Hardware components	I		R	R	A	A
	2. Software components	I		R	R	A	A
	3. Interaction of components	I		R	R	A	A
	4. Selection of appropriate components	I					
	5. Search engine fundamentals	I		R			
	6. Collaborative tools	I		R			
	7. Evaluating websites	I		R			
	8. Security on the Internet	I		R			
2. Models of intelligent behavior (~2 weeks)	1. What is intelligence?	I					
	2. Computers vs. humans	I	R	R	R	R	R
3. Algorithms and abstraction (~6 weeks)	1. Understanding the problem		I	R	R	A	A
	2. Exploring problems: problem-solving heuristics and strategies		I	R	R	A	A
	3. Design creation and representation		I	R	R	A	A
	4. Problem data		I	R	R	A	A
	5. Solution accuracy		I	R	R	A	A
	6. Design re-evaluation and refinement		I	R	R	A	A
	7. Decompose the complex		I	R	R	A	A
	8. Communicate results		I	R	R	A	A
	9. Algorithm efficiency		I		R	R	R
	10. Computationally intensive problems		I			R	R
	11. Unsolvable problem for a computer		I			R	R
	12. Computationally hard problems.		I			R	R
4. Connections between mathematics and computer science (~2 weeks)	1. Logic		I		R	A	A
	2. Binary number system		I				
	3. Basic Sets		I		R	A	A
	4. Concepts of functions		I		R	A	A
	5. De Morgan's laws		I		R	A	A
	6. Graphs		I		R	A	A
5. Creating computational artifacts (Web pages, programs, and robots) (~14 weeks)	1. Break a problem statement into specific requirements		I	R	R	R,A	R,A
	2. Design a solution to a problem		I	R	R	R,A	R,A
	3. Choose appropriate tools and techniques		I	R	R	R,A	R,A
	4. Code a solution from a design			I	R	R,A	R,A

	5. Test a solution to identify errors		I	R	R	A	A
	6. Refine solution		I	R	R	A	A
	7. Documentation and justification		I	R	R	A	A
6. Data and information (~7 weeks)	1. Representation and storage	I	R		R	A	
	2. Methods for collection and generation	I	R			A	
	3. Patterns, trends, and discoveries	I	R			A	
	4. Evaluation		I			R,A	
	5. Computational models	I	R			A	
	6. Rapid testing		I			R,A	
7. Societal impacts of computing (weave throughout)	1. Fostering innovation						
	2. Legal and ethical concerns						
	3. Privacy and cyber security						
	4. Exploitation of information						
	5. Intellectual property						
	6. Limits on information access						
	7. Cultural influence						
	8. Equity, access, and power						
	9. Social and economic values						

Overview Chart

Human Computer Interaction Unit Overview	
Instructional Day	Topic
1-2	Explore the concepts of <i>computer</i> and <i>computing</i> .
3-4	“Demystify” and learn the function of the parts of a personal computer. Learn the terminology of hardware components necessary for the purchase of a home computer.
5-7	Explore the world wide web and search engines. Experiment with a variety of search techniques, internet resources, and Web 2.0, applications. Evaluate websites.
8-9	Examine the implications of data on society and how computers are used for communications.
10	Tell a story with data.
11-14	Explore how computers are used as a tool for visualizing data, modeling and design, and art in the context of culturally situated design tools.
15-16	Introduce the concept of a computer program as a set of instructions.
17-19	Explore the idea of intelligence—especially as it relates to computers. Explore what it means for a machine to “learn”. Discuss whether computers are intelligent or whether they only behave intelligently.
Problem Solving Unit Overview	
Instructional Day	Topic
1-2	Introduce data collection and problem solving.
3	Introduce the four steps of the problem solving process.
4-6	Apply the problem solving process. Use different strategies to plan and carry out the plan to solve several problems.
7-9	Reinforce the four steps of the problems solving process.
10-12	Count in the binary number system. Convert between binary and decimal numbers in the context of topics that are important to computer science.
13-14	Introduce the linear and binary search algorithms.
15-16	Explore sorted and unsorted lists and various sorting algorithms.
17	Introduce minimal spanning trees and how graphs can be used to help solve problems.
18-21	Final projects and presentations

Web Design Unit Overview	
Instructional Day	Topic
1-2	Explore issues of social responsibility in web use as well as the relative merits of the influence of the web on society, personal lives, and education.
3-4	Introduce the use of basic html.
5	Introduce basic formatting in html.
6-7	Explore image editing for the web using Photoshop or an image editor of choice.
8-10	Introduce basic css.
11-13	Explore the concept of separating style from structure by keeping separate html and css files.
14	Add hyperlinks to other websites.
15-16	Introduce a variety of page layout styles.
17-19	Practice the use of various design elements.
20-21	Introduce several different enhancements for website design, including web user interface elements combining Javascript, html, css, and Photoshop, accordion menus, lightbox and sliding images.
22-25	Final projects and gallery walk
Introduction to Programming Unit Overview	
Instructional Day	Topic
1	Introduce the Scratch programming language, including the basic terms utilized in the language.
2-3	Practice using the basic features of Scratch in the context of creating a simple program.
4	Create a dialogue between two sprites.
5-6	Introduce the methods of moving sprites in Scratch.
7-8	Practice the concept of event driven programming through the creation of an alphabet game.
9	Introduce the concept of broadcasting via role play.
10-13	Write Scratch stories and present them to the class. Conduct peer reviews.

14	Introduce the concept of variable.
15	Introduce the concept of conditionals.
16-17	Introduce And, Or and randomness.
18	Apply knowledge of conditionals to develop a Rock Paper Scissors program in Scratch.
19	Build on previous programming concepts to create a timer.
20-23	Create a timing game in Scratch and present it to the class. Peer reviews are conducted.
24	Investigate two types of games that may provide ideas for the final project.
25	Explain final project and the rubric for the final project.
26-28	Write Scratch programs for either My Community or Game project. Conduct peer reviews.
29	Complete final projects.
30	Presentations of final projects
Computing and Data Analysis Unit Overview	
Instructional Day	Topic
1	Review how data can be used for making a case/discovery and provide an overview of the final project.
2	Discuss photo ethics and student safety related to android phone use.
3-5	Distribute phones. Create groups. Discuss group roles and responsibilities. Navigate the android application. Navigate the online system.
6	Data check-in—Discuss issues that arise (aggregating data, etc.).
7-10	Introduce R/Deducer. Create maps using the latitude and longitude of a location and then create maps from a file of data.
11	Create maps with student data and related data set.
12-14	Discuss bar plots, categorical and continuous data, and mosaic plots as a vehicle for comparing categorical data, and looking at trends in data.
15	Create bar plots and mosaic plots with student data and related data set.
16-18	Review mean, median, minimum, maximum. Discuss various ways to subset data. Represent data with box plots and histograms.

19	Identify mean, median, minimum, maximum, create subsets, and create box plots and histograms with student data and related data set.
20-22	Use a variety of filters and queries to create subsets of text data. Create bar plots to graphically display the information.
23	Analyze text in student data and related data set.
24-26	Finalize data analysis for final project.
27-29	Develop website or Scratch program to present data analysis campaign.
30	Final project presentations
Robotics Unit Overview	
Instructional Day	Topic
1	What is a robot? Identify the criteria that make an item a robot.
2-3	Evaluate robot body designs and create algorithms to control robot behavior.
4	Set up LEGO® Mindstorms® NXT® kit.
5	Build robot base.
6-7	Introduce the features of NXT Brick—the “brain” of the robot.
8-9	Introduce the features of the Mindstorms NXT software.
10-13	Program the robot using the Mindstorm Robot Educator Software tutorials.
14	Introduce RoboCup real life robotic competition and write instructions for tic-tac-toe.
15	RoboTic-Tac-Toe Tournament and introduction to RoboCupJunior Dance Challenge.
16-18	Build, program, and present a dancing robot.
19-23	Build program and present a rescue robot.
24-33	Final projects and presentations

Topic Descriptions and Objectives

Unit 1: Human Computer Interaction (4 weeks)

Topics to be addressed:

- Computers and the internet
- Models of Intelligent Behavior
- Societal impacts of computing

Topic Description:

In this unit students are introduced to the concepts of computer and computing while investigating the major components of computers and the suitability of these components for particular applications. Students will experiment with internet search techniques, explore a variety of websites and web applications and discuss issues of privacy and security. Fundamental notions of Human Computer Interaction (HCI) and ergonomics are introduced. Students will learn that “intelligent” machine behavior is not “magic” but is based on algorithms applied to useful representations of information, including large data sets. Students will learn the characteristics that make certain tasks easy or difficult for computers, and how these differ from those that humans characteristically find easy or difficult. Students will gain an appreciation for the many ways in which computing-enabled innovation have had an impact on society, as well as for the many different fields in which they are used. Connections among social, economical and cultural contexts will be discussed.

Objectives:

The student will be able to:

- Analyze the characteristics of hardware components to determine the applications for which they can be used.
- Use appropriate tools and methods to execute Internet searches which yield requested data.
- Evaluate the results of web searches and the reliability of information found on the Internet.
- Explain the differences between tasks that can and cannot be accomplished with a computer.
- Analyze the effects of computing on society within economic, social, and cultural contexts.
- Communicate legal and ethical concerns raised by computing innovation.
- Explain the implications of communication as data exchange.

Unit 2: Problem Solving (4 weeks)**Topics to be addressed:**

- Algorithms and abstraction
- Connections between Mathematics and Computer Science
- Societal impacts of computing

Topic Description:

This unit provides students with opportunities to become “computational thinkers” by applying a variety of problem-solving techniques as they create solutions to problems that are situated in a variety of contexts. The range of contexts motivates the need for students to think abstractly and apply known algorithms where appropriate, but also create new algorithms. Analysis of various solutions and algorithms will highlight problems that are not easily solved by computer and for which there are no known solutions. This unit also focuses on the connections between mathematics and computer science. Students will be introduced to selected topics in discrete mathematics including Boolean logic, functions, graphs and the binary number system. Students are also introduced to searching and sorting algorithms and graphs.

Objectives:**The student will be able to:**

- Name and explain the steps they use in solving a problem.
- Solve a problem by applying appropriate problem-solving techniques.
- Express a solution using standard design tools.
- Determine if a given algorithm successfully solves a stated problem.
- Create algorithms that meet specified objectives.
- Explain the connections between binary numbers and computers.
- Summarize the behavior of an algorithm.
- Compare the tradeoffs between different algorithms for solving the same problem.
- Explain the characteristics of problems that cannot be solved by an algorithm.

Unit 3: Web Design (5 weeks)**Topics to be addressed:**

- Web page design and development
- Computers and the internet
- Algorithms and abstraction
- Societal impacts of computing

Topic Description:

This section prepares students to take the role of a developer by expanding their knowledge of algorithms, abstraction, and web page design and applying it to the creation of web pages and documentation for users and equipment. Students will explore issues of social responsibility in web use. They will learn to plan and code their web pages using a variety of techniques and check their sites for usability. Students learn to create user-friendly websites. Students will apply fundamental notions of Human Computer Interaction (HCI) and ergonomics.

Objectives:**The student will be able to:**

- Create web pages to address specified objectives.
- Create web pages with a practical, personal, and/or societal purpose.
- Select appropriate techniques when creating web pages.
- Use abstraction to separate style from content in web page design and development.
- Describe the use of a website with appropriate documentation.

Unit 4: Introduction to Programming (6 weeks)**Topics to be addressed:**

- Programming
- Algorithms and abstractions
- Connections between mathematics and computer science
- Societal impacts of computing

Topic Description:

Students are introduced to some basic issues associated with program design and development. Students design algorithms and create programming solutions to a variety of computational problems using an iterative development process in Scratch. Programming problems include mathematical and logical concepts and a variety of programming constructs.

Objectives:**The student will be able to:**

- Use appropriate algorithms to solve a problem.
- Design, code, test, and execute a program that corresponds to a set of specifications.
- Select appropriate programming structures.
- Locate and correct errors in a program.
- Explain how a particular program functions.
- Justify the correctness of a program.
- Create programs with practical, personal, and/or societal intent.

Unit 5: Computing and Data Analysis (6 weeks)**Topics to be addressed:**

- Data and information
- Algorithms and abstraction
- Connections between mathematics and computer science
- Programming
- Societal impacts of computing

Topic Description:

In this unit students explore how computing has facilitated new methods of managing and interpreting data. Students will use computers to translate, process and visualize data in order to find patterns and test hypotheses. Students will work with a variety of large data sets that illustrate how widespread access to data and information facilitates identification of problems. Students will collect and generate their own data related to local community issues and discuss appropriate methods for data collection and aggregation of data necessary to support making a case or facilitating a discovery.

Objectives:**The student will be able to:**

- Describe the features of appropriate data sets for specific problems.
- Apply a variety of analysis techniques to large data sets.
- Use computers to find patterns in data and test hypotheses about data.
- Compare different analysis techniques and discuss the tradeoffs among them.
- Justify conclusions drawn from data analysis.

Unit 6: Robotics (7 weeks)**Topics to be addressed:**

- Robotics
- Algorithms and abstraction
- Connections between mathematics and computer science
- Programming
- Societal impacts of computing

Topic Description:

This unit introduces robotics as an advanced application of computer science that can be used to solve problems in a variety of settings from business to healthcare and how robotics enables innovation by automating processes that may be dangerous or otherwise problematic for humans. Students explore how to integrate hardware and software in order to solve problems. Students will see the effect of software and hardware design on the resulting product. Students will apply previously learned topics to the study of robotics.

Objectives:**The student will be able to:**

- Identify the criteria that describe a robot and determine if something is a robot.
- Match the actions of the robot to the corresponding parts of the program.
- Build, code, and test a robot that solves a stated problem.
- Explain ways in which different hardware designs affect the function of a machine.
- Describe the tradeoffs among multiple ways to program a robot to achieve a goal.

The societal impacts of computing should be woven throughout the course.

Topic Description:

Throughout the course, is placed on how computing enables innovation in a variety of fields and the impacts that those innovations have on society. Computing is situated within economic, social and cultural contexts and, therefore, influences and is influenced by each of these. The proliferation of computers and networks raises a number of ethical issues. Technology has had both positive and negative impacts on human culture. Students will be able to identify ethical behavior and articulate both sides of ethical topics. Students study the responsibilities of software users and software developers with respect to intellectual property rights, software failures, and the piracy of software and other digital media. They are introduced to the concept of open-source software development and explore its implications. Students identify and describe careers in computing and careers that employ computing.

Objectives:

The student will be able to:

- Describe ways in which computing enables innovation.
- Discuss the ways in which innovations enabled by computing affect communication and problem solving.
- Analyze how computing influences and is influenced by the cultures for which they are designed and the cultures in which they are used.
- Analyze how social and economic values influence the design and development of computing innovations.
- Discuss issues of equity, access, and power in the context of computing resources.
- Communicate the legal and ethical concerns raised by computational innovations.
- Discuss privacy and security concerns related to computational innovations.
- Explain positive and negative effects of technological innovations on human culture.

Unit 1:

Human Computer Interaction



Introduction

Computers and other computing technology have become an integral part of our society. Because of that it is easy to forget that not every student will come to this course with the same background and skills. For many students this course will be the first opportunity they have to become facile with using the keyboard and navigating the internet.

The topics in this unit are designed to allow all students to gain familiarity with computers and computing in the context of activities that give students an opportunity to work at their own pace, collaborate in groups where they can learn from each other and generally gain an overview of the many and varied ways in which computers and computing are used.

The unit is divided into three main sections.

- Exploring the concepts of computer and computing by investigating computer hardware components and a variety of internet resources (Days 1-7)
- Exploring the use of computers in a variety of circumstances and fields (Days 8-13)
- The computer as a machine that needs to be provided specific instructions (Days 14-19)

The goal is for all students in the class to reach a level of comfort in using the computer and understand that the computer is not magic. The fact that for a computer to accomplish its tasks it needs to be given precise instructions motivates the need for the problem solving techniques that will be addressed in Unit 2.

Specific topics for each instructional day are listed in the overview chart on the next page.

Daily Overview Chart	
Instructional Day	Topic
1-2	Explore the concepts of <i>computer</i> and <i>computing</i> .
3-4	“Demystify” and learn the function of the parts of a personal computer. Learn the terminology of hardware components necessary for the purchase of a home computer.
5-7	Explore the world wide web and search engines. Experiment with a variety of search techniques, internet resources, and Web 2.0, applications. Evaluate websites.
8-9	Examine the implications of data on society and how computers are used for communications.
10	Tell a story with data.
11-14	Explore how computers are used as a tool for visualizing data, modeling and design, and art in the context of culturally situated design tools.
15-16	Introduce the concept of a computer program as a set of instructions.
17-19	Explore the idea of intelligence—especially as it relates to computers. Explore what it means for a machine to “learn”. Discuss whether computers are intelligent or whether they only behave intelligently.

Daily Lesson Plans

Instructional Days: 1-2

Topic Description: What is a computer? In this lesson the concepts of *computer* and *computing* are explored through examples of each.

Objectives:

The student will be able to:

- Explain and give examples of the concepts of *computer* and *computing*.

Outline of the Lesson:

- Journal Entry. (10 minutes)
- Exploring computers (60 minutes)
- Classification of computing groups (10 minutes)
- Definition of the terms *computer* and *computing* (10 minutes)
- Demo of Computer Buying Project Assignment (20 minutes)

Student Activities:

- Complete journal entry.
- Groups of students create lists of their ideas of what a computer is.
- Groups do presentations.
- Participate in discussion of classification groups and definition of computer and computing.

Teaching/Learning Strategies:

- Journal Entry: How many computers are in the room?
 - Have students write responses to the question in their journals and then share the response with their elbow partner.
 - Ask a few student pairs to share their responses.
- Exploring computers
 - Some students may have counted only the desktop computers in the room while others may have recognized that there are other items that are computers as well.
 - Divide students into groups of 3 or 4. Ask the students to discuss additional examples of computers (or things containing computers). Have students write their examples on post-its and add them to a large chart at the front of the room or have groups make a list on flipchart paper and tape their list somewhere in the classroom. (Examples of computers include: Macintosh, Windows PC, cell phone, mp3 player, most appliances (television, coffee maker,

- washer, dishwasher, etc.), cars, medical equipment, planes, watches, cash registers, ATMs, traffic lights, scoreboards, humans, and calculators.)
- Have student groups share their ideas. After each presentation, give the other students an opportunity to suggest why any particular example seems not to be a computer (or is not obviously a computer). If necessary, ask questions to draw out the student questions and responses. (For example, if the student says “dishwasher,” you might ask, “why is a dishwasher a computer.”)
 - Have a brief discussion of the power of cell phones. Mention collection of data as a foreshadowing of unit 5.
 - Classification of computing groups
 - Ask students to suggest possible classifications for the items on the list; create a new list with the various items listed under a group classification.
 - Definition of the terms *computer* and *computing*.
 - Revisit the question “What is a computer?” and ask the possibly more pertinent question, “What is *computing*?”
 - Have the students use their list of “computers” and their classifications to help formalize their answers.
 - Note that there is no “correct” answer. These definitions will be revisited and possibly modified throughout the course of the unit.
 - Reinforce the idea of different types of computers and classifications by reviewing the lists and groups created by the students.
 - Computer Buying Project Assignment
 - Each student will interview a family member or friend to find out what features that person would like to have if they were buying a new personal computer.
 - Demonstrate the interview process by asking a student to participate in an interview and ask them questions such as: What will be the uses of the computer? What are the space constraints? What is the price range? Etc. You may wish to provide students with a specific list of interview questions.

Resources:

- No additional resources needed

Instructional Days: 3-4

Topic Description: Students complete a project related to choosing appropriate components for a personal computer.

Objectives:

The student will be able to:

- Describe the uses for computer hardware components.
- Choose hardware components for various types of users.

Outline of the Lesson:

- Research and development of computer buying project (85 minutes)
- Project presentations by student teams (25 minutes)

Student Activities:

- Student teams research and complete projects.
- Student teams present projects.

Teaching/Learning Strategies:

- Distribute project information and rubric.
 - Have students work in the same teams as they did on the previous day.
 - Explain project and rubric; answer questions; Have each team choose one of the interviews they did to use for the project.
- Project presentations by student teams
 - Have each team discuss the scenario they used, display their comparison chart and explain which computer was chosen and why.

Resources:

- Computer Buying Project
- Computer Components Checklist
- Computer Buying Project Sample Rubric

Computer Buying Project

Choose one of the interviews of buying a new computer to use for this project. Your task is to give them at least 4 options and then give them advice on which one to buy. Your project will be presented to the class.

The final product can be a:

- PowerPoint
- Debate
- Skit
- Video
- Other approved product

Your final product should have:

- A title with group members' names
- The questions and responses from the interview chosen
- Computer comparison chart
- Which computer is chosen
- Justification for choosing that computer, including completion of the Computer Components Checklist.
- You should be prepared to answer questions about your choices from other students.

Example computer comparison chart (more information can be added):

	Dell Inspiron	MacBook
Laptop or Desktop		
Processor (CPU)		
Operating System		
Memory (RAM)		
Hard Drive (storage)		
Optical Drive		
Monitor or Screen Size		
Video Card		
Sound Card		
Other Accessories		
Cost		

Student Grouping:

You will be in a group of up to 4 students.

Computer Components Checklist

Below are the different components for you to investigate:

1. Processor
2. Operating System
3. Memory
4. Hard Drive
5. Optical Drive
6. Monitor
7. Video Card
8. Sound Card
9. Speakers
10. Keyboard
11. Mouse
12. Modem

For each of the components write down:

- The name
- What it is used for
- What are the different options or sizes for the components
- Why you included the component in the list of components for the person who is buying the computer.

Computer Buying Project Sample Rubric

Group Members Names: (up to 4)

Do you have?	Points Possible	Yes	No	Points Earned
Product				
Title with group members' names	5			
Interview scenario is described	10			
Computer comparison chart with at least 4 options	10			
Chart has specifications of options	5			
Options on chart fit your scenario	10			
Show justification for your computer choice	10			
Your choice fits the scenario	10			
Visuals of your choices (pictures or video of choices)	10			
Computer Components Checklist	5			
Presentation				
Present all required parts of project	10			
All group members participate	5			
Able to answer questions	10			
Extra Credit				
Project exhibits creativity above and beyond	Up to 10			
TOTAL:	100 + 10			

Instructional Days: 5-7

Topic Description: Search engines and how they work are explored through trying various internet search techniques. A selection of Internet resources that are useful for finding information are introduced as well as a selection of Web 2.0 applications. Several websites are evaluated by using a rubric to determine if they are “good” websites.

Objectives:

The student will be able to:

- Perform searches and explain how to refine searches to retrieve better information.
- Identify resources for finding information in addition to ranking based search engines.
- Differentiate between ranking based search engines and social bookmarking (collaborative) search engines.
- Use a variety of Web 2.0 applications.
- Develop and use a rubric to evaluate websites.

Outline of the Lesson:

- Journal Entry (5 minutes)
- Internet Scavenger Hunt (25 minutes)
- Discussion of other resources for finding information (10 minutes)
- Experimentation with these resources (15 minutes)
- Jigsaw activity involving Web 2.0 applications (55 minutes)
- Journal entry (5 minutes)
- Web site evaluation criteria (20 minutes)
- Hands-on evaluation of web sites (30 minutes)

Student Activities:

- Complete journal entry.
- Perform internet searches using varying levels of refinement.
- Identify other resources for finding information.
- In groups use the other resources to find relevant information.
- In groups complete jigsaw Web 2.0 activity and presentations.
- Complete journal entry
- Identify evaluation criteria and work in groups to evaluate websites using the rubric.

Teaching/Learning Strategies:

- Journal Entry: List at least three ways in which you currently use the internet.
 - Have students share their responses with their elbow partner.
- Have student groups complete an internet scavenger hunt.
 - A sample is provided, but you may want to create your own that is more specific to the interests of your

students or have the groups create their own and exchange with another group.

- Other resources for finding information
 - In discussing the results of the scavenger hunt have students identify at least three resources other than search engines that they use to find information on the internet along with advantages (or disadvantages) over a general search engine.
 - Some examples might be:
 - Sites such as Google Maps or Mapquest to get directions or see satellite or street view images of anywhere in the country.
 - Address and telephone number lookup sites such as Switchboard or Yellow Pages to get personal and business information.
 - Sites such as the Internet Movie Database to get information on movies and television shows.
 - Sites such as Dictionary and Thesaurus to look up the meaning or spelling of a word or to find a synonym of a word.
 - Encyclopedic sites such as Wikipedia, Encyclopedia Britannica, or How Stuff Works to find an overview of a particular topic.
 - The Wayback Machine which stores snapshots of websites on various dates so that you can “go back in time” to see a site as it used to be.
 - Video-based information sources such as YouTube and Howcast
- Experimentation with these resources
 - Have the students work in groups to use the resources identified above in ways that are relevant to them. For example,
 - Use Google maps and StreetView to find and display where they live or the location of the school.
 - Use Wikipedia and Encyclopedia Britannica to find information on a topic they’re studying in another class. Have them compare the two articles and decide which provides more information.
 - Use the Wayback Machine to view an early version of the school website. Compare how much it has changed from the school’s current website.
- Jigsaw activity involving Web 2.0 applications
 - Divide students into groups to work on each of three different Web 2.0 applications. (Depending on the size of the class, more than one group may need to work on each application.) Applications should include a social bookmarking site (delicious.com or stumbleupon.com), a word cloud site (wordle.net) and a list creation site (tadalist.com). Each group should:
 - Set up an account in the application.
 - Explore the site and its features.
 - Prepare a presentation on their site for the remainder of the class.
 - During the student presentations, ensure that the following questions/issues are addressed:
 - What are the differences between ranking based and social bookmarking search engines?
 - Why would you want to create word clouds?
 - What are the advantages of using tadalist? Disadvantages?

- What issues might there be with creating accounts online? (Lead into a discussion on privacy— what information should be kept private and why? Discuss encryption.)
- Journal Entry: Why do you need to evaluate websites?
- Web site evaluation criteria
 - Display or distribute a copy of the front page to <http://www.martinlutherking.org> This is a website which purports to be a “True Historical Examination” of the life of Martin Luther King, Jr., but is, in reality, a hateful site run by a white nationalist organization.
 - This particular site is obviously biased. However, it is important to be able to tell when a site is more subtly biased.
 - Brainstorm criteria that they could use to evaluate websites. As a group develop an evaluation rubric. (You can use the Sample Website Evaluation Rubric as a reference.)
- Hands-on evaluation of web sites.
 - Have the students work in groups and ask them to use the website evaluation rubric to evaluate websites they might need for a school project or an assignment from another class.
 - Discuss the results of their evaluations.

Resources:

- Sample Scavenger Hunt
- The Wayback Machine: <http://www.archive.org>
- Google Maps (including StreetView): <http://maps.google.com>
- Wikipedia: <http://www.wikipedia.org>
- Encyclopedia Britannica: <http://www.britannica.com>
- Mapquest: <http://www.mapquest.com>
- Internet Movie Database: <http://www.imdb.com>
- Switchboard: <http://www.switchboard.com>
- Yellow Pages: <http://www.yellowpages.com>
- How Stuff Works: <http://www.howstuffworks.com>
- YouTube: <http://www.youtube.com>
- Howcast: <http://howcast.com>
- <http://www.delicious.com>
- <http://www.stumbleupon.com>
- <http://www.wordle.net>
- <http://www.tadalist.com>
- The white nationalist site on Martin Luther King, Jr.: <http://www.martinlutherking.org>
- Website Evaluation Rubric

Sample Scavenger Hunt

In your group, use the internet to find the following items. For each item include the steps you took to find each item.

1. A picture of the mayor of your town or city
2. A bus schedule
3. The address of the Chamber of Commerce for your town or city
4. A map of your state—and you have to point out where your town or city is!
5. A copy of the front page of your town's or city's web site
6. Something in writing that tells how many people live in the city
7. A picture of any historical landmark in the city
8. A picture of your congressman
9. A program or flyer from a local arts event
10. The names of all the city council members
11. Something that gives information about your local hospital
12. A list of schools in your town or city
13. The phone number of the local police department
14. Anything with the colors or mascot of a local college or community college
15. A picture of the state flag
16. A picture of the state bird
17. A schedule of activities or a pamphlet from a local nursing home or senior citizens center
18. A sticker or button from a local election
19. A list of safety tips from the local fire department
20. A speech by your governor

Sample Website Evaluation Rubric			
Authority			
Is the author identified?	Yes	No	Unsure
Does the author have appropriate qualifications with respect to the information being presented?	Yes	No	Unsure
Purpose			
Is the purpose to inform or give factual information?	Yes	No	Unsure
Coverage			
Is the information primary or secondary in nature?	Yes	No	Unsure
Is the information presented comparable to information on the same topic presented by other sites?	Yes	No	Unsure
Accuracy			
Is the information free of factual errors?	Yes	No	Unsure
Do the conclusions appear to be well-reasoned and supported by the facts presented?	Yes	No	Unsure
Is the information properly referenced?	Yes	No	Unsure
Objectivity			
Is the information free from obvious bias?	Yes	No	Unsure
Does the author avoid the use of emotional or inflammatory language?	Yes	No	Unsure
Does the author avoid trying to sell something or persuade the reader of a particular viewpoint?	Yes	No	Unsure
Currency			
Is the information up-to-date?	Yes	No	Unsure
Are there creation and revision dates?	Yes	No	Unsure
Appearance			
Does the site have a professional appearance?	Yes	No	Unsure
Does it use proper grammar, spelling, and composition?	Yes	No	Unsure

Instructional Day: 8-9

Topic Description: The use of computers for communications and the impact this has had on society is discussed.

Objectives:

The student will be able to:

- Explain how computers are used for communications.
- Recognize various forms of communication as data exchange.
- Describe the implications of data exchange on social interactions.
- Consider privacy of data that they create.

Outline of the Lesson:

- Journal Entry (5 minutes)
- Identification of communications mechanisms (10 minutes)
- Communication Methods Chart (30 minutes)
- Impact of changes to communications on society (10 minutes)
- Privacy Activity (55 minutes)

Student Activities:

- Complete journal entry.
- Identify communications mechanisms.
- Pairs of students complete the Communication Methods Chart.
- Groups discuss the impact of changes to communications on society.
- Groups share a summary of their discussions with the class.
- Groups of students complete the Privacy Activity.

Teaching/Learning Strategies:

- Journal Entry: List as many computer-based communications mechanisms as you can.
- Identification of communications mechanisms
 - Volunteers provide examples from their journal entry. Post the responses.
 - Prompt students as necessary with examples.
 - Internet-based communication (email, chat, Facebook, Internet telephony)
 - Telephone-based communication (cell phones, texting, “landline” telephone service)
 - News and information “on demand”

- Communication Methods Activity
 - Students complete Communication Methods Chart.
 - Students work in pairs (option to have each pair complete one sheet for both or each complete one).
 - Take a poll for each category to see what form of communication is most popular for each scenario.
 - Have a few students share their answer to “Why you chose this Method” for each scenario.
 - Have a few students share their answers to 1-2.
- Impact of changes to communications on society
 - Ask students to do the following:
 - Imagine life without some or all of the computer-based communications mechanisms that we now take for granted.
 - List some of the consequences of an absence of technology (for example, without cell phones, the ability to instantly reach anyone goes away).
 - Based on these consequences, draw conclusions about the impact of the presence of the communications mechanism. (For example, if the absence of cell phones means the absence of the ability to instantly contact anyone, then the presence of cell phones means that we now have the ability to instantly contact anyone. One conclusion we can draw is that we have less privacy than we used to.)
- Privacy Activity
 - Students are placed into groups and assigned a scenario.
 - They are to prepare a presentation to answer the questions for their given scenario.
- Groups deliver presentations.
- Challenge the students to predict what communications will be like in 5 years, 10 years, and 25 years.

Resources:

- Communication Methods Chart
- Privacy Activity

Communication Methods Chart

Let's look at the different ways you can communicate with others. For each of the following examples, fill in the table below with which method you would choose for the given scenario and why.

The methods are:

- **Texting**
- **Phone call**
- **Talking in person**
- **MySpace / Facebook**
- **Twitter**
- **Email**

Scenario to communicate	Method	Why you chose this method
1. Breaking up with a significant other (boyfriend/girlfriend)		
2. Asking parents' permission to do something when you think they will likely say 'no'		
3. Figure out where and when to meet a friend to see a movie		
4. Gossip about someone who could hear you if you spoke		
5. Gossip about someone not around you		
6. Getting help on homework		
7. Feedback on a big decision (like what color prom dress, what game to buy, what phone to get)		
8. Announce you met someone famous		
9. Complain about your parents		
10. Mourn someone you lost		
11. Buy something from someone you don't know well		

1. How would life be different if you could only communicate 1 on 1 instead of with multiple people at once?

2. How would life be different if you could only communicate in person?

Privacy Activity

Data are everywhere. You are 'giving off data' and providing data to others all the time. Sometimes this data can be directly linked to you as an individual; sometimes not.

Read the scenario assigned to you (many of these are based on real stories):

- A. A boss sees an employee who called in “sick” in a picture that someone posted on Facebook. In this picture the employee is partying the night before. The boss fires the employee.
- B. A company who has contracts with the Federal Government doesn’t want to hire you because a Facebook friend leaves lots of enthusiastic “legalize marijuana” postings on your wall.
- C. A teacher is fired because there’s a picture of this teacher holding alcoholic drinks on her MySpace page.
- D. Someone’s Netflix rental history is being used as evidence in a murder case because this person rented a lot of horror movies.
- E. An 18-year-old boy is charged with distributing child pornography when he uses his cell phone to send naked images of his 17-year-old ex-girlfriend to his friends.

Prepare a 3-5 minute presentation for the class that includes answers to the following:

1. Names of everyone in your group
2. The scenario you were assigned
3. In your scenario, did the people have the right to use the information they did or should it have been private? Why?
4. Are the people who were affected by the use of this information at fault? Why?
5. Give another example of something unexpected happening because of information shared at sites such as Facebook, Twitter, MySpace, blogs.
6. Think about what data you’ve made available in different places/spaces such as Facebook, Twitter, texting, Netflix, email, at the grocery store, etc. What might other people think about who you are based on these data? Is it an accurate impression of who you are?
7. Consider each of the following broad categories of societal change:
 - Privacy
 - Safety
 - Globalization
 - Connectivity (keeping in touch with people)
 - Permanence of historical information

Discuss whether availability of data has had a positive or negative impact on each aspect of society and, if negative, how these consequences can be minimized.

Instructional Day: 10

Topic Description: Points of view: Telling a story with data

Objectives:

The students will be able to:

- Explain how different views of data can tell a different story.
- Recognize that data is an incomplete record of reality.
- Describe the limits of measurement (what can and can't be captured in data).

Outline of the Lesson:

- Journal Entry (10 minutes)
- Room Activity (45 minutes)

Student Activities:

- Complete journal entry.
- Groups complete first part of Room Activity.
- Groups share responses with another group.
- Groups complete second part of Room Activity and share responses.

Teaching/Learning Strategies:

- Journal Entry: What do you think about when you hear the word data? Where can it be found? Where does it come from?
 - Class discussion of journal entries
 - Write down ideas from students.
 - Emphasize that this lesson is meant to stretch their thinking about data.
- Room Activity
 - Assign various groups different versions of the Room Activity.
 - Depending on the amount of time available and the size of the class, you can have some students work with the picture (in the Supplemental Materials) and some work with the entire word list OR you can assign different subsets of the word list to different groups of students.
 - When each group is finished with their first 4 instructions, have them compare with a group that has a different version.
 - Have groups share their answers to #2 with the entire class.
 - Show groups the different versions of the room.
 - Have groups complete questions 5-7.
 - Have some groups share their answers for 5-7

- Emphasize that the appearance of the data and amount of data collected inform the inferences that can be made.
- Homework: Complete Communications Methods and Data Chart and Data Journal
 - Explain that they will be adding to the Communications Methods Chart of the previous day by adding information about data.
 - Introduce the Data Journal.
 - Both of these assignments will be due on the first day of Unit 2.
 - Clarify questions.

Resources:

- Room Activity Picture (Supplemental Materials)
- Room Activity Picture Instructions (to go with the picture)
- Room Activity Entire Word List
- Room Activity Possible Word Subsets
- Communications Methods and Data Chart
- Data Journal

Room Activity Picture Instructions

Look at the picture displayed on your computer or on the handout provided.

1. Make a list of the objects in the picture.
2. What does this data tell you about the person who lives in this room? What does it not tell you?
3. What are most of the items in this room related to?
4. How many toy soldiers are there?

Now compare with the other group.

5. Are there any advantages to one representation or another?
6. Do you think different representations can tell different stories?
7. What do you think came first the drawing of the room or the list?

Room Activity Entire Word List

Look at the following list of items found in someone's room.

Laptop	Violin
Nintendo DS	Globe
Mobile phone	Person
iPod with ear buds	Paintings (2)
PS3	Shoe
Radio (2)	Burger
Television	Books (75)
Toy Car (3)	Pizza
Calendar	Guitar
Ribbons (3)	Sandwich
Sprite Can	Goldfish
Plaques (3)	Toy soldiers (3)
Gold Medal	Pringles can
Phone	
Mug	
Trophy	
Picture Frame	
Posters (8)	
Harry Potter poster	
Lava lamp	
Glasses	

1. Draw a picture of the room including each of the items from the list above.
2. What does this data tell you about the person who lives in this room? What does it not tell you?
3. What are most of the items in this room related to?
4. How many Radios are there?

Now compare with the other group.

5. Are there any advantages to one representation or another?
6. Do you think different representations can tell different stories?
7. What do you think came first (the drawing of the room or the list?)

Room Activity Possible Word Subsets

1. Lava lamp, Books (10), Burger, Cell phone, Pringles can, Television, Calendar, Glasses, Lava lamp, Sandwich, Pizza, Paintings (2), Person, Sprite Can
2. Trophy, Pizza, Guitar, Sandwich, IPod with ear buds, Radio(2), Toy soldiers(3), Person, Shoe, Cell phone, Violin, Harry Potter poster, Ribbons(3), Sandwich, Laptop, Goldfish
3. Books (75), Burger, Globe, Gold Medal, Goldfish, Harry Potter poster, Phone, Paintings(2), Person, Plaques(3), Posters(8), Ribbons(3), Toy soldiers(3), Trophy, Violin
4. iPod with ear buds, Television, laptop, Radio(2), Cell phone, Guitar, Toy Car(3), Nintendo DS, PS3, Burger, Pizza, Person, Pringles can, Sprite Can
5. Word cloud pictured below.



Communication Methods and Data Chart

Let's look at what kinds of data you 'give off' when using the different forms of communication. For each of the following examples, fill in the table below with which method you would choose for the given scenario and why (You should already have completed that part.). Keep in mind that "data" here is not just the content you communicate (what you say or write) but could also refer to details like the time of a telephone call and the number.

The methods are:

- **Texting**
- **Phone call**
- **Talking in person**
- **MySpace / Facebook**
- **Twitter**
- **Email**

Scenario to communicate	Method	What data is available?	Who has access to the data?	What can be learned from the data in aggregate?	Why you chose this method
1. Breaking up with a significant other (boyfriend/girlfriend)					
2. Asking parents' permission to do something when you think they will likely say 'no'					
3. Figure out where and when to meet a friend to see a movie					
4. Gossip about someone who could hear you if you spoke					
5. Gossip about someone not around you					
6. Getting help on homework					

7. Feedback on a big decision (like what color prom dress, what game to buy, what phone to get)					
8. Announce you met someone famous					
9. Complain about your parents					
10. Mourn someone you lost					
11. Buy something from someone you don't know well					

1. How does the type of data being exchanged affect which method you choose?

Name_____

Data Journal

During the next several days, take note of situations when you “generate data”. We’re looking for specific moments when some activity you perform can be observed, recorded and, possibly, combined with similar data from others. Ideally you will carry this paper with you and take notes over the course of your day. To start you off, think about what happens when you ride the bus or make a telephone call or browse a web site!

Time	Describe what you did to generate data

Instructional Days: 11-14

Topic Description: In this lesson, students learn how computers can be used as a tool for visualizing data, modeling and design, and art in the context of culturally situated design tools. Connections between the design of the tools and mathematics will be explored.

Objectives:

The student will be able to:

- Explain how computers can be used as tools for visualizing data, modeling and design, and art.
- Identify mathematical connections in the output of the tools.
- Edit an image using Photoshop.

Outline of the Lesson:

- Research on the cultural background associated with the design tool (25 minutes)
- Design tool tutorials (30 minutes)
- Creation of designs using the design tools (65 minutes)
- Online presentation on how to get started using Photoshop (15 minutes)
- Design editing (30 minutes)
- Preparation of presentations (40 minutes)
- Group presentations (15 minutes)

Student Activities:

- Groups do research on the cultural background information associated with the design tools they are assigned and discuss their findings.
- Groups prepare and deliver brief presentations on the cultural aspects of their design tools.
- Students complete design tool tutorials.
- Groups create designs using the design tools.
- Watch an online presentation on how to get started using Photoshop.
- Edit images created with the design tools.
- Groups prepare presentations.
- Groups deliver presentations.

Teaching/Learning Strategies:

- Post the possible design tools:
 - Virtual Bead Loom
 - Pacific Northwest Basket Weaver
 - Navajo Rug Weaver
- Note: You may substitute the Graffiti Art tool if you choose.

- Display the first page of each tool in order to give students an idea of what each does.
(<http://www.csdt.rpi.edu>)
- Students divide into groups to work on the tool of their choice. Group sizes will depend on the size of the class. You may need to have more than one group per tool.
- Each member of the group should go through the entire cultural background section individually.
 - Answer any questions posed in the section in their journal.
 - Look for and write down the mathematical connections.
- All group members discuss the section.
 - Resolve answers to questions and mathematical connections.
- Each member of the group completes the tutorial.
 - Students should go through the tutorial at their own pace, but discuss with other members as questions arise. (Note: The bead loom tutorial is online; the other two are not. The print versions included here have been adapted from the bead loom tutorial.)
 - Encourage students to record in their journal points that they want to remember.
- Groups create designs using the design tool software.
 - Each person should choose one of the goal pictures for practice and discuss any issues with the other group members.
 - Groups decide whether they want to create one design as a group or have multiple designs for their presentation.
 - Groups work on design/designs—these should be their own creations rather than a mimic of one of the preloaded designs.
- Edit designs with Photoshop (or another photo editor of choice).
 - Have students watch the online tutorial and create an account.
 - Edit the design.
- Prepare presentations to include:
 - Culture—an explanation in their own words
 - Math connections—an explanation in their own words
 - Demo of software
 - Display of designs—include a written description of how they created their design
- Groups deliver presentations
 - Groups respond to questions from other students and teacher.
 - Specifically ask what did the computer scientists who created the tools need to know? (computer science, graphics, culture, anthropology, visual arts, language arts, history)
- Display of designs—include a written description of how they created their design.

Resources:

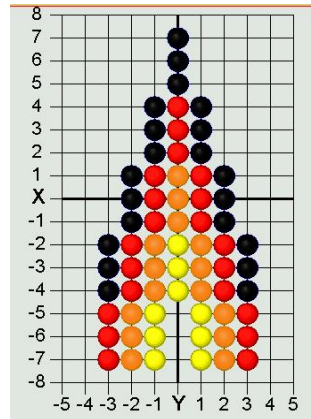
- Culturally Situated Design Tools—<http://www.csdt.rpi.edu> (site and adaptations of tutorials courtesy Ron Eglash)
- Virtual Bead Loom Tutorial
- Pacific Northwest Basket Weaver Tutorial

- Navajo Rug Weaver Tutorial
- Culturally Situated Design Tools Project Sample Rubric
- <http://www.photoshop.com>

Virtual Bead Loom Tutorial

Part 1

The Virtual Bead Loom simulates the same grid pattern as the traditional bead loom. Users place colored circles in columns (the Y-axis) and rows (the X-axis).



There are several tools for placing beads on the virtual loom. In each case you use the "tab" key or the mouse to move your cursor to the field for entering the coordinates, then you enter them, and then press the button for the shape tool. The point tool places a single bead:

x = y =

The line tool places lines of beads. You specify the two endpoints of the line. Diagonal lines tend to be jagged, but resizing the grid can help that (see "Options menu" on next page).

x1 = y1 =

x2 = y2 =

The rectangle tool fills in a rectangle of beads. You specify two vertices (lower right and upper left). The rectangles of this tool are always aligned with the axes.

x1 = y1 =

x2 = y2 =

Part 2

The triangle tool fills in a triangle of beads. You specify the three vertices.

x1 = -3	y1 = 1
x2 = 0	y2 = 4
x3 = 3	y3 = 1

The iterative triangle tool: Our first triangle tool made jagged edges, while traditional beadwork has beautifully regular edges. We interviewed some native beadworkers, and found that their algorithms were iterative. The triangle iteration tool reflects this tradition of indigenous mathematics. For example, the triangle in the beadwork at the top of this page was made by adding one bead on each side of the row, every three rows, as you go in the -Y direction.

Starting at x=0 y=0
 after every 3 rows
 add 1 to both ends
 for 9 rows in total
 +y
 in the -x +x direction.
 -y

- "Direction"—determines in which direction your rows will accumulate
- Starting at X, Y—that is the center of the starting row.
- "After every ___ rows"—lets you determine how many rows you go through before adding more beads to the end.
- "Add ___ to both ends"—the number of beads that will be added on each side of the center each time.
- "For ___ rows in total"—how many rows you will bead in this triangle.

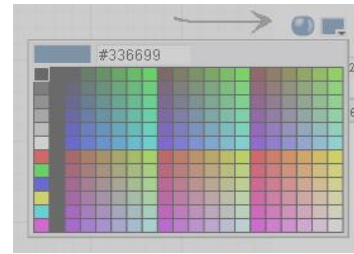
Note that this tool has two colors—some traditional bead work shifts color in each iteration. This allows you to select the starting color and ending color; the software does the shifting for you.

Part 3

There are also controls that apply to all the tools. "Clear" deletes everything. Normally "Create" is selected, so that your tools will fill their specified shape with beads. "Remove" will erase all beads in the specified shape, so if you make an error use "undo" not "remove." The color button allows you to select the bead color. Clicking on the little square in the upper right of the screen will give you a list of all the colors you have selected so far. The "Save" menu allows you to save the work on your hard drive and edit the design later. Make sure your file name is only letters, not spaces or numbers, and that you go back to the same computer when you want to edit your work.

The "Options" menu allows you to resize the grid smaller or larger—maximum size is 150 by 150. You can also change the location of the coordinate values, hide the grid, or create a title or notes about your design. You can also switch to Wampum beads, using either traditional 1X2 Wampum or a 1X1 Wampum (which is easier for math teaching—special thanks to Joyce Lewis of the Onondaga Nation for that concept!).

Printing: after you have your design completed, do a screen capture. In windows you can do that by pressing the "print screen" button on your keyboard, usually located at the upper right above the F10 key. On a Macintosh press shift + apple + 3 at the same time (also shift + apple +4 to select just a portion). That screen capture will save an image of the entire screen to your clipboard. You can then paste the clipboard image into a blank canvas in Word, Photoshop, Imaging (comes free in the "Accessories" folder in Windows) or other image editor.



Pacific Northwest Basket Weaver Tutorial

Part 1

The Virtual Basket Weaver simulates the same grid pattern as the traditional basket weaving loom. Users place colored circles in columns (the Y-axis) and rows (the X-axis).



There are several tools for placing wefts on the virtual loom. In each case you use the "tab" key or the mouse to move your cursor to the field for entering the coordinates, then you enter them, and then press the button for the shape tool. The point tool places a single weft:

x = y =

The line tool places lines of wefts. You specify the two endpoints of the line. Diagonal lines tend to be jagged, but resizing the grid can help that (see "Options menu" on next page).

x1 = y1 =
x2 = y2 =

The rectangle tool fills in a rectangle of wefts. You specify two vertices (lower right and upper left). The rectangles of this tool are always aligned with the axes.

x1 = y1 =
x2 = y2 =


Part 2

The triangle tool fills in a triangle of wefts. You specify the three vertices.

x1 = -3	y1 = 1
x2 = 0	y2 = 4
x3 = 3	y3 = 1

The iterative triangle tool: Our first triangle tool made jagged edges, while traditional basket work has beautifully regular edges. The triangle iteration tool reflects the tradition of indigenous mathematics. For example, a triangle can be made by adding one weft on each side of the row, every three rows, as you go in the -Y direction.

Starting at x=0 y=0
 after every 3 rows
 add 1 to both ends
 for 9 rows in total
 +y
 in the -x +x direction.
 -y



- "Direction"—determines in which direction your rows will accumulate
- Starting at X, Y—that is the center of the starting row.
- "After every ___ rows"—lets you determine how many rows you go through before adding more wefts to the end.
- "Add ___ to both ends"—the number of wefts that will be added on each side of the center each time.
- "For ___ rows in total"—how many rows you will weft in this triangle.

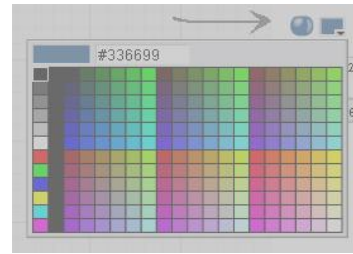
Note that this tool has two colors—some traditional basket work shifts color in each iteration. This allows you to select the starting color and ending color; the software does the shifting for you.

Part 3

There are also controls that apply to all the tools. "Clear" deletes everything. Normally "Create" is selected, so that your tools will fill their specified shape with wefts. "Remove" will erase all wefts in the specified shape, so if you make an error use "undo" not "remove." The color button allows you to select the weft color. Clicking on the little square in the upper right of the screen will give you a list of all the colors you have selected so far. The "Save" menu allows you to save the work on your hard drive and edit the design later. Make sure your file name is only letters, not spaces or numbers, and that you go back to the same computer when you want to edit your work.

The "Options" menu allows you to hide the grid, or create a title or notes about your design.

Printing: after you have your design completed, do a screen capture. In windows you can do that by pressing the "print screen" button on your keyboard, usually located at the upper right above the F10 key. On a Macintosh press shift + apple + 3 at the same time (also shift + apple +4 to select just a portion). That screen capture will save an image of the entire screen to your clipboard. You can then paste the clipboard image into a blank canvas in Word, Photoshop, Imaging (comes free in the "Accessories" folder in Windows) or other image editor.



Navajo Rug Weaver Tutorial

Part 1

The Virtual Rug Weaver simulates the same grid pattern as the traditional rug loom. Users place colored circles in columns (the Y-axis) and rows (the X-axis).



There are several tools for placing wefts on the virtual loom. In each case you use the "tab" key or the mouse to move your cursor to the field for entering the coordinates, then you enter them, and then press the button for the shape tool. The point tool places a single weft:

x = 2	y = 2
-------	-------

The line tool places lines of wefts. You specify the two endpoints of the line. Diagonal lines tend to be jagged, but resizing the grid can help that (see "Options menu" on next page).

x1 = -3	y1 = -3
x2 = 3	y2 = 3

The rectangle tool fills in a rectangle of wefts. You specify two vertices (lower right and upper left). The rectangles of this tool are always aligned with the axes.

x1 = -3	y1 = -3
x2 = 3	y2 = 3


Part 2

The triangle tool fills in a triangle of wefts. You specify the three vertices.

x1 = -3	y1 = 1
x2 = 0	y2 = 4
x3 = 3	y3 = 1

The iterative triangle tool: Our first triangle tool made jagged edges, while traditional rug work has beautifully regular edges. The triangle iteration tool reflects the tradition of indigenous mathematics. For example, a triangle can be made by adding one bead on each side of the row, every three rows, as you go in the -Y direction.

Starting at x=0 y=0
 after every 3 rows
 add 1 to both ends
 for 9 rows in total
 +y
 in the -x +x direction.
 -y



- "Direction"—determines in which direction your rows will accumulate
- Starting at X, Y—that is the center of the starting row.
- "After every ___ rows"—lets you determine how many rows you go through before adding more wefts to the end.
- "Add ___ to both ends"—the number of wefts that will be added on each side of the center each time.
- "For ___ rows in total"—how many rows you will weft in this triangle.

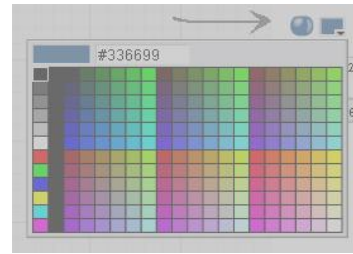
Note that this tool has two colors—some traditional basket work shifts color in each iteration. This allows you to select the starting color and ending color; the software does the shifting for you.

Part 3

There are also controls that apply to all the tools. "Clear" deletes everything. Normally "Create" is selected, so that your tools will fill their specified shape with wefts. "Remove" will erase all wefts in the specified shape, so if you make an error use "undo" not "remove." The color button allows you to select the weft color. Clicking on the little square in the upper right of the screen will give you a list of all the colors you have selected so far. The "Save" menu allows you to save the work on your hard drive and edit the design later. Make sure your file name is only letters, not spaces or numbers, and that you go back to the same computer when you want to edit your work.

The "Options" menu allows you to hide the grid, or create a title or notes about your design.

Printing: after you have your design completed, do a screen capture. In windows you can do that by pressing the "print screen" button on your keyboard, usually located at the upper right above the F10 key. On a Macintosh press shift + apple + 3 at the same time (also shift + apple +4 to select just a portion). That screen capture will save an image of the entire screen to your clipboard. You can then paste the clipboard image into a blank canvas in Word, Photoshop, Imaging (comes free in the "Accessories" folder in Windows) or other image editor.



Culturally Situated Design Tools Project Sample Rubric

Group Members Names: (up to 4)

Do you have?	Points Possible	Yes	No	Points Earned
Product				
Title with group members' names	5			
Information related to the culture (explanation in own words)	10			
Mathematical connections (explanation in own words)	10			
Demo of software	5			
Display of designs (explanation of how the design was created)	15			
Design is original	10			
Design has been edited with a photo editor	10			
Presentation				
All group members participate	10			
Present all required parts of project	15			
Answer questions from audience	10			
Extra Credit				
Project exhibits creativity above and beyond	Up to 10			
TOTAL:	100 + 10			

Instructional Days: 15-16

Topic Description: This lesson introduces the concept of a computer program within the context of a set of instructions for completing a common activity.

Objectives:

The student will be able to:

- Explain the concept of a computer program.

Outline of the Lesson:

- Following directions (55 minutes)
- Designing a program (15 minutes)
- Running a program (25 minutes)
- Being more precise with instructions (15 minutes)

Student Activities:

- Complete the Following Directions quiz.
- Complete the Drawing Pictures Activity.
- Write the instructions for making a peanut butter and jelly sandwich.

Teaching/Learning Strategies:

- Following directions
 - Distribute copies of Following Directions Quiz to each student face down in front of them. Each student should have a blank piece of paper and a pencil as well.
 - Give the students five minutes to do the quiz. Make note of how many students stand up and shout “hooray.”
 - Collect the papers when time has expired.
 - Point out that a perfect paper is one which has only the word “December” written in the top left corner. (The directions said to read all parts of the test before doing anything and step 14 says to only complete step #3.)
 - Give students about 10 minutes to complete the Drawing Pictures Activity.
 - Ask volunteers to show their pictures and explain why they drew the pictures as they did.
 - After the first volunteer, ask if someone drew it differently.
 - Ask the students what following directions has to do with computers. Prompt them as necessary that a computer follows a specific set of instructions called a computer program and must follow all of the directions precisely.
- Designing a program

- Ask the students to write down a set of instructions for a computer to make a peanut butter and jelly sandwich. Give them 5-10 minutes to write down these instructions.
- Collect the instructions.
- Running a program
 - Take out the bread, peanut butter, jelly, and knife and put them on your desk. Pick a set of instructions for making a sandwich (best to pick one which is not too detailed).
 - Read each instruction and carry it out—literally. For example, if the first instruction is “put the peanut butter on the bread,” take the jar of peanut butter and put it on the loaf of bread. If an instruction says to “spread the peanut butter on the bread,” use your fingers rather than a knife. If an instruction says to “cut the sandwich in half,” be creative and cut it between the two slices of bread. In other words, your goal is to show that instructions need to be very precise.
 - Repeat the process with another set of instructions.
 - Highlight the implicit knowledge that students bring to the task and how that has to be “unpacked” for the computer.
- Being more precise with instructions
 - Clearly, no matter how precise they tried to be, the instructions for making a peanut butter and jelly sandwich were open to interpretation. Ask the students to brainstorm how we could overcome this problem so that a computer could follow the instructions and make a perfect sandwich each time.
 - Guide the students toward the idea that we need a better “language” than English for describing the instructions. This is, in fact, the idea behind a computer program. There is a limited set of instructions which define very precisely what the computer does. For example, we can have a computer turn on a “dot” of a specific color in a specific location on the screen. By having the computer turn on many different dots in different colors, we can have the computer draw a picture. Note though that we don’t have an instruction for the computer to “draw a picture of a house” as that’s much too general and too open for interpretation.

Resources:

- <http://www.justriddlesandmore.com/direct.html>
The basis for the “following directions” quiz (the quiz was modified slightly.)
- Following Directions Quiz
- Drawing Pictures Activity
- Bread, peanut butter, jelly, and a knife.

Following Directions Quiz

Directions: You have a 5 minute time limit to complete the parts of this quiz. Carefully read all of the parts of the quiz before doing anything. In order to ensure the accuracy of this quiz, you should not use more than the allotted time of 5 minutes. Good Luck!!

You may begin now!!

1. Write today's date—month-day-year in the top right hand corner of your quiz paper.
2. Write the answer to the following multiplication problem directly underneath the date on your quiz paper— $6 \times 5 = ?$
3. Write the name of the month that begins with the letter "D" in the top left hand corner of your quiz paper.
4. Add 15 to the answer you got in part #2, and write this new total directly underneath your answer for part #3.
5. In the lower left hand corner of your quiz paper, write the names of your favorite singer and your favorite group.
6. Just above your answer to part #5, write "This quiz is very easy."
7. In the lower right hand corner of your quiz paper, draw a rectangle and inside the rectangle draw a five pointed star. The size of these drawings is not important.
8. Directly above your answer to part #7, draw a row of three small circles. Once again, size is not important.
9. Write the name of the first president of the United States on the back of your quiz paper anywhere you choose. If you don't know who this is, write your own name instead.
10. Write the name of any country that begins with the letter "I" directly underneath you answer to part #2.
11. Stand up, shout "hooray!", and sit down.
12. Take the number of dwarfs in the Snow White story and add it to the number of bears in the Goldilocks story. Divide by 2. Write this total in the approximate center of your quiz paper.
13. Think of a number between 1 and 50. Double that number. Add 20. Add 6. Subtract 17. Subtract 9. Divide by 2. Write this number on your quiz paper directly underneath your answer to part #11.
14. Now that you have carefully read all of the parts so far, and you have not carried out any of the actual work, skip the next 2 parts and go back and only complete part #3.
15. The name of the first president of the United States is George Washington. He was president from 1789 until 1797. Add the 2 dates together to see if the total is less than 5000.
16. You should not be reading the end of the exam before the beginning of the exam, but now that you are here you have just wasted some of the time you may need to complete the quiz.

Drawing Pictures Activity

1. Draw a picture of a house in the middle of the page.
2. Draw a picture of a stick figure father, mother and daughter.
3. Draw a picture of a mustang next to the house.
4. Draw a picture of the sun in the sky.

Instructional Days: 17-19

Topic Description: The question “What is intelligence?” is addressed through discussion of the differences between humans and computers. Various models of machine learning are investigated along with the concept of natural language understanding.

Objectives:

The student will be able to:

- Explain the idea of intelligence especially as it relates to computers.
- Explain what it means for a machine to “learn”.
- Discuss whether computers are intelligent or whether they only behave intelligently.

Outline of the Lesson:

- Journal Entry (15 minutes)
- Differentiation between humans and computers (95 minutes)
- A simple model of machine learning (55 minutes)

Student Activities:

- Complete journal entry.
- Complete CS Unplugged Activity 20: Conversations with Computers—The Turing Test.
- Interact with web-based chatterbots (Part I of The Computer Intelligence Activity).
- In groups, play several rounds of a guessing game (Part II of The Computer Intelligence Activity).

Teaching/Learning Strategies:

- Journal Entry: What is intelligence? Are computers intelligent? Why or why not?
 - Volunteers share their responses.
- Differentiating between humans and computers
 - CS Unplugged Activity 20: Conversations with Computers—The Turing Test
 - This activity can be downloaded from <http://csunplugged.com> From the menu, click on Activities, click on Turing Test, and then download the pdf for Activity 20. Note there are many additional resources listed that you may wish to explore.
 - It will be helpful for you to read through the entire activity before beginning it with your students. In addition to the explanation of the activity, it provides good background information that you will want to ensure is part of the discussion you have with students.
 - Based on the directions under “What to Do” (p. 214), assign and explain roles to 4 students.
 - Follow the remaining directions under “What to Do” (p.214-215).

- Have students complete Part I of Computer Intelligence Activity.
 - Assign each pair of students (students work with their elbow partner) two of the questions from the Turing Test Activity.
 - Discuss the results.
- A simple model of machine learning.
 - Have students complete Part II of the Computer Intelligence Activity.
 - Assign students to groups of 3 or 4 and assign each group 2 of the games in the activity.
 - Discuss the results.
- As part of the discussion make the distinction between a computer’s ability to perform some operations quickly and what it means to “know”.
 - Speed is not necessarily knowledge.
 - Humans are “smart”; computers are “fast” and follow code explicitly.
 - Intelligence is context dependent—e.g., being able to thrive in mountains versus downtown in a city. If you are familiar with a city, you can find streets easily, etc. whereas if you are a newcomer, you need to have a map and more precise instructions. This ties to the peanut butter and jelly sandwich example.

Resources:

- Computer Science Unplugged Activity 20: Conversations with Computers—The Turing Test (<http://www.csunplugged.com>), pp. 213-226
- Computer Science Unplugged Activity 20: Conversations with Computers—The Turing Test, p. 225—questions (one copy for each pair of students)
- Computer Science Unplugged Activity 20: Conversations with Computers—The Turing Test, p. 226—answers (one copy to post or display)
- Computer Intelligence Activity

Computer Intelligence Activity

Part I

A program passes [The Turing Test](http://en.wikipedia.org/wiki/Turing_Test) (en.wikipedia.org/wiki/Turing_Test) if a person can have a conversation with both it and a person and not be able to tell which one is the computer.

Try each of these chatterbots with the questions you were assigned. (Note: all of these websites were correct at the time of writing. You can also use a Google search to find these and others.)

1. Try to chat with [Eliza](http://nlp-addiction.com/eliza/) (http://nlp-addiction.com/eliza/). How realistic is she? Would she pass the Turing Test?
2. Try to chat with [Athena](http://Athena.blueinfos.com) (Athena.blueinfos.com). How realistic is she? Would she pass the Turing Test?
3. Try to chat with [Friend4U](http://virtualentities.com/friend4u) (virtualentities.com/friend4u). How realistic is she? Would she pass the Turing Test?
4. Try to chat with [InteliAvatar](http://inteliwise.com) (inteliwise.com). How realistic is she? Would she pass the Turing Test?
5. Which of the above chatterbots was the most like a real person?
6. What is the [Chatterbox Challenge](http://chatterboxchallenge.com) (chatterboxchallenge.com)?

Part II

1. Go to 20q.net. Choose your language (Think in American is recommended). Choose one of the games from the bottom that was assigned to your group. You are supposed to think of something in that category and answer the computer's questions by clicking them. The computer will try to guess what you chose in 20 questions or less. Play the game several times addressing each of the following:
 - Pick an item and see how many questions are required.
 - Choose the same item and see if you can make it require more questions
 - Repeat this with another item.
 - How intelligent is this? Would this pass the Turing Test?

Play the second game you were assigned and repeat the process above.

2. The Turing test is a person checking to see if it is talking to a computer. Can you think of any occasions that a computer might want to know if it is talking to *another* computer or a real life person?

Unit 2:

Problem Solving



Introduction

In order for students to become “computational thinkers” they need experience solving a wide range of problems and the opportunity to experiment with a variety of solution strategies. This unit begins with an introduction to the problem solving process. Students are asked to solve new problems by planning a strategy, designing and producing solutions, and then reflecting on their solutions and strategies.

Throughout the unit the emphasis should be on the process rather than the solution. Most of the world’s problems today do not have single simple solutions. In order to contribute effectively to the solution of these problems, students need to be comfortable in a collaborative environment where multiple approaches are valued and encouraged and where failure is seen as part of the process toward solution. Students must learn to think abstractly and apply known algorithms where appropriate, but also create new algorithms that can be applied to complex problems.

As students reflect on their solution processes and solutions and share those reflections with their peers, it is an opportunity to pull out instances where one strategy might be preferred over another and problems for which there are “standard” solutions versus those where there are many possible solutions.

Many of the problems presented have a mathematical basis and can serve to provide connections between mathematics and computer science. Common computer science topics such as searching, sorting, and graphing are introduced. Although programming the solutions to many of these problems is beyond the scope of this course, students will gain a basic understanding of the algorithms and be able to analyze them. In particular, it is important to emphasize that the models used for solving computational problems are the underpinnings of computer science and as such remain largely the same even as we add new tools and languages.

A key point of emphasis throughout the unit is the connection between the solution “process” and the discussion toward the end of Unit 1 related to how computers are programmed. It is also important to emphasize that not all problems are easily solved by computers.

Specific topics for each instructional day are listed in the overview chart on the next page.

Daily Overview Chart	
Instructional Day	Topic
1-2	Introduce data collection and problem solving.
3	Introduce the four steps of the problem solving process.
4-6	Apply the problem solving process. Use different strategies to plan and carry out the plan to solve several problems.
7-9	Reinforce the four steps of the problems solving process.
10-12	Count in the binary number system. Convert between binary and decimal numbers in the context of topics that are important to computer science.
13-14	Introduce the linear and binary search algorithms.
15-16	Explore sorted and unsorted lists and various sorting algorithms.
17	Introduce minimal spanning trees and how graphs can be used to help solve problems.
18-21	Final projects and presentations

Daily Lesson Plans

Instructional Days: 1-2

Topic Description: This lesson sets the stage for the unit. It provides an overview of data collection and problem solving that will be needed in order to complete the final project.

Objectives:

The student will be able to:

- Recognize various forms of communication as data exchange.
- Describe the implications of data exchange on social interactions.
- Consider privacy of data that they create.
- Explain the difference between data used for making a case and data that informs a discovery.
- Describe good research questions.

Outline of the Lesson:

- Communication Methods and Data Chart (15 minutes)
- Data Journal (15 minutes)
- How data is linked back to us (25 minutes)
- Solving Community Problems Activity (25 minutes)
- Difference between data used for making a case and data that informs discovery (15 minutes)
- Research questions (10 minutes)
- Journal Entry (5 minutes)

Student Activities:

- Discuss Communications Methods and Data Chart with elbow partner.
- Discuss Data Journal with elbow partner.
- Groups complete Solving Community Problems activity.
- Complete journal entry.

Teaching/Learning Strategies:

- Have students share their Communications Methods and Data Chart with their elbow partner.
 - Remind students that this was assigned as homework in Unit 1.
 - Have a few students share “What data is available?”, “Who has access to the data?”
- Data Journal Class Discussion
 - Have students:
 - Compare journals with elbow partners.
 - Write down 5 ways that they give off data.

- Take time to discuss what these data sources might tell us about ourselves if we “aggregated” or collected these data from lots of people – What good might be done? What services might be improved?
 - Think about which of these ways can be linked directly back to them. What are possible implications of the data being linked back to them?
 - Have students read this article about aggregate search data—technically, making search data available to researchers would help improve search engines, but it turns out that search history is intensely personal.
<http://www.nytimes.com/2006/08/09/technology/09aol.html>
 - The Netflix prize is another example. It has recently been cancelled due to FTC concerns over privacy. <http://blog.netflix.com/2010/03/this-is-neil-hunt-chief-product-officer.html>
 - Volunteering data on Facebook and other social networking sites might tell people more about you than you intend.
<http://www.nytimes.com/2010/03/17/technology/17privacy.html>
- Solving Community Problems
 - Present students with a scenario related to the local community. For example, the city council wants to find out about trash disposal in the community in order to clean up the streets.
 - Have students work in groups of 3-4 to outline how they would
 - Approach the problem
 - What kind of data they might need to collect
 - How they would collect and analyze the data
 - Lead a discussion to get at their thoughts. Highlight the differences between making a case and discovery. How would the choice between these determine the kinds of data you would collect?
 - Making a case (advocacy)—Use data to document situations that contribute to make a positive or negative case for something. (e.g., Let the Metro know about timing of trains and buses; tell the principal about something that needs to be done at the school; tell someone about something you’d like to see continued.
 - Discovery—Collect data to document situations and then use the data to learn something. (e.g., could your food choices be improved?; do I always take an efficient route to activities?)
 - What research questions might you ask in each case?
 - What is your research question?
 - Why did you choose to collect these data for this question?
 - What are the limits of this data?
 - What can you confidently say based on your data?
 - What perspectives are left out based on your data?
- Assignment (will be used for the Unit 2 final project)
 - Every day collect data related to where you go after school—location, means of transportation (walk, bike, etc.), how long it takes to get from one location to the next, any other data that you think would be interesting.
- Journal Entry: What are the steps you use to solve a problem?

- Ask students to reflect on whether these steps are the same in all types of problems they solve.

Resources:

- Communication Methods and Data Chart
- Data Journal

Instructional Day: 3**Topic Description:**

This lesson introduces the four main phases of the problem-solving process as defined by G. Polya in [How to Solve It](#).

Objectives:

The students will be able to:

- Name and explain the steps in the problem-solving process.
- Solve a problem by applying the problem-solving process.
- Explain what the word algorithm means.

Outline of the Lesson:

- Candy bar activity (25 minutes)
- Discussion of solutions (10 minutes)
- Introduction of the steps in the problem-solving process (15 minutes)
- Journal Entry (5 minutes)

Student Activities:

- In groups, participate in the candy bar activity.
- Participate in discussion of solutions.
- Reflect on the candy bar activity as it relates to the problem-solving process.
- Complete journal entry.

Teaching/Learning Strategies:

- Candy bar activity
 - Divide the students into groups of 2 or 3. Give each group a candy bar.
 - Explain that their task is to determine how many "breaks" it will take to break the candy bar into 12 equal pieces. One break of one piece of the candy bar will result in that one piece being divided into two pieces. Demonstrate a "break" by breaking the bar into two pieces. Then stack the two pieces together and break or cut the two pieces into four.
 - At this point, have each student write in their journal the number of breaks they think it will take to break the bar into 12 equal pieces. This should be done without talking to their partner or group members.
 - Working together with their partner or group, have the students discuss and then write their plan for solving the problem. They may revise their guess at this point.
 - Once this is completed, the students should implement the plan by opening the candy, breaking the candy, and counting the number of breaks it takes to get 12 equal pieces.
- Discussion of solutions
 - Choose a group to present their plan to the class.

- Sample questions to ask—Was your guess correct? What process did you use to come up with your guess? Did working with your group and creating your plan change your guess? How many breaks did it take (11 is the answer)? Did your plan work?
 - How do the steps they used match what they wrote in their journal?
- Introduction to the steps in the problem-solving process
 - How do the steps they used relate to the “formal” steps of the problem-solving process?
 1. Understand the problem—read or listen to the problem statement.
 2. Make a plan to solve the problem—use pictures, charts, graphs, systematic lists, objects, or act out the solution to help you devise a plan to solve the problem
 - In Computer Science we call this plan an **algorithm**.
 3. Carry out the plan—once the plan is conceived and understood, follow the plan. If you have planned well, this is the easy part.
 4. Review and reflect on how the problem was solved—once the problem is solved, reflect on the plan that was used.
- Extend breaking the candy into N pieces.
 - Post the Number of Pieces/Number of Breaks Chart (without solutions), including N and have students give you the # of breaks needed for each number of pieces.
 - As you go through the chart, ask questions that lead students to the following understandings.
 - One problem-solving strategy used in solving a problem is to solve a problem for specific values, find the pattern and then generalize the solution. In this case, they are generalizing the solution for an unknown positive number of pieces.
- Reflections on the candy bar problem: Ask the students to reflect on the candy bar problem. Why is this problem an important problem to solve for: a carpenter, a chef, a teacher?
- Journal Entry: How is solving this kind of problem the same/different from how you solve a problem in “real life”?
 - Discuss what makes a problem solvable by computer—being able to provide a step-by-step algorithm is one important piece, but context matters. Think back to unit 1 and making a peanut butter and jelly sandwich. Even if we refined our algorithm would a computer be able to make one? No, but a robot could. (Foreshadow Unit 6.)

Resources:

- Polya, G. How to Solve It. 2nd. Princeton, NJ: Princeton University Press, 2004.
Candy bar problem suggested by Dr. Manuel Blum, Carnegie Mellon University
- Candy bars for student groups to use
- Number of Pieces/Number of Breaks Chart

Number of Pieces/Number of Breaks Chart

Number of Pieces	Number of Breaks
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10
12	11
N	<i>N-1</i>

Instructional Days: 4-6**Topic Description:**

Students will apply different strategies to help them make a plan and carry out the plan to solve several problems. These strategies may include (but are not limited to): draw a diagram or picture, make systematic lists, divide and conquer, find the pattern, and guess and check.

Objectives:

The students will be able to:

- Name and explain the steps in the problem-solving process.
- Solve a problem by applying the problem-solving process.
- Express a solution using standard design tools.
- Determine if a given solution successfully solves a stated problem.

Outline of the Lesson:

- Handshake Activity and Fence Post Activity (20 minutes)
- Explanation of solutions (15 minutes)
- Handshake Activity and reflections (75 minutes)
- Presentations of Handshake Activity (40 minutes)
- Discussion of reflections (15 minutes)

Student Activities:

- Work individually on Handshake Activity problem #1 and the Fence Post Activity.
- Volunteers present solutions to problems.
- Work in groups to complete Handshake Activity problem #2.
- Groups give presentations of their problem solutions.
- Discuss reflections on the process.

Teaching/Learning Strategies:

- Handshake Activity problem #1 and Fence Post Activity
 - Students work individually on Handshake Activity problem #1 and the Fence Post Activity.
- Explanation of solutions
 - Have some students volunteer their solutions to the problems.
 - Reinforce each step of the problem-solving process by asking questions similar to those from the candy bar problem. You want students to understand that
 - Diagrams can be very useful in problems like this to look at a smaller version of the problem before trying to solve for N.

- The fencepost problem is a variation of the candy bar problem or the handshake problem.
- Handshake Activity problem #2 and Reflections
 - In groups of 3 or 4, have students discuss, plan, execute, and reflect on Handshake Activity problem #2. Students should follow the directions given in the activity document and write their group's thoughts on paper.
 - Encourage students to make drawings or charts and/or act out the solution. Chart paper can be given to students to display pictures, charts, or graphs. Their job is to explain the process and the solution so that everyone understands.
- Student Presentations
 - Each group should be given about 5-10 minutes (depending on the size of the class) to present their plan and solution to the class. Be sure the students show all 4 steps in the problem-solving process.
 - Students groups should explain their solutions—why they did what they did
- Discussion of reflections
 - Ask students questions that will get them to reflect on why they proceeded in the manner they did. Where did they start? (chart, etc.) What did they do next and why?
 - Is their solution complete enough that it could be given to a computer (if they knew the language the computer was using)? Why or why not?

Resources:

- Polya, G. How to Solve It. 2nd. Princeton, NJ: Princeton University Press, 2004.
- Handshake and Fencepost Activity
- Handshake Activity #2 Sample Solution

Handshake and Fencepost Activity

For each problem, complete the following information.

Understanding the problem:

- What data or information is known?
- What is unknown?
- What are the conditions?

Plan the solution: Show your plan for solving this problem.

Carry out the plan: Using your plan, show your work and your solution.

Review and discuss your solution: Reflect on your solution.

Complete problems #1 and #2 individually.

1. Handshake Problem #1: Assume there are 20 people in a room, including you. You must shake hands with everyone else in the room. How many hands will you shake? If there are N (where $N > 0$) people in the room, how many hands will you shake?
2. Fence Post Problem: You need to build one side of a fence that is 12 yards long. This fence will be built with fence posts and rails that connect one fence post to another. If each fence post is 1 yard away from the next fence post, how many fence posts will be needed for this side of the fence? How many fence posts will be needed for a side of a fence that is N (where $N > 0$) yards long?

Read and begin planning your solution for problems #3 and #4. These problems will be completed in class tomorrow with your group. Each group will present their solutions to the class.

3. Handshake Problem #2: Assume there are 10 people in a room, including you. Each person in the room must shake hands one time, and only time, with all the other people in the room. How many handshakes will occur? If there are 20 people in the room, how many handshakes will occur? If there are N (where $N > 0$) people in the room, how many handshakes will occur?
4. Reflections: Why are problems like these important to learn how to solve? How could this type of solution be of benefit to a carpenter, a chef, a teacher?

Handshake Activity #2 Sample Solution

The sample solution is only one possibility. Student groups may have a wide variety of strategies. Ask questions that probe their understanding of the steps of the problem-solving process they used.

Understanding the problem:

- What data or information is known? *There are 10 people or N people in the room.*
- What is unknown? *Total number of handshakes*
- What are the conditions? *Each person must shake hands only one time with all others in the room. All of the handshakes must be added together.*

Plan the solution: *A sample plan could be to describe the plan in words or use a chart or draw a picture and then act it out.*

Have the people line up in the room. The first person in the line walks down the line and shakes hands with all of the people in the line and then leaves the room. Count the number of handshakes and add to the total.

The next person in line walks down the line and shakes hands with all of the people left in the line and then leaves the room. Count the number of handshakes and add to the total.

This continues until there are only 2 people left. They shake hands and leave together. Increase the total by one.

Once the answer is known for 10 people, look for a pattern. Try the process for 5 people, 2 people. See if the pattern holds.

Carry out the plan: Using your plan, show your work and your solution.

Person	Shakes Hands with # of people left in line
A	9
B	8
C	7
D	6
E	5
F	4
G	3
H	2
I	1
J	0 (last person in line— no one left to shake hands with)

Now add up the number of handshakes: $9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 + 0 = 45$

For 10 people, the answer is the sum of the numbers from 1 to 9, which is 45. 9 is $10 - 1$.

For 5 people, the answer is the sum of the numbers from 1 to 4, which is 10. 4 is $5 - 1$.

For 2 people, the answer is the sum of the numbers from 1 to 1, which is 1. 1 is $2 - 1$.

For N people, the answer is the sum of the numbers from 1 to $(N-1)$.

Review and discuss your solution: Each person shakes hands with $N - 1$ other people. The answer is not $N(N-1)$, though, because each handshake counts as the one handshake for each person, but only one handshake for the total. The Hershey Bar problem helped to start the plan for this problem, but I needed to adjust the plan to only allow one handshake between each pair in the room.

So the 10 people make 9 handshakes each, but each handshake happens between 2 people, and can only be counted once. I could "divide" the handshake and let each person count the handshake as a $1/2$ handshake. So 10 people make 9 half-handshakes each = 45 handshakes.

N people make $N-1$ half-handshakes each = $N(N-1)/2$

The sum of the numbers from 1 to $N-1$ = $N(N-1)/2$

Instructional Days: 7-9**Topic Description:**

This lesson reinforces the four main phases in the problem-solving process.

Objectives:

The students will be able to:

- Solve a problem by applying the problem-solving process.
- Express a solution using standard design tools.
- Determine if a given solution successfully solves a stated problem.

Outline of the Lesson:

- Cultural background of cornrow braiding (15 minutes)
- Group discussion on cultural background of cornrow braiding (15 minutes)
- Cornrow curves design tool tutorial (80 minutes)
- Cornrow curves project (50 minutes)
- Gallery walk (5 minutes)

Student Activities:

- Work individually to review the history of cornrow braiding.
- Work in groups to answer reflection question and share with the remainder of the class.
- Work with elbow partner to complete the tutorial.
- Work individually to complete cornrow curves project.
- Participate in gallery walk.

Teaching/Learning Strategies:

- Cultural background of cornrow braiding
 - Students read the cultural background and how to braid sections (csdt.rpi.edu, Cornrow Curves).
- Group discussion on cultural background of cornrow braiding
 - Divide students into groups of 3-4 and ask each group to reflect on one of the following sections:
 - African Origins
 - Middle Passage
 - Civil War to Civil Rights
 - Hip Hop
 - Each group shares their response with the rest of the class.
- Cornrow curves design tool tutorial
 - Individual students complete Part I of the tutorial following all instructions and checking their work with their elbow partner.
 - Discuss any issues as a class before proceeding to Part II.
 - Complete Part II of the design tutorial.

- Stress mathematics and structured inquiry.
 - Reinforce concepts such as iteration, dilation, translation.
- Cornrow curves project
 - Each group of students should complete the following:
 - Students create their own design.
 - Describe each step of the problem-solving process used.
 - Highlight the mathematical concepts used and where and how they are used.
 - Reinforce the strategy of finding a similar problem that has already been solved to help solve the new problem.
- Gallery walk of designs
 - Students share their solutions.

Resources:

- Culturally Situated Design Tools Cornrow Curves—csdt.rpi.edu (courtesy Ron Eglash)

Instructional Days: 10-12

Topic Description: This lesson introduces the binary number system and how to count in binary. Students will learn how to convert between binary and decimal numbers in the context of topics that are important to computer science.

Objectives:

The students will be able to:

- Count forward and backward in binary.
- Explain why binary numbers are important in computer science.
- Use binary digits to encode and decode messages.

Outline of the Lesson:

- Journal Entry (5 minutes)
- CS Unplugged Activity 1: Count the Dots—Binary Numbers (counting in binary) (50 minutes)
- CS Unplugged Activity 1: Count the Dots—Binary Numbers (binary number system) (50 minutes)
- Revisit journal entry (5 minutes)
- Journal Entry (5 minutes)
- Discussion of why binary numbers are important in computer science (15 minutes)
- CS Unplugged Activity 1: Count the Dots—Binary Numbers (Email and Modems, Counting Higher than 31) (35 minutes)

Student Activities:

- Complete journal entry.
- Participate in the Count the Dots activities.
- Revisit journal entry.
- Complete journal entry.
- Participate in a discussion of why binary numbers are important in computer science.
- Complete Count the Dots activities.

Teaching/Learning Strategies:

- Journal Entry: How high can you count with your ten fingers?
- Use the CS Unplugged: Count the Dots activity to introduce binary representation and counting in binary.
 - Start with the introductory activity on p. 4 of the activity. (The activity can be downloaded from <http://csunplugged.com>) It will be helpful to read through the entire activity in advance, so that you can revise questions, add your own questions, and think about how you might want to structure each part of the activity. The goal is for students to be actively involved in some way and for all students to be able to represent numbers and count in binary. What follows is the minimal suggestion.

- Have 5 students come to the front of the room and demonstrate as you follow the instructions and ask the questions. (Each student should receive a large card with one of the numbers of dots—1, 2, 4, 8, 16)
- Use the CS Unplugged: Count the Dots activity to explain the binary number system and have the students practice counting forward and backward.
 - Complete the Binary Numbers activity on p. 5 and Working with Binary activity on p 7.
 - Have 5 students come to the front of the room and try counting as you call out the numbers. (Each student should receive a large card with one of the numbers of dots—1, 2, 4, 8, 16)
 - Have different groups of 5 students at a time come to the front and have the other students provide counting and representation challenges. You could also have a competition with multiple teams of students each trying to get the answer. There are many other possibilities. Be creative!!
- Revisit Journal Entry.
- Journal Entry: Complete the Sending Secret Messages activity on p. 8 of the CS Unplugged: Count the Dots activity. (Solution is on p. 13.)
- Discussion of why binary numbers are important in computer science
- Complete the remaining activities in CS Unplugged: Count the Dots. (Email and Modems—p. 9, Counting Higher than 31—p. 10, and/or More on Binary Numbers—p. 11)

Resources:

- Bell, Tim, Ian Witten and Mike Fellows. Computer Science Unplugged. Canterbury, New Zealand: 2002.
- Computer Science Unplugged Activity 1: Count the Dots—Binary Numbers, pp. 3-13
- Binary number cards for each student
- Large binary number cards for the demonstrations

Instructional Days: 13-14

Topic Description: This lesson introduces the linear and binary search algorithms.

Objectives:

The students will be able to:

- Describe the linear search algorithm.
- Describe the binary search algorithm.
- Explain conditions under which each search might be appropriate.

Outline of the Lesson:

- Tower Building Activity (55 minutes)
- Model tower building algorithm. (25 minutes)
- Model binary search (15 minutes)
- Comparison of linear and binary search (15 minutes)

Student Activities:

- In pairs complete the Tower Building Activity.
- Model the tower building algorithm.
- Students participate in the activity modeling binary search.

Teaching/Learning Strategies:

- Tower Building Activity
 - Have students complete the Tower Building Activity with their elbow partner and write their solutions in their journals.
- Model tower building activity.
 - Have students share their solutions with another elbow partner pair.
 - Have one set of students use 10 legos (or checkers or some other easily manipulated piece) to model the algorithm for solving the problem in front of the entire class.
 - Note: The solution is to start by taking half of the height of the tower and create that number of stacks of 2. Continue halving the number of stacks and doubling the height (plus one stack of any remainder) until the desired height is reached. This foreshadows binary search. (See sample solutions.)
- Model binary search.
 - Use 2 copies of the same dictionary. Hand one dictionary to 2 students and have them pick out a word in the dictionary.
 - Choose 2 other students to count the number of times you choose a word from the dictionary to search for the students' word.
 - Start by using a linear search. It should not take long for students to suggest that this is not a good strategy. Ask them to provide a better strategy.

- Guide them to binary search.
 - Discuss the number of guesses required and how this is similar to the tower building problem.
- Comparison of linear and binary search.
 - Linear—start at the beginning, look at each item until you find it or there is no more data. Data can be sorted or not.
 - Binary—look at middle item, eliminate the half where the value is not located. Find the new middle element and continue the process until you find it, or there is no more data. Ask students to describe what is necessary in order to use a binary search—the list must be sorted.
 - Have students provide examples of where each type of search is appropriate and why.
 - Note that decisions often need to be made about whether to maintain lists in sorted order, provide an option for sorting should it be necessary, etc. based on the types of searches that are expected to be performed on the data.

Resources:

- Shasha, Dennis. [The Puzzling Adventures of Doctor Ecco](#). Mineola, New York: Dover Publications, Inc., 1998.
- Tower Building Activity
- Sample Solutions for Tower Building Activity

Tower Building Activity

Donald Trump wants to build a 100 meter high tower as quickly as possible. He has unlimited resources and an unlimited budget and is willing to spend any amount to get the job done.

He has chosen to build the tower with blocks that are 100 meters long and 100 meters wide, but only 1 meter tall. The blocks interlock on top and bottom (like legos). They cannot be stacked sideways.

Using special lifters, putting one stack on top of another stack takes one week regardless of how high the stacks are.

What is the shortest amount of time that it will take to build the tower?

Suggestions:

- Use something like legos or a graph to help solve this problem.
- Start with a smaller tower of 5 or 10—solve a smaller problem.
- Extend that knowledge to the larger problem.

Sample Solutions for Tower Building Problem**5 meter tower**

Week #	# of stacks	# of blocks in stack	Remainder
1	2	2	1 stack of 1
2	1	4	1 stack of 1
3	1	5	

Three weeks needed for completion.

10 meter tower

Week #	# of stacks	# of blocks in stack	Remainder
1	5	2	
2	2	4	1 stack of 2
3	1	8	1 stack of 2
4	1	10	

Four weeks needed for completion.

100 meter tower

Week #	# of stacks	# of blocks in stack	Remainder
1	500	2	
2	250	4	
3	125	8	
4	62	16	1 stack of 8
5	31	32	1 stack of 8
6	15	64	1 stack of 40
7	7	128	1 stack of 104
8	3	256	1 stack of 232
9	1	512	1 stack of 448
10	1	1000	

Ten weeks needed for completion.

In general: The number of weeks is the smallest n such that the height of the tower is less than 2^n .

Instructional Days: 15-16

Topic Description: In this lesson the concept of a list (sorted and unsorted) and sorting algorithms will be explored.

Objectives:

The students will be able to:

- Define sorted and unsorted lists.
- Describe various sorting algorithms.
- Compare various sorting algorithms.

Outline of the Lesson:

- Journal Entry (15 minutes)
- CS Unplugged Activity 7: Lightest and Heaviest—Sorting Algorithms (explore sorting) (30 minutes)
- CS Unplugged Activity 7: Lightest and Heaviest—Sorting Algorithms (discover and describe sorting algorithms) (30 minutes)
- CS Unplugged Activity 7: Lightest and Heaviest—Sorting Algorithms (compare sorting algorithms) (35 minutes)

Student Activities:

- Complete journal entry.
- Groups participate in the various parts of the CS Unplugged: Lightest and Heaviest activity.

Teaching/Learning Strategies:

- Journal Entry: List examples of where it matters whether items are in order (sorted).
 - Have volunteers provide examples from their lists and explain why it matters that they are sorted; in other words, what are the consequences if the list is not sorted?
 - Ask students about the data that they have been collecting. How easy would it be for them to sort their data by hand? Does it get harder to do this with more data? Point out that this is one of the major advantages of computing—the ability to manage large sets of data that could not easily be managed by humans.
- CS Unplugged: Lightest and Heaviest activity
 - The activity can be downloaded from <http://csunplugged.com>. It will be helpful to read through the entire activity in advance, so that you can revise questions, add your own questions, and think about how you might want to structure each part of the activity. The goal is for students to be actively involved in some way and for all students to be able to describe the various types of sorting. What follows is the minimal suggestion.
 - Divide students into groups of 3-4 and give each group a set of weights and a balance scale as described in steps 1 and 2 on p. 66 of the Sorting Weights Activity. (There are many possible ways to make the weights. One would be to use bags with varying numbers of pieces of candy. If you don't have balance scales, you can help students come up with a strategy that will

- simulate a scale. For example, if you make the weights clearly different in weight, they could do this by feel.)
- Have students complete #3 and #4 on p. 66 and then discuss their answers as indicated.
 - Have students complete #5 on p. 66.
 - At this point in the activity, students should present their findings to the class and discuss. Point out the selection sort information on p.66.
 - Have students complete the Divide and Conquer activity on p.67. Throughout, guide students as necessary and have them keep track of the processes they use.
- If time permits, have students try both sorting methods to sort cards that have 50 random numbers on them and analyze the number of comparisons required for each.

Resources:

- Bell, Tim, Ian Witten and Mike Fellows. Computer Science Unplugged., New Zealand: 2002.
- Computer Science Unplugged Activity 7: Lightest and Heaviest—Sorting Algorithms, pp. 64-70
- Containers of the same size with different weights
- Balance scales

Instructional Day: 17

Topic Description: Minimal spanning trees and graphs will be explored. Students will learn how graphs can be used to help solve problems.

Objectives:

The students will be able to:

- Solve a minimal spanning tree.
- Draw a graph to solve a problem.

Outline of the Lesson:

- CS Unplugged Activity 9: The Muddy City—Minimal Spanning Trees (20 minutes)
- CS Unplugged Activity 9: The Muddy City—Minimal Spanning Trees (extension) (35 minutes)

Student Activities:

- Participate in the various parts of the CS Unplugged: The Muddy City activity.
- Participate in the various parts of the CS Unplugged: The Muddy City activity extension.

Teaching/Learning Strategies:

- CS Unplugged: The Muddy City activity
 - The activity can be downloaded from <http://csunplugged.com>. It will be helpful to read through the entire activity in advance, so that you can revise questions, add your own questions, and think about how you might want to structure each part of the activity. The goal is for students to be actively involved in some way and for all students to be able to describe shortest path strategies. What follows is the minimal suggestion.
 - Follow the directions in The Muddy City Problem on p. 78.
 - Have students work with their elbow partners.
 - Have students share their solutions and lead the follow-up discussion p.77.
- CS Unplugged: The Muddy City activity extension
 - Have students repeat the Muddy City Problem with the abstract representation in the figure on p. 79 and answer the questions on p 79.
 - Discuss various applications of this problem in anticipation of the final project (p.80).
 - Emphasize the idea of shortest path.

Resources:

- Bell, Tim, Ian Witten and Mike Fellows. Computer Science Unplugged. Canterbury, New Zealand: 2002.
- Computer Science Unplugged Activity 9: The Muddy City—Minimal Spanning Trees, pp. 76-80

Instructional Days: 18-21

Topic Description: Students work on final unit project.

Objectives:

The students will be able to:

- Incorporate all unit objectives into the final project.

Outline of the Lesson:

- Explanation of final project (15 minutes)
- Completion of final projects (150 minutes)
- Presentations of final projects (55 minutes)

Student Activities:

- Groups work on final projects.
- Groups present final projects.

Teaching/Learning Strategies:

- Complete Muddy City activity if necessary.
- Explanation of final project
 - Distribute final project explanation.
 - Note: You may wish to modify the scenario of the problem to address student interests and abilities. Another possible example might be a variation on finding the cheapest route between locations based on the price of gasoline. This could be in the context of a family vacation, carpool routes, running errands, etc.
 - Divide students into groups of 3-4.
- Completion of final projects
 - Monitor student work, answering questions as necessary.
- Presentations of final projects
 - Have each group present the information in their final project.
 - Groups respond to questions from other students and teacher.

Resources:

- Mapping website such as www.maps.google.com
- Final Project (This project is adapted from MathmaniaCS Lesson 13 (<http://www.mathmaniacs.org/lessons>)
- Final Project Sample Rubric

Final Project

For this project you will use the data you have been collecting about your activities after school. Each group member should determine the day on which they visited the most locations after school and what those locations were.

Scenario:

Assume that for one day you need to carpool with the other members of the group in order to get to all of the locations you each identified on the day you visited the most different locations. Determine the shortest route in terms of miles and then determine the shortest route in terms of time. Are these the same? Why or why not? What other conditions might you want to consider? (Your data may give you some ideas.)

You will need to use a map in order to calculate the distances. The data you collected has the times. Your presentation can be given as a poster, a PowerPoint, a video or other pre-approved product.

Your presentation should include:

1. The names of people in your group
2. A picture (graph) representing all the locations with all roads between them labeled with mileages and times
3. A detailed plan of your solution
4. A written explanation of the strategies you used to find the shortest route
 - The solution on the graph and the total number of miles
5. A written explanation of the strategies you used to find the least amount of time
 - The solution on the graph and the total time
6. Why the shortest routes are the same or why not
7. Other conditions you might want to consider and what data gave you the idea

Final Project Sample Rubric

Group Members Names: (up to 4)

Do you have?	Points Possible	Yes	No	Points Earned
Detailed Plans				
Overall plan to solve the problem (explanation of strategies)	10			
Data Journal for each group member	20			
Other parts of your project:				
Graph labeled with locations and mileage	10			
Graph labeled with times	5			
Solution labeled on graph	15			
Total mileage and total time	10			
Other conditions to consider	10			
Presentation				
All group members participate	5			
Present all parts of the project	10			
Answer questions from audience	5			
TOTAL:	100			

Unit 3: Web Design



Introduction

The Web Design unit builds on the concepts presented in the previous units by having students apply problem solving strategies to web design; thus, it also serves as a bridge to the Introduction to Programming unit as students move from user to creator. The unit also provides an opportunity to expand upon the issues of ethics and privacy related to the internet that were introduced in the first unit.

The basics of html and css are introduced as a method for describing features of web pages that students can use to design and develop web pages based on their own culture, interests and unique experiences.

The html and css lessons are scaffolded in order to provide all students an entry point, but it is likely that as students explore they will encounter features they wish to add for which they do not yet know the correct tags. Many students will be able to figure these out on their own and should be encouraged to do so.

Resist the temptation to provide lists of appropriate font and color palettes and/or best layout designs. As students experiment and share their work, challenge them to explain why they chose the features they did and encourage peers to comment.

Example projects are provided as a starting point, but students should be encouraged to work on projects that are authentic for them. They may choose to create web pages on different topics for each assigned project or build on a prior one as appropriate.

There are two supplements—Flash animation and Javascript—at the end of the unit for use if there is additional time and interest.

Specific topics for each instructional day are listed in the overview chart on the next page.

Daily Overview Chart	
Instructional Day	Topic
1-2	Explore issues of social responsibility in web use as well as the relative merits of the influence of the web on society, personal lives, and education.
3-4	Introduce the use of basic html.
5	Introduce basic formatting in html.
6-7	Explore image editing for the web using Photoshop or an image editor of choice.
8-10	Introduce basic css.
11-13	Explore the concept of separating style from structure by keeping separate html and css files.
14	Add hyperlinks to other websites.
15-16	Introduce a variety of page layout styles.
17-19	Practice the use of various design elements.
20-21	Introduce several different enhancements for website design, including web user interface elements combining Javascript, html, css, and Photoshop, accordion menus, lightbox and sliding images.
22-25	Final projects and gallery walk

Daily Lesson Plans

Instructional Days: 1-2

Topic Description: This lesson engages students in a discussion of the web as social experience. Issues of social responsibility in web use are explored as well as the relative merits of the influence of the web on society, personal lives, and education.

Objectives:

The student will be able to:

- Set up a blog.
- Explain basic security issues on the internet.
- Identify web applications which influence society and education.
- Identify appropriate vs. inappropriate use of social websites.

Outline of the Lesson:

- Set up a blog (25 minutes)
- Discussion of online security issues (20 minutes)
- Blog entry of students' online experiences (10 minutes)
- Discussion of parts 1-3 of *Growing Up Online* from the PBS series *Frontline* (40 minutes)
- Blog entry reflecting on *Growing Up Online* (15 minutes)

Student Activities:

- Set up a blog.
- Participate in discussion of online security.
- Create a blog entry on online experiences.
- Participate in a discussion about online experiences with social networking sites, blogs, email, online chatting and the kind of impact it has had on their lives.
- View and discuss *Growing Up Online*.
- Create a Blog entry reflecting on *Growing Up Online*.

Teaching/Learning Strategies:

- Set up a blog
 - Complete the portions of the setup that were not done prior to class.
 - Guide a discussion of online security as students work on setting up their blogs.
- Create a blog entry on online experiences.
 - Show students how to create a blog entry using the blogging tool chosen for the class.
 - Have students create an entry that describes some of their current online experiences.

- Guide a discussion regarding student use of social networking applications. (Note: This discussion may be a review of discussions from Unit 1.) Ask questions such as:
 - Which social networking applications do you use? (blogging, Facebook, MySpace)
 - How often? How many of your friends use them?
 - How important are these web applications to your lives? How have they changed your lives?
 - Living their lives essentially online
 - A revolution in classrooms and in social life
 - Self expression, trying on new Identities
- Display parts 1-3 of *Growing Up Online* from the PBS series *Frontline*.
 - After viewing the video, lead a discussion on the content.
- Blog entry reflecting on *Growing Up Online*
 - Have students create a blog entry reflecting on the video. Did any of their thoughts change after viewing the video?

Note: Helping each student set up a blog will require some time outside of class and should be completed in advance of the lesson. For example, students need to get a free Google gmail account before signing up for blogger. This can only be done outside of the LAUSD firewall.

Resources:

- <http://www.pbs.org/wgbh/pages/frontline/kidsonline/>

Suggested blogging tools:

- <http://www.blogger.com>
- <http://www.wordpress.com>
- <http://www.tumblr.com>

Instructional Days: 3-4**Topic Description:** An introduction to the use of basic html**Objectives:**

The student will be able to:

- Create a storyboard
- Navigate an html editor.
- Create an html page with a title and a body.
- Create an html page with paragraph tags, headings, line breaks, and horizontal lines.

Outline of the Lesson:

- Create a storyboard (15 minutes)
- Demo of html editor (15 minutes)
- Html page with a title and body (15 minutes)
- Html page with paragraphs and headings (25 minutes)
- Html page with line breaks and horizontal lines (40 minutes)

Student Activities:

- Create a storyboard for a webpage.
- Follow along during the demo of the html editor.
- Create an html page with a title and body.
- Create an html page with paragraphs and headings.
- Create an html page with line breaks and horizontal lines.

Teaching/Learning Strategies:

- Create a storyboard for a webpage
 - Explain to students that a storyboard is a visual plan for a website. It usually consists of a series of pages that include a rough sketch outlining the content, navigation, and design elements of the website.
 - Have students create a storyboard for a website that includes a title, several paragraphs, and headings for each paragraph before beginning the demo of the html editor.
- Demo of html editor
 - Display the html editor that you have chosen for the class. Point out the following html tags.

Tag	Description	End Tag
<html>	Defines an HTML document	</html>
<head>	Defines information about the document	</head>

<code><title></code>	Defines the title of the document	<code></title></code>
<code><body></code>	Defines the main part of the document	<code></body></code>

- Enter a title and a one sentence body. Demonstrate how to save the document as an html file and how to view the output page in a browser. Point out that the title appears in the bar at the top of the window. Also point out that the end tag is a necessary part of the syntax in order to tell the computer when to stop doing a particular thing.
- **Html page with a title and a body**
 - Have students write a paragraph in the body section and give it a title. (Students can choose to write about themselves or another topic of interest.)
 - Demonstrate the creation of a basic html/css page in the html editor of choice.
- **Html page with paragraphs and headings**
 - Have students add a second paragraph to their web page and note what happens.
 - Then have them add two lists related to their topic (favorite movies, music, hobbies, etc.) and note what happens.
 - Guide students to notice that everything runs together no matter how they type it.
 - Explain the following html tags.

Tag	Description	End Tag
<code><p></code>	Defines a paragraph	<code></p></code>
<code><h1></code> to <code><h6></code>	Defines headings at levels 1-6	<code></h1></code> to <code><h6></code>

- Have students try inserting these new tags into their web page and note what happens.
- Remind students that they need the end tag.
- This is a good place to point out that html is one language that can be used to give the computer instructions as discussed in Unit 1 and that the computer will produce exactly the output that the user indicates with the syntax provided. Html is not a programming language; it is a markup language.
- **Html page with line breaks and horizontal lines**
 - Explain the following html tags.

Tag	Description	End Tag
<code>
</code>	Defines a single line break	
<code><hr/></code>	Defines a horizontal line	

- Have students try inserting these new tags into their web pages and note what happens.
- Give students time to experiment and determine what combination of tags will allow them to put their lists in a column, with each list having its own heading.
- Point out that trying different tags and checking the output is an example of testing and verification. If the output is not what is intended, then they need to debug the code they wrote.

- Note that you can retrieve an html reference from <http://www.w3schools.com/html/>

Resources:

html editors

- <http://www.tacosw.com> (mac only)
- <http://www.barebones.com/products/TextWrangler/index.shtml> (mac only)
- <http://smultron.sourceforge.net/> (mac only)
- <http://www.alleycode.com/download.htm> (windows only)

html tutorial

- <http://www.w3schools.com/html/>

Instructional Day: 5**Topic Description:** An introduction to basic formatting in html**Objectives:**

The student will be able to:

- Create an html page that includes emphasized text.

Outline of the Lesson:

- Review of tags learned to date (5 minutes)
- Html pages that include emphasized text (50 minutes)

Student Activities:

- Participate in review of tags.
- Create an html page that includes emphasized text.

Teaching/Learning Strategies:

- Review of tags
 - Have students open their files; then lead a quick review of the tags.
- Html pages that include emphasized text
 - Explain the following html tags.

Tag	Description	End Tag
	Defines where you want text to be strongly emphasized (appears bold)	
	Defines where you want text to be emphasized (appears in italics)	

- Have students try inserting these new tags into their web pages and note what happens.
- Give students time to experiment.
- Ask students to think about what things they would like to be able to do with web pages that they have not done already. Answers will vary: pictures, different types of fonts, colors, etc.

Resources:

html tutorial

- <http://www.w3schools.com/html/>

Instructional Days: 6-7

Topic Description: Explore image editing for the web using Photoshop or an image editor of choice.

Objectives:

The student will be able to:

- Identify the standard image resolution for the web (72 dpi).
- Resize and crop images for the web.
- Identify and differentiate between the various image formats used in web sites: jpg, gif, png.
- Create an html page that includes images.

Outline of the Lesson:

- Discussion of various web image formats (5 minutes)
- Demo of resizing and cropping images (15 minutes)
- Selecting and cropping a images (35 minutes)
- Html pages that include images (55 minutes)

Student Activities:

- Participate in discussion and follow along with demo.
- Select and crop a few images.
- Create an html page that includes images.

Teaching/Learning Strategies:

- Discussion of various web image formats
 - Explain that image properties are relevant to web use.
 - It is important to check the size when preparing an image for use on the web. Resolution can be set under image size.
 - Remind students of the website evaluation they did in unit 1 and features that made a particular site user-friendly.
- Selecting and cropping an image
 - Demonstrate how to crop and resize images in Photoshop or image editor of choice. Part of this will be review from Unit 1.
 - Have students choose a few images that they will add to their web page and crop them.
 - Explain that students should save their images for use in this project and later projects.
- Html pages that include images
 - Explain the following html tag.

Tag	Description	End Tag
-----	-------------	---------

------	------------------	----

- Point out that the correct syntax for defining an image is ``
- xxxx is the name of the image file. The image should be in the same folder as the html file.
- Have students insert their image into their html page.
- They can resize the photo on the screen with: ``
- They can add a title by: ``
- Give students time to experiment with placement, sizes, headings, and additional images.
- For students who finish early, you can have them view the filters and effects section of <http://www.georgebenainous.com/web> and try modifying their images.

Resources:

- <http://www.georgebenainous.com/web> (photoshop—filters/effects)
- <http://morph.cs.st-andrews.ac.uk/>
- <https://www.photoshop.com/express>

Instructional Days: 8-10

Topic Description: An introduction to the use of basic css.

Objectives:

The student will be able to:

- Create inline styles with css.
- Create a web page that uses inline styles.
- Create an internal style sheet with css.
- Create a web page that uses an internal style sheet.

Outline of the Lesson:

- Overview of css (10 minutes)
- Sample inline styles (15 minutes)
- A web page that uses inline styles (15 minutes)
- Sample internal style sheet (15 minutes)
- Movie review html/css page (45 minutes)
- Share student work (10 minutes)
- A second html/css page (45 minutes)
- Share student work. (10 minutes)

Student Activities:

- Examine sample web content.
- Create a web page that uses inline styles.
- Examine sample web content.
- Complete movie review html/css project.
- Share completed projects.
- Complete second html/css project.
- Share completed projects.

Teaching/Learning Strategies:

- Overview of css
 - CSS stands for Cascading Style Sheets.
 - CSS provides the formatting and style for a web page, while html provides the content.
 - There are three methods for inserting styles.
 - Inline styles
 - Internal style sheet
 - External style sheet

- The basic format for a style is:

Selector	Declaration	Declaration
h1	{color:blue;	Font_size:12px;}

- The selector is the element you want to style; each declaration consists of a property and a value; the property is the attribute you want to change and each property has a value.
- To make it more readable you can put each declaration on a separate line.
- Demonstrate creating a header with the inline style listed above.
- Note that you can retrieve a css reference from <http://www.w3schools.com/css/>.
- Display http://www.w3schools.com/tags/ref_colornames.asp and http://www.w3schools.com/tags/ref_colorpicker.asp as sources for choosing colors.
- Have students suggest a few different declarations and demonstrate the results.
- Create a web page that uses inline styles.
 - Have students add a few styles to their web page.
- Sample internal style sheet.
 - Point out that inline styles should be used sparingly because they defeat the purpose of separating the style from the content.
 - Have students view <http://www.georgebenainous.com/web> (html/css--basic markup). They should view the page before and after the styling is added.
 - Point out what each piece of the styling does to the original page. Point out the format and that the internal style sheet is included in the <head>. Also note that the style applies to the entire page unless a specific inline style is added.
 - Talk about the fact that this is a way to do decomposition in their design process because they can choose the content and provide the style in two separate stages. They can also then test the various style elements one at a time to verify correctness. You may want to note that this is similar to the way you can think about writing the algorithm for making a peanut butter and jelly sandwich—write the instructions to work for any “bread”, any kind of peanut butter, and any kind of jelly; then the specific kinds of bread, peanut butter, and jelly can be changed according to particular taste. You could even extend this further to make it any kind of sandwich—bread, filling 1, filling 2, etc. (Note: This is an example of abstraction.)
- Movie review html/css page
 - Students create a website with one or more movie reviews. The html page will contain the following paragraphs for each review: title, director, synopsis, review. The css stylesheet will have corresponding classes. The page will also include:
 - At least one picture
 - The name of at least one of the actors in italics
 - The background and text colors
- Share student work.
 - Guide students in sharing their work either by a gallery walk, volunteers, etc.
- Complete second html/css project

- Note: you may choose to have students continue working on their movie project rather than start a second one if time is short.
- Some examples of projects from which to have students choose are provided in the tutorial or have students create their own project.
- Share results of student work.
 - Guide students in sharing their work either by a gallery walk, volunteers, etc.
 - Have students provide feedback to their peers.

Resources:

- <http://www.georgebenainous.com/web> (html/css--basic markup)
- http://www.w3schools.com/tags/ref_colornames.asp
- http://www.w3schools.com/tags/ref_colorpicker.asp

Instructional Days: 11-13

Topic Description: Explore the concept of separating style from structure by keeping separate html and css files.

Objectives:

The student will be able to

- Create an html page which links to a separate css file.
- Use html tags and css styling elements to separate style from structure.

Outline of the Lesson:

- Review of html/css concepts and description of how to link to a separate css file (15 minutes)
- Sample web content (20 minutes)
- Creation of separate html and css pages for the previous movie review project (40 minutes)
- Creation of separate html and css pages for a second project (75 minutes)
- Share student work (15 minutes)

Student Activities:

- Review html/css concepts.
- Examine sample web content.
- Complete html/css movie review project.
- Complete html/css project 2.
- Share completed projects.

Teaching/Learning Strategies:

- Review of html/css concepts
 - Guide a discussion of the highlights of the previous lesson.
 - Using the text editor, demonstrate how to create an external file for the styles. Emphasize that whatever the name of the style sheet, it needs to have a .css extension. Save the file in the web folder.
 - Note that this allows designers to further separate the style from the content. Such separation would be particularly useful in maintaining a large website that has a standard look and feel. That standard look and feel can be added to a page by just include the external css file. By utilizing reusable code, there is consistency and it means that the designer does not need to retype the entire content of the style every time a new page is added.
- Sample web content
 - Answer questions as students view and read the html/css reusable code section at <http://www.georgebenainous.com/web> (html/css—reusable code)
 - Demonstrate how to add the appropriate link to the html file.
- Complete html/css movie review project.

- Have students revise their previous movie review project to use an external css file.
- Complete html/css project 2.
 - One example of a project might be to create a website with information on their favorite band.
 - A paragraph with the name of the band in large bold type.
 - At least one picture
 - The genre of the band in italics (i.e. Rock, Rap, etc.)
 - A list of some of the songs from the band in a paragraph in regular type
 - A separate section that explains why the band is their favorite
 - The background and text in different colors
 - Other examples can be found in the tutorial
- Share student work.
 - Guide students in sharing their work either by a gallery walk, volunteers, etc.

Resources:

- <http://www.georgebenainous.com/web> (html/css—reusable code)

Instructional Day: 14

Topic Description: This lesson explores the use of links to other websites.

Objectives:

The student will be able to:

- Create an html page that includes hyperlinks.

Outline of the Lesson:

- Explanation of how to add hyperlinks (15 minutes)
- Addition of hyperlinks to web pages (40 minutes)

Student Activities:

- Participate in discussion of hyperlinks.
- Add hyperlinks to web page.

Teaching/Learning Strategies:

- Html pages that include hyperlinks
 - Explain the following html tag.

Tag	Description	End Tag
<code><a href = "url"</code>	Defines what is to be displayed.	<code></code>

- Point out that the correct syntax for defining a hyperlink is `Link text `
- The start tag contains information about the link address.
- What is to be displayed can be text, an image, etc.
- Give students time to experiment with adding hyperlinks to their previous project, including placement and sizes.

Resources:

- No additional resources needed

Instructional Days: 15-16

Topic Description: In this lesson a variety of page layout styles are introduced.

Objectives:

The student will be able to:

- Use table, row, and column tagging in an html page.
- Add css styling to an html table.
- Use ordered and unordered list tagging in an html page.
- Add css styling to an html list.
- Use grid elements in css div placement.
- Add a menu to an html page.
- Create a web page that includes layout styles.

Outline of the Lesson:

- Explanation of how to create an html table (15 minutes)
- Examples of data that lends itself to being presented in a table (5 minutes)
- Explanation of how to create html ordered and unordered lists and how to add styling to list elements. (15 minutes)
- Examples of data that lends itself to being presented in a list. (5 minutes)
- Preliminary css positioning and opacity exercise (15 minutes)
- Explanation of how to create a menu (15 minutes)
- Creation of a web page that includes layout styles (40 minutes)

Student Activities:

- Participate in the discussion of creating an html table.
- View examples of data that lends itself to being presented in a table.
- Participate in the discussion of creating ordered and unordered lists.
- View examples of data that lends itself to being presented in a list.
- Complete css positioning exercise.
- Participate in the discussion of menu creation.
- Create a web page that includes layout styles.

Teaching/Learning Strategies:

- Use the tutorial (<http://www.georgebenainous.com/web/html/css—tables>) to demonstrate how to create a table, how to add rows and columns and how to add css styling to table, row, and column elements. Have students view the example code and predict the results prior to viewing.
- Demonstrate examples of data that lends itself to being presented in a table

- Use the tutorial (<http://www.georgebenainous.com/web> html/css—styled lists) to demonstrate how to create ordered and unordered lists and how to add css styling to list elements. Have students view the example code and predict the results prior to viewing.
- Demonstrate examples of data that lends itself to being presented in a list.
 - Remind students of the data they worked with in units 1 and 2. Ask questions about how that might have been displayed on a web page. Point out that they may want to keep this in mind for the final project in Unit 5.
- Use the tutorial (<http://www.georgebenainous.com/web> html/css—page layout) to demonstrate div positioning using css. Have students view the example code and predict the results prior to viewing.
- Use the tutorial (<http://www.georgebenainous.com/web> html/css—menus) to demonstrate how to create a menu. Have students view the example code and predict the results prior to viewing.
- Have students create a web page which serves as an advertisement for a product of their choice.
 - Explain project requirements.
 - The page must make use of at least one of the page layout methods discussed.
 - Circulate room and help students with projects.

Resources:

- <http://www.georgebenainous.com/web> (html/css—tables, styled lists, page layout, menus)

html and css tutorials

- <http://www.w3schools.com/html/>
- <http://www.w3schools.com/css/default.asp>

Instructional Days: 17-19

Topic Description: Practice the use of various design elements.

Objectives:

The student will be able to:

- Create web pages which incorporate design elements previously studied.

Outline of the Lesson:

- Explanation of project (10 minutes)
- Design and creation of a web page that links to at least 5 other websites (135 minutes)
- Share student work. (20 minutes)

Student Activities:

- Design and create a 3 page website about their future that links to at least 5 other websites and includes a variety of design elements.
- Share completed work.

Teaching/Learning Strategies:

- Design and create a web page about their future that links to at least 5 other websites and includes a variety of design elements.
 - The three pages may either scroll or link to each other.
 - The project should include images related to their future.
- Share student work.

Resources:

- <http://www.georgebenainous.com/web> (html/css)

Instructional Days: 20-21**Topic Description:**

This lesson introduces a variety of enhancements for website development. Possible enhancements include: several web user interface elements combining Javascript, html, css, and Photoshop, accordion menus, lightbox, slideshow and jquery.

Objectives:

The student will be able to:

- Create a multi-page web site that includes a variety of enhancements.

Outline of the Lesson:

- Exploration of a variety of enhancement possibilities
- Creation of a multi-page website (55 minutes)
- Share student work. (10 minutes)

Student Activities:

- Review the sections of the tutorial website related to the enhancement possibilities.
- Create a multi-page website.
- Share completed work.

Teaching/Learning Strategies:

- Have students review the sections of the tutorial website dealing with
 - the creation of rollover buttons.
 - the lasso for inverted selections.
 - menus.
 - accordion menus.
 - lightbox.
 - sliding images.
- Creation of a multi-page website
 - Have students choose 2 or more of the enhancements to include in their website.
 - Students may add to the website about their future or create something new.
- Share student work.

Resources:

- <http://www.georgebenainous.com/web> (photoshop—filters/effects, javascript—rollover buttons, photoshop—project 2)

- <http://www.georgebenainous.com/web> (javascript—mootools)
- <http://www.georgebenainous.com/web> (javascript—light box, javascript—jquery)
- <https://www.photoshop.com/express/index.html>

Instructional Days: 22-25

Topic Description: Students complete final projects.

Objectives:

The students will be able to:

- Incorporate all objectives of the unit into the final project.

Outline of the Lesson:

- Explanation of final project (15 minutes)
- Final project (135 minutes)
- Gallery walk and vote on final projects (15 minutes)

Student Activities:

- Complete final project.
- Participate in gallery walk to view and vote on completed projects.

Teaching/Learning Strategies:

- Final project
 - Explain final project choices.
 - Help students with projects as necessary.
- Gallery walk
 - Encourage students to ask each other questions as they view the websites.
 - Have students vote on their favorite.

Resources:

- Final Project
- Final Project Sample Rubric

Final Project

Your task is to create a website that includes

- Images and text with references to sources
- Pages with headers, navigation and content
- An external css file to define layout and styling

You may choose any of the following topics

- An ethical dilemma
- A career
- A worldwide or community problem
- A topic of your choice that has been approved

Ethical Dilemma Web Site

Your task is to analyze an ethical dilemma. Choose one of the four dilemmas listed below or get approval for a different one. You must consider the alternatives and give reasons for the why and the why not you should do what is described. Then you must choose what you would do and explain why. The website should include pages that

1. Describe the dilemma you have chosen.
2. Give 3 reasons why you should do what is described.
3. Give 3 reasons why you should NOT do what is described.
4. Describes what you will do and explains why.

Ethical Dilemmas:

1. People illegally download music over the internet. Although it's free, it is still illegal. What do you choose to do? Why?
2. Your parent loses his/her job. You could help out by selling illegal dvds on the streets. What should you do?
3. You have the ability to hack into the school computer system. You can change people's grades. Would you change your own? Why or why not? What if you could change the grade for a basketball player who has a scholarship to play for a big university?
4. Someone you know works at a store that sells iPods. He steals some and asks if you want to buy one for half the price the store sells it for? Should you buy it? Why or why not?

Career Website

Research a career and create a website that provides information about it.

The website should include pages that

- Provide a brief description of the career
- Explain the education required
- Describe tasks performed in the career, salaries and how computer science is used in the career.

Worldwide or Community Problem Website

Research a worldwide or community problem and create a website that provides information about it.

The website should include pages that

- Provide a brief description of the problem.
- Explain how the problem is affecting people.
- Describe possible solutions to the problem and what people reading the website can do to help solve it.

Final Project Sample Rubric:

Do you have?	Points Possible	Yes	No	Points Earned
Website Content				
A home page with an image and a brief description of your topic	5			
3 or more additional pages on your site	15			
Images that support your topic	10			
Cite the source(s) of you images	5			
Complete information for your topic	10			
Cite the source(s) of your information	5			
Website Design				
Have a background color or image.	5			
Incorporate one of the layout styles into your website	5			
Include rollover images	5			
Links to all the pages of your website on each page.	10			
Integrate the lightbox slideshow, sliding images, or mootools accordion into your website.	5			
Use one shared external .css file for your site.	5			
Peer Grading	15			
Your project is voted best by your peers. (EXTRA CREDIT)	up to 10			
Total	100			

Flash Animation Supplement

(These activities can be used as the last days prior to the final project if students finish other projects prior to the time allotted.)

Instructional Day: 1

Topic Description: Adobe Flash (formerly Macromedia Flash) is a proprietary web animation platform. The introductory lesson demonstrates how to use stop action photography and Flash to create a flipbook effect.

Objectives:

The student will be able to:

- Use stop action photography in animated flip books.
- Create a simple flash animation by importing a series of images.

Outline of the Lesson:

- Preview the stop action photography study of the galloping horse (5 minutes)
- Demonstration of how to clip each image in Photoshop (5 minutes)
- Demonstration of how to import a series of images into Flash and how to play the movie (10 minutes)
- Practice of the import procedure (5 minutes)
- Creation of a movie from stop action photography (25 minutes)
- Share student work (5 minutes)

Student Activities:

- Preview the stop action photography study of the galloping horse from the Flash section of the tutorial website.
- View how to clip each image in Photoshop.
- View how to import a series of images into Flash and how to play the movie.
- Create a movie.
- Share completed work.

Teaching/Learning Strategies:

- Preview of the stop action photography study of the galloping horse from the Flash section of the tutorial website
 - Discuss the historical significance of Eadweard Muybridge and stop action photography.
 - Preview various Eadweard Muybridge photographic studies.
- Demonstration of how to clip each image in Photoshop
 - Create eleven separate images.
 - Follow a numerical naming convention: 01.jpg, 02.jpg...11.jpg.
- Demonstration of how to import a series of images into Flash and how to play the movie

- Guide students as they follow the procedure after it is demonstrated.
- Creation of a movie
 - Suggest students download another Eadweard Muybridge stop action photographic study and follow the same procedure or have them photograph their own stop action study.
 - Circulate room and help students choose and complete projects.
- Share student work

Resources:

- <http://www.georgebenainous.com/web> (flash—flipbook)
- <http://www.adobe.com/cfusion/designcenter/search.cfm?product=Flash&go=Go>

Instructional Day: 2

Topic Description: An animation technique called tweening is explored in Adobe Flash.

Objectives:

The student will be able to:

- Use frame/timeline based animation.
- Use an automatic frame based animation technique called tweening.
- Create several examples of tweened animations.

Outline of the Lesson:

- Demonstration of tweening techniques (15 minutes)
- Creation of a visual composition (30 minutes)
- Share student work (10 minutes)

Student Activities:

- View tweening techniques (outlined in the tutorial website).
- Create a visual composition.
- Share completed work.

Teaching/Learning Strategies:

- Demonstration of tweening techniques (outlined in the tutorial website)
 - motion, size, rotation, color
 - Discuss elements of design as they pertain to objects in motion.
- Creation of a visual composition
 - Explain the requirements for one or more of the following sample projects and guide students as they create their versions.
 - Create a visual composition using Flash tweening. (Remember to put each tween on a separate layer). Study the ideas of symmetry (balance) and asymmetry (imbalance) in motion.
 - Create a visual composition using Flash tweening applied to initials. Students can use their own initials. (Type can be tweened in Flash.)
- Share student work

Resources:

- <http://www.georgebenainous.com/web> (flash—tweening)
- <http://www.adobe.com/cfusion/designcenter/search.cfm?product=Flash&go=Go>

Instructional Day: 3

Topic Description: The movie clip is the basic unit of Flash animation which allows for reusability and scripting. This lesson is an introduction in the creation of movie clips.

Objectives:

The student will be able to:

- Explain the concept of a Flash movie clip.
- Differentiate between a movie clip and an instance of a movie clip.
- Create a movie clip based on keyframed animation.

Outline of the Lesson:

- Demonstration of creating a movie clip (10 minutes)
- Demonstration of how to reuse multiple instances of a movie (5 minutes)
- Creation of a horse movie clip (20 minutes)
- Creation and implementation of movie clips (20 minutes)

Student Activities:

- View examples of Flash movie clips.
- Create various Flash movie clips based on the galloping horse study as outlined in the Flash section of the tutorial website.
- Design and create follow-up movie clips.
- View the creation of a movie clip.
- View how to reuse multiple instances of a movie clip.
- Create the horse movie clip as outlined in the tutorial website.
- Create and implement movie clips.

Teaching/Learning Strategies:

- Demonstration of examples of Flash movie clips
 - Explain how to create a keyframed animation based on the galloping horse example from the Flash section of the tutorial website.
 - Guide students as they create various Flash movie clips based on the galloping horse study as outlined in the Flash section of the tutorial website.
 - Guide students as they design and create follow-up movie clips.
- Demonstration of creating a movie clip.
- Demonstration of how to reuse multiple instances of a movie clip
 - including a secondary tweening
- Creation of a horse movie clip as outlined in the tutorial website
- Creation and implementation of movie clips

- Guide students as they create their own ideas and then implement.

Resources:

- <http://www.georgebenainous.com/web> (flash—movie clips)
- <http://www.adobe.com/cfusion/designcenter/search.cfm?product=Flash&go=Go>

Javascript Supplement

Instructional Day: 1

Topic Description: Introduce basic Javascript; add interactivity to web pages.

Objectives:

The student will be able to:

- Add a Javascript to an html page.
- Link to an external Javascript file.
- Create alerts and prompts in Javascript.
- Write basic math statements in Javascript.

Outline of the Lesson:

- Demonstration of Javascript basic markup code samples (5 minutes)
- Creation of Javascript's (10 minutes)
- Extension of code samples provided in tutorial website (40 minutes)

Student Activities:

- View code samples from tutorial website.
- Create initial Javascript's.
- Extend code samples provided in tutorial website.

Teaching/Learning Strategies:

- Demonstrate how to create and link to a Javascript file.
- Guide students in the creation of initial Javascript files.
- Extension of code samples provided in tutorial website
 - Demonstrate how to extend the code samples.
 - Students extend the code samples.

Resources:

- <http://www.georgebenainous.com/web> (javascript—basic scripting)

Javascript tutorial

- <http://www.w3schools.com/JS/default.asp>

Instructional Day: 2

Topic Description: Introduce Javascript functions. Create modular, reusable code and use Javascript to learn fundamental programming concepts.

Objectives:

The student will be able to:

- Use the correct syntax rules for creating functions in Javascript.
- Create Javascript math functions.
- Create Javascript functions which apply css styling to a div.

Outline of the Lesson:

- Demonstration of Javascript function code samples (10 minutes)
- Creation of a Javascript function (10 minutes)
- Extension of code samples provided in tutorial website (35 minutes)

Student Activities:

- View Javascript code samples from tutorial website.
- Create a simple Javascript function.
- Extend code samples provided in tutorial website and create math functions.

Teaching/Learning Strategies:

- Demonstrate how to create a Javascript function.
- Guide students in the creation of initial Javascript functions.
- Extension of code samples provided in tutorial website
 - Demonstrate how to extend the code samples provided in the tutorial website
 - Students extend the code samples to create their own math functions a

Resources:

- <http://www.georgebenainous.com/web> (javascript—functions)

Javascript tutorial

- <http://www.w3schools.com/JS/default.asp>

Unit 4:

Introduction to Programming



Introduction

Programming is one of the creative processes that can transform ideas into reality. The intention of this unit is to highlight what can be created by using programming as a tool. As with the previous unit, students will create projects that reflect the diversity of interests in the classroom and that are personal to individual students.

Scratch provides an environment that lends itself to “tinkering”. The drag and drop nature of the blocks moves the focus away from messy syntax and allows for making modifications quickly. As students work through the unit, they should be encouraged to reflect on their tinkering and the thought processes that go into it. They should engage in discussions of why a particular set of instructions didn’t work the way they thought they would and in discussions of “what if” scenarios. It is through these discussions that you can help students connect mathematics and logic to computation in programs and highlight the various abstractions they are using in creating their projects. It is also through these conversations that programming constructs such as conditionals, iteration and looping can be highlighted.

The projects listed are examples only and represent a minimal set of activities. When students complete the assigned projects, they should expand their work by adding features, collaborating with other students, and adding more personalization. There are many more projects on the Scratch website (<http://sctrch.mit.edu>). The website is also a vibrant community. Students should be encouraged to become part of the community where they can collaborate beyond their classroom and get additional ideas for projects. There is also a Scratch Ed community for teachers (<http://scratched@media.mit.edu>).

Specific topics for each instructional day are listed in the overview chart on the next page.

Daily Overview Chart	
Instructional Day	Topic
1	Introduce the Scratch programming language, including the basic terms utilized in the language.
2-3	Practice using the basic features of Scratch in the context of creating a simple program.
4	Create a dialogue between two sprites.
5-6	Introduce the methods of moving sprites in Scratch.
7-8	Practice the concept of event driven programming through the creation of an alphabet game.
9	Introduce the concept of broadcasting via role play.
10-13	Write Scratch stories and present them to the class. Peer reviews are conducted.
14	Introduce the concept of variable.
15	Introduce the concept of conditionals.
16-17	Introduce And, Or and randomness.
18	Apply knowledge of conditionals to develop a Rock Paper Scissors program in Scratch.
19	Build on previous programming concepts to create a timer.
20-23	Create a timing game in Scratch and present it to the class. Conduct peer reviews.
24	Investigate two types of games that may provide ideas for the final project.
25	Explain final project and the rubric for the final project.
26-28	Write Scratch programs for either My Community or Game project. Conduct peer reviews.
29	Complete final projects.
30	Presentations of final projects

Daily Lesson Plans

Instructional Day: 1

Topic Description: This lesson introduces the Scratch programming language, including the basic terms utilized in the language.

Objectives:

The students will be able to:

- Name the basic terms used in Scratch.
- Create the beginning of a simple program in Scratch.

Outline of the Lesson:

- Journal Entry (5 minutes)
- KWL chart about programming/Scratch (15 minutes)
- Scratch introductory video (10 minutes)
- Model of how to start name assignment (25 minutes)

Student Activities:

- Complete journal entry.
- Complete KWL chart about programming/Scratch.
- Groups take turns sharing out their K's and W's orally.
- Watch Scratch introductory video.
- Follow along with Scratch open as teacher models how to start name assignment.

Teaching/Learning Strategies

- Journal Entry: How do you think programs like Microsoft Word, Internet Explorer and Windows are made?
 - Discuss what it means to program a computer. Remind students that in the previous unit they used a markup language to provide instructions to the computer on the layout and content of web pages. Programming languages are used to translate algorithms into a language that a computer can execute.
- KWL chart
 - Students meet with groups and each group completes a KWL chart. (Know, Want to Learn, Learned)
 - Groups take turns sharing out their K's and W's orally. Encourage them not to repeat anything that has already been said.
 - Put KWL charts up in the classroom; tell students that they will finish the L towards the end of the unit.
- Scratch introductory video
 - Played with sound. Can be played over a projector.
- Model of how to start name assignment
 - Address how sound will be handled in the classroom.

- Scratch lends itself to playing sounds so it can get noisy. The teacher needs to decide how to address this. Headsets with microphones are one solution.
- Build a name project similar to name.sb.
- Emphasize
 - Every character in Scratch is called a Sprite.
 - Although Scratch is programming, it is not used in industry. Point out a few languages that are used in industry—Java, C, C++. Throughout the unit, you will want to reinforce that the basic constructs used in Scratch are also used in “industrial strength” languages.
 - How to choose a Sprite from a file
 - How to paint your own sprite
 - Each sprite has its own scripts.
 - You can right click any block and select help to get more information on how to use it.
 - How to change the language in Scratch (for your English Learners)
 - How to go to full screen mode and back
 - How to switch back and forth between sprites by clicking on them
 - X and Y coordinates on the screen are shown on the bottom right below the stage
 - How to save in the proper location (the default is to save in the Scratch Projects folder (C:\\Program Files\\Scratch\\Projects))
 - The following blocks should be modeled:
 - Move _ steps
 - If on edge, bounce
 - Turn _ degrees
 - Forever
 - Change color effect by _
 - When the green flag is clicked
 - Encourage students to experiment. They can’t break the computer by dragging the wrong block.
 - Show students where they can access ScratchGettingStarted.pdf. (It would probably be useful to have printed copies for each student.)
 - Show students Name Sample Rubric.

Resources:

- KWL Graphic Organizer Chart.pdf (UCLA SMP)
- ScratchIntro.wmv (scratch.mit.edu)
- ScratchGettingStarted.pdf (scratch.mit.edu)
- name.sb
- Name Sample Rubric
- <http://scratch.mit.edu>

Name Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
Have a separate sprite for each letter of your name.	5			
Have at least 3 different interesting behaviors for the letters in your name.	5			
All the letters have a behavior	4			
Use the “when green flag clicked” block	3			
Use the “forever” block	3			
Extra Credit				
Have your name reinitialize itself when the green flag is clicked. In other words, all the letters will start off in the right location facing the correct way.	2			
TOTAL:	20			

Instructional Days: 2-3

Topic Description: This lesson provides students an opportunity to practice using the features of Scratch outlined on Day 1 in the context of creating a simple program.

Objectives:

The students will be able to:

- Complete a simple Scratch program.
- Utilize the green flag feature.

Outline of the Lesson:

- Journal Entry (5 minutes)
- Class discussion of journal entry (15 minutes)
- Name programs (90 minutes)

Student Activities:

- Complete journal entry.
- Share journal entry responses with the entire class.
- Write programs based on their own names.

Teaching/Learning Strategies:

- Journal Entry: What do you remember about Scratch from yesterday? What do some of the blocks do?
- Class discussion of journal entry
 - Allow students to share their responses.
 - In the process, make sure to review concepts needed to finish the name project.
 - Review rubric for name project.
 - Tell students that they will do a gallery walk of the projects at the beginning of tomorrow.
- Name programs
 - Students write programs based on their own names.
 - Teacher circulates room checking progress and answering questions.
 - Before time is up, remind students to save their work.
 - Remind students that Scratch is free to download at scratch.mit.edu.

Resources:

- ScratchGettingStarted.pdf (scratch.mit.edu)
- name.sb
- Name Sample Rubric
- <http://scratch.mit.edu>

Instructional Day: 4

Topic Description: This lesson describes how to create a dialogue between two sprites by first creating a written dialogue.

Objectives:

The students will be able to:

- Develop a dialogue between two or more Scratch sprites.
- Explain the reasoning behind how their dialogue works.

Outline of the Lesson:

- Gallery walk of name projects (10 minutes)
- Assignment introduction (5 minutes)
- Develop dialogue (30 minutes)
- Student presentations (10 minutes)

Student Activities:

- Participate in a gallery walk of name projects.
- Participate and listen to assignment introduction.
- Develop a dialogue.
- Present dialogues.

Teaching/Learning Strategies:

- Gallery walk of name projects
 - Facilitate gallery walk by giving the students an order or pattern to follow in walking around the room (dependent on lab).
- Assignment introduction
 - Tell students that they'll be making a dialogue between two or more sprites.
 - Have a sample dialogue with a student; for example,
 - Teacher: Knock-Knock
 - Student: Who's there?
 - Teacher: Juana
 - Student: Juana who?
 - Teacher: Juana go write a program in Scratch!
 - Student: Ha ha!
 - Make a sample program using only "say _ for _ secs" blocks.
 - Ask students what was the difference between the live dialogue and the sample program? (Answer is that in the program they are talking at the same time.)
 - Have students help you find the "wait _ sec" block. Add a few so students can see the sprites taking turns.
 - Show students Dialogue Sample Rubric and tell them that they can create their own dialogue. They can do their own knock-knock joke, or they can use their creativity.
- Develop Dialogue
 - Circulate room and help students.

- Student presentations
 - Have students volunteer to present their dialogues for the entire class.

Resources:

- Dialogue Sample Rubric

Dialogue Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
Have 2 or more sprites talking in dialogue.	4			
Have 3 or more sprites talking in dialogue.	5			
All the sprites are polite and they take turns talking	4			
Each sprite says at least 3 things.	3			
The conversation starts “when green flag clicked”	4			
Extra Credit				
Have 4 or more sprites talking in dialogue	2			
TOTAL:	20			

Instructional Days: 5-6

Topic Description: This lesson describes the methods of moving Sprites in Scratch.

Objectives:

The students will be able to:

- Explain the 3 major ways to move sprites.
- Choose the appropriate method of moving to make a cat circle the bases.

Outline of the Lesson:

- Journal Entry (5 minutes)
- moving.sb (20 minutes)
- Discussion of answers to questions (15 minutes)
- baseball.sb (70 minutes)

Student Activities:

- Complete journal entry.
- Examine moving.sb.
- Discuss answers to questions.
- Complete baseball.sb.

Teaching/Learning Strategies:

- Journal Entry: Briefly describe how you would graph in your Algebra class (The x-y coordinate plane, etc)
 - Have students work individually and then share with their elbow partners.
- moving.sb
 - Circulate the room and help students answer the questions.
- Discussion of answers to questions
 - Emphasize that the “repeat” block will do whatever is inside it n times. This behavior can be called iteration or looping.
 - Point out that iteration is a construct that is used in other programming languages.
 - Emphasize the differences between the 3 ways to move.
 - Emphasize how the sprites will reinitialize themselves when the green flag is clicked.
- baseball.sb
 - Circulate the room and help students finish baseball.sb.
 - After a student can get the cat around the bases, encourage them to use the “point in direction” block to get the cat to turn the correct way when running.
 - If students need a hint for the extra credit, show them the “next costume” and “switch to costume” blocks under the “Looks” tab.




Resources:

- Moving Project
- Moving Project Solutions
- moving.sb
- baseball.sb

- baseball solution.sb

Moving Project

There are basically 3 ways to move sprites in Scratch. Try the file moving.sb and answer the questions below:

1. Click the green flag. What do the three animals do?
 2. Look at the scripts for each of the 3 sprites. What 3 blocks do all three sprites use?
 3. What blocks does the cat use to move?
 4. What block does the dog use to move?
 5. What block does the monkey use to move?
6. Describe in your own words how the move block works.
- 
7. Describe in your own words how the go to xy block works.
- 
8. Describe in your own words how the glide block works.
- 
9. Some of the blocks require x: and y: coordinates. Place the mouse over the white window and look at the mouse x: and mouse y: numbers underneath the bottom. How are the x: and y: coordinates determined in Scratch?
 10. Use what you've learned about moving to get the cat to run the bases (as realistically as possible – bases are run counter clockwise) in baseball.sb. Make sure that when you click the green flag, the cat starts at home plate again.
 11. Extra Credit: Make the cat change costumes so that it looks like it is running as it circles the bases.

Moving Project Solutions

There are basically 3 ways to move sprites in Scratch. Try the file moving.sb and answer the questions below:

12. Click the green flag. What do the three animals do?

They move across the screen.

13. Look at the scripts for each of the 3 sprites. What 3 blocks do all three sprites use?

When green flag clicked, go to x:_ y:_, and wait _ sec

14. What blocks does the cat use to move?

Repeat _ and move _ steps

15. What block does the dog use to move?

go to x:_ y:_

16. What block does the monkey use to move?

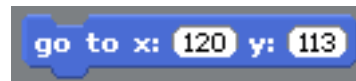
glide

17. Describe in your own words how the move block works.



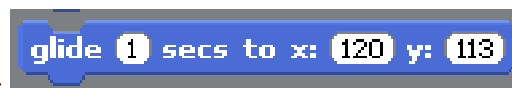
Move the sprite n steps. If n is positive, the direction will be to the right.

18. Describe in your own words how the go to xy block works.



Move immediately to that (x,y) position.

19. Describe in your own words how the glide block works.



Take n seconds to move from my current position to (x,y).

20. Some of the blocks require x: and y: coordinates. Place the mouse over the white window and look at the mouse x: and mouse y: numbers underneath the bottom. How are the x: and y: coordinates determined in Scratch?

It is just like the 2 dimensional x y graphs from Algebra. (0,0) is in the exact middle of the stage.

21. Use what you've learned about moving to get the cat to run the bases (as realistically as possible – bases are run counter clockwise) in baseball.sb. Make sure that when you click the green flag, the cat starts at home plate again.

22. Extra Credit: Make the cat change costumes so that it looks like it is running as it circles the bases.

Instructional Days: 7-8

Topic Description: This lesson introduces the concept of event driven programming and provides practice through the creation of an alphabet learning game.

Objectives:

The students will be able to:

- Explain event driven programming.
- Write a program that responds to user created events from the mouse and keyboard.

Outline of the Lesson:

- Presentation of solution for baseball extra credit (10 minutes)
- Journal Entry (10 minutes)
- Event lecture/description of Alphabet Learning Game (20 minutes)
- Alphabet Learning Game (60 minutes)
- Student presentations (10 minutes)

Student Activities:

- Present solution for baseball extra credit.
- Complete journal entry.
- Develop an Alphabet Learning game.
- Volunteers complete presentations.

Teaching/Learning Strategies:

- Presentation of solution for baseball extra credit
 - A student may present while others watch. If no student completed the extra credit, teacher presents. See baseball solution.sb from previous lesson.
- Journal entry: How do the programs on the computer know what the user wants to do next? In other words, if you are surfing the web, how does the computer know what page to go to next?
- Event lecture/description of Alphabet Learning Game
 - Allow some students to share journal entry with class. Steer them towards the idea of user events (clicks, typing) driving the program and causing it to respond.
 - Scratch provides some blocks that allow you to write programs that respond to user events relatively easily.
 - When green flag clicked (we've already seen this)
 - When Sprite clicked
 - When _ key pressed
 - Tell students that they will be making an alphabet learning game.
 - Share Alphabet Sample Rubric with the students.
 - Create the first letter in front of the class with the students helping you. See alphabet learning.sb.
 - Show how to create new costumes.
 - Explain that students may bring in pictures from the internet.
 - Download a .gif or .jpg.
 - Use import or paint to make it the second costume for your letter.

- Show how to change costumes.
 - Use a “switch to costume _” block.
- Show students how to output in talk bubbles.
 - Use a “say _ for _ sec” block.
 - Remind students that they may pick the theme of alphabet game (animals, food, etc).
- Alphabet Learning Game
 - Circulate room and answer questions.
- Voluntary student presentations
 - Facilitate students in presenting.

Resources:

- [alphabet learning.sb](#)
- [Alphabet Sample Rubric](#)

Alphabet Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
Have at least 10 different letters.	4			
Have a theme for your letter game (i.e. animals, food, etc.)	3			
Sprites change costume when clicked on.	4			
Sprites change costume when letter is typed on keyboard	4			
Use the “say _ for _ sec” to output what the letter stands for (i.e. “E is for Elephant”)	3			
Sprites all turn to letters when the “when green flag clicked”	2			
Extra Credit				
Use a microphone to record sounds for all the letters and play the sound when the letter is clicked or typed (i.e. “E is for Elephant”)	2			
TOTAL:	20			

Instructional Day: 9

Topic Description: This lesson introduces the concept of broadcasting through role play and then provides students an opportunity to complete a broadcast event in Scratch.

Objectives:

The students will be able to:

- Broadcast events.
- Listen to and respond to events they create.
- Change the background of the stage.

Outline of the Lesson:

- Journal Entry (5 minutes)
- Discussion of journal entry (2 minutes)
- Role Play (20 minutes)
- Scratch Summer Story (28 minutes)

Student Activities:

- Complete journal entry.
- Participate in discussion of journal entry.
- Participate in role play.
- Create a Scratch summer story.

Teaching/Learning Strategies:

- Journal Entry: What does it mean to broadcast something (example the radio station is broadcasting music right now)? If a radio or television station is broadcasting something, does that mean that everyone is listening to it?
- Discussion of journal entry
 - Have a few students share their responses.
 - Stress that even though a lot of things are being broadcast, not everyone is listening to every thing that is being broadcast.
- Role Play
 - Solicit Volunteers to be the various characters.
 - Give the performers a paper with ONLY their part. See Scratch Broadcast Role Play.
 - Pass out the chart that shows all the parts to students that are not performing. See Scratch Broadcast Role Play Interwoven.
 - The students can think of it as a three act play where the scenes change. The difference here is that there are no curtains so they will see everything change.
 - The teacher will be the director and will make sure everything and everyone is in place during each scene. The teacher can yell action before the scene starts to signify that everything checks out.
 - Each performer's paper is broken into scripts for the various scenes.
 - One performer will be in charge of setting the stage. They can do this by erasing and drawing pictures on the white board behind the stage.
 - The Cat's first two scripts end with broadcasts. The cat will tell the director (teacher) that it is time to go on to the next scene.

- You might want to have different students perform the role play a second time. This time the teacher will only yell out when the green flag is clicked. The students can check themselves to make sure that everything is okay.
- Interesting Questions
 - Why do The Crab and the Date have only two scripts? (Possible answer: They remain hidden during the other scene.)
 - Instead of using broadcast, do you think you could just use “wait _ secs” blocks? (Answer: yes.)
 - What might be an advantage to using broadcast instead? (Possible answer: One advantage is that if your entire program is based on waits, if you edit something in scene 1, it could possibly throw the timing off for the rest of the program.)
- Discussion
 - In Scratch, any sprite can broadcast its own event.
 - One reason why The Cat is doing the broadcasts is because he is the last one to act in the first two scenes. Therefore he knows when the scene is over.
 - Other sprites (including the one that broadcasts the event) can receive the event and perform a script
- Scratch Summer Story
 - Show students
 - Directions: Summer Story Project
 - File to edit: summer.sb
 - Rubric: Summer Story Project Sample Rubric
 - Circulate the room and answer questions.

Resources:

- Scratch Broadcast Role Play
- Scratch Broadcast Role Play Interwoven
- Summer Story Project
- summer.sb
- Summer Story Project Sample Rubric

Scratch Broadcast Role Play

This is meant to be performed in front of a white board. This can also be done using more elaborate props. Each character's parts are broken down by events that are broadcast out to everyone. Select characters and give them their parts of the scripts. There is also a script so that observers can see the flow of the entire program.

Useful props: sunglasses, a basketball, and a bag of popcorn or chips

Characters:

The Cat: our main character

The Crab:

The Opponent:

The Date:

Stage: in charge of drawing the background of the scene on the board

Scripts for the individual actors:

The Cat

when GREEN FLAG clicked:

switch to costume: sunglasses

say: Hello!

say: I'm going to tell you about my summer.

say: I spent some time at the beach.

broadcast BASKETBALL SCENE (tell everyone it's time for the next scene)

when I receive BASKETBALL SCENE:

switch to costume: basketball

say: I played lots of ball.

broadcast MOVIE SCENE (tell everyone it's time for the next scene)

when I receive MOVIE SCENE:

switch to costume: bag of popcorn or chips

say: I went on a date. We went to the movies.

The Crab

when GREEN FLAG clicked:

show: (Go up on stage. You might want to pose like a crab by making your hands into claws.)

when I receive BASKETBALL SCENE:

hide: (Disappear from the stage)

The Opponent

when GREEN FLAG clicked:

hide: (Disappear from the stage)

when I receive BASKETBALL SCENE:

show: (Go up on stage. You might want to pose like a basketball player.)

when I receive MOVIE SCENE:

hide: (Disappear from the stage)

The Date

when GREEN FLAG clicked:

hide: (Disappear from the stage)

when I receive MOVIE SCENE:

show: (Go up on stage.)

wait 2 secs:

say: I can't wait to see Get Smart. Would you believe I spent my entire check from work on these tickets? No? Would you believe that it cost me \$15 for both using my student discount? No? How about a stick of gum and a nickel I found on the floor?

Stage

when GREEN FLAG clicked:

Switch to background BEACH: (Draw a picture of the beach on the white board. A sun in one corner and a wavy line for sand is fine.)

when I receive BASKETBALL SCENE:

Switch to background BASKETBALL COURT: (Draw a picture of a basketball court. Drawing the backboard and rim should be fine.)

when I receive MOVIE SCENE:


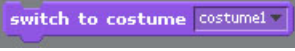


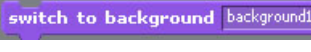
Switch to background MOVIES: (Draw a picture of a movie theater. Drawing a sign that says movies should be ok.)

Scratch Broadcast Role Play Interwoven

	when GREEN FLAG clicked:	when I receive BASKETBALL SCENE:	When I receive MOVIE SCENE:
The Cat	<p>switch to costume: sunglasses</p> <p>say: Hello!</p> <p>say: I'm going to tell you about my summer.</p> <p>say: I spent some time at the beach.</p> <p>broadcast BASKETBALL SCENE</p>	<p>switch to costume: basketball</p> <p>say: I played lots of ball.</p> <p>broadcast MOVIE SCENE</p>	<p>switch to costume: bag of popcorn or chips</p> <p>say: I went on a date. We went to the movies.</p>
The Crab	<p>show: (Go up on stage. Pose like a crab.)</p>	<p>hide: (Disappear from the stage)</p>	
The Opponent	<p>hide: (Disappear from the stage)</p>	<p>show: (Go up on stage. Pose like a basketball player.)</p>	<p>hide: (Disappear from the stage)</p>
The Date	<p>hide: (Disappear from the stage)</p>		<p>show: (Go up on stage.)</p> <p>wait 2 secs:</p> <p>say: I can't wait to see Get Smart. Would you believe I spent my entire check from work on these tickets? No? Would you believe that it cost me \$15 for both using my student discount? No? How about a stick of gum and a nickel I found on the floor?</p>
Stage	<p>Switch to background BEACH: (Draw the beach.)</p>	<p>Switch to background BASKETBALL COURT: (Draw a basketball court.)</p>	<p>Switch to background MOVIES: (Draw a movie theater.)</p>

Summer Story Project

Finish a story about what the cat did over summer. Answer questions 1, 2, and 6 on paper.

1. Open the file [summer.sb](#). Click the flag. What does it do so far?
2. Click on the cat and look at his script. What does the cat broadcast in the last block?
3. We'll make a basketball scene (a second script)
 - a. Drag a  block into the script section.
 - b. Click the empty box and choose "basketball scene".
 - c. Under looks, drag a  block into your script
 - d. Change costume1 to costume3
 - e. Give the cat something to say about playing basketball over summer.
 - f. Drag a  block to the end of this second script.
 - g. Click the empty box.
 - h. Choose new
 - i. Type in movie scene and hit ok.
4. We'll make the background change as well.
 - a. Click on the stage
 - b. Choose scripts
 - c. Drag a  block into the script section.
 - d. Click the empty box and choose "basketball scene".
 - e. Under looks, drag a  block into your script
 - f. Change background1 to basketball-court
5. Now add a third scene about going to the movies.
6. Summarize how you can use broadcast to change scenes in a story. Get your work checked off.
7. Now add in another character into each scene like in the role play (i.e. The Crab, The Opponent and The Date). These characters should show and hide.
8. Feel free to add in additional scenes.

Summer Story Project Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
Answer question 1, 2 and 6	5			
Add in the second scene (basketball)	5			
Add in the third scene (movies)	5			
Add in additional characters into each scene that show and hide	5			
Extra Credit				
Add in additional scenes	1			
TOTAL:	20			

Instructional Days: 10-13

Topic Description: Students will review how to broadcast events by developing a Scratch story and presenting it to the class.

Objectives:

The students will be able to:

- Broadcast events.
- Complete a Scratch story.
- Develop a Scratch story project.
- Assess their peers to help them gauge their progress.
- Complete a rubric.
- Prepare and make a presentation of a Scratch story to the class.

Outline of the Lesson:

- Introduction of project (5 minutes)
- Journal Entry (5 minutes)
- Review of brainstorming (10 minutes)
- Scratch story (35 minutes)
- Scratch story project (40 minutes)
- Peer review and discussion (15 minutes)
- Completion of Scratch story project (55 minutes)
- Presentation of stories (55minutes)

Student Activities:

- Complete journal entry.
- Participate in discussion of brainstorming.
- Develop a Scratch story.
- Develop Scratch story project.
- Participate in peer review and discussion.
- Complete Scratch story project.
- Present stories.

Teaching/Learning Strategies:

- Introduction of project
 - Show rubric: Story Project Sample Rubric.
 - Emphasize that they will make a small presentation along with showing their story.
 - Emphasize that there is extra credit for the best stories.
 - Show example: cat story.sb
- Journal Entry: Brainstorm some ideas for your story.
- Review of brainstorming
 - Split students into groups of three.
- Have students rotate so that each student will share brainstorms and receive feedback/suggestions from the other students.

- Develop the story
 - Have them first develop their story on paper.
- Scratch story project
 - Remind students of the problem solving that they did in unit 2 and that they can use that process to develop their overall plan and algorithm; they can also make a storyboard similar to those used in unit 3.
 - Circulate room and help students with projects.
 - If students don't know where to start
 - Suggest that they first develop their story further on paper.
 - If they have their story and don't know where to start
 - Suggest that they make a title screen or a first scene.
 - If students get stuck, break the project into smaller parts. Have them focus on the next part that they can complete.
 - Refer them to the rubric to make sure they know what they need to complete in order to earn the grade that they want.
- Peer review and discussion
 - Circulate the room and make sure students understand the rubric and what they still need to accomplish to finish their project.
- Completion of Scratch story project
 - Circulate room and help students with projects.
 - Collect projects and rubrics.
 - Help students prepare their presentations.
- Presentation of stories
 - Have students complete the Peer Grading sheet.
 - Discuss the features used in the various stories. Ask students why certain things work/don't work well.
 - To help students vote on the best, you may need to do a quick recap of the stories, i.e. Bob's story about Poodles.
 - Students should consider both the content and the items in the rubric in their voting.
 - If the class does not finish presentations in one day, the voting will be done the next day.

Resources:

- cat story.sb
- Story Project Sample Rubric
- Peer Grading

Story Project Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
The Story				
Have at least 3 scenes	10			
Have at least 4 scenes	5			
Have at least 3 different sprites	10			
Have at least 8 say or think boxes	10			
Animate the movement of your characters	5			
Use broadcast to change scenes in your story	10			
Have the characters take turns speaking to each other	5			
Have at least one conversation between characters	5			
Have a title scene with your name on it	10			
Story initializes itself when the flag is clicked	4			
The entire story plays once you click the flag	4			
The Presentation				
Explain an example from your program of how events (broadcast) were used to transition from one scene to another.	10			
Peer Grading	12			
Extra Credit				
Have the best project as voted on by peers	Up to 10			
TOTAL:	100			

Name _____ Computer # _____

VOTING

From **ALL** the projects, choose

1ST Place _____

2nd Place _____

PEER GRADING

For **EACH** of the following give the student a score from 1 to 4.

Use the rubric online to decide the score.

4 – Student has everything on the rubric: A

3 – Student has most things on the rubric: B

2 – Student has some things on the rubric: C

1 – Student turned in project, but is missing many items: D

Student Name	Score (1-4)		Student Name	Score (1-4)

Instructional Day: 14

Topic Description: This lesson provides an introduction to the concept of variables.

Objectives:

The students will be able to:

- Explain the concept of variables.
- Create examples of variables.
- Explain the concept of iteration.
- Create examples of iteration.

Outline of the Lesson:

- Finish Presentations (25 minutes)
- Journal Entry (5 minutes)
- Make Variable Example (15 minutes)
- Enhance Variable Example (10 minutes)

Student Activities:

- Finish Presentations.
- Complete journal entry.
- Participate in a discussion of the Make Variable example.
- Enhance the variable example.

Teaching/Learning Strategies:

- Finish Presentations
 - Have students fill out Peer Grading sheet.
 - To help students vote on the best, you may need to do a quick recap of the stories. You may also want to quickly replay some of the better ones.
 - Another option is to have students pick the best of each day and do a run off just replaying the top 3 from each day.
- Journal Entry: What does the word variable mean in both mathematical and English terms?
 - Time the students so they work 3 minutes individually and 2 minutes sharing with their elbow partners.
- Make Variable Example
 - Give two math examples. $x + 3 = 5$, $2x = 12$
 - Ask: What is the name of the variable here? (Answer: x)
 - Although you have x in both equations, its value **varies**: it is 2 in one equation and 6 in another.
 - The notion is the same in a program—a variable is a name that represents a value that can be changed. In the math example, the name was x .
 - Make the variable example with the students (variable example.sb) having the students help you and build their own at the same time. A possible sequence might be
 - Start by explaining that you want to make a game where you earn points for picking healthy foods and lose points for picking unhealthy ones.
 - What do you think the variable will be? If no answer, ask what name will represent a number that will change? (Answer: Points (or Good Nutrition Points in the example))

- Add the sprites for the banana, cheesie poofs and text that says “Click on food to eat it”.
- Ask: What tab do you think you should click on to make a variable? (Answer: Variables.)
- Click “make a variable” calling it Good Nutrition Points.
 - Clicking the checkbox next to the variable will show or hide it.
- Ask: If I want to make my points increase by 1 every time I choose the banana, how would I write that script? (See example.)
 - Explain that this is an example of iteration.
- As: How about making the points decrease by one when I click on the cheesie poofs? (See example.)
- Ask: What do you think should happen when the green flag is clicked? (Answer: reset the points to 0.)
 - Ask: How do you think we should do that? (In example the script for this is under Sprite4)
 - Ask: Does it matter which script the “when green flag clicked” is under? (Answer: No.)
 - Explain that this is an example of initialization.
- Enhance Variable Example
 - Have Students enhance the variable example by
 - Adding a food that is worth 2 points when clicked on.
 - Adding a food that is -3 points when clicked on.

Resources:

- Peer Grading

Instructional Day: 15

Topic Description: This lesson provides an introduction to the concept of conditionals.

Objectives:

The students will be able to:

- Explain the concept of conditionals.
- Enhance a variable program with conditionals.

Outline of the Lesson:

- Journal Entry (5 minutes)
- Conditional lecture (15 minutes)
- Age program (10 minutes)
- Age solutions (5 minutes)
- Enhance variable example (20 minutes)

Student Activities:

- Complete journal entry.
- Participate in discussion of conditions.
- Develop an Age program.
- Review Age solutions.
- Enhance the variable example.

Teaching/Learning Strategies:

- Journal Entry: What comes to mind when you hear the word “if”? What are some ways we use the word “if” in English?
 - Time the students so they work 3 minutes individually and 2 minutes sharing with their elbow partners.
- Conditional lecture
 - Have a few students share their responses for the “if” parts and use that as a springboard.
 - In English, if is used to state a condition where something might happen if the condition is true. Hence this topic is called conditionals. Point out that this is a common computer science construct.
 - An example from computing is when a program like Microsoft Word asks you if you want to save your work when you hit close. If you click yes, it saves your changes. If you click no, it discards your changes.
 - if (some condition)
 - then do this
 - Show students “if” block in Scratch.
 - Notice that only hexagon shaped blocks can fit within it.
 - Notice that if the condition is true, it will do anything that is enclosed within the top and bottom of the “if” block.
 - Show the students age.doc and age.sb.
 - Remind students that since we are using integers (whole numbers) > 15 it means people that are over 15 not including 15.
 - Show them how to use the slider to change the age.
- Age Program

- For solution, see age solution.sb.
- Age solutions
 - Show a solution like age solution.sb.
 - Show an alternate solution.
 - Since numbers are integers (whole numbers) we can do “age > 2” to mean “age >= 3”.
 - To do >= in scratch, you need to use the “or” block. See age – greater-equal.sb.
- Enhance variable example
 - Instruct students to go back into their variable example about nutrition and add:
 - A message about being nutritious if the number of points becomes greater than 9.
 - A message about eating healthier food if the number of points becomes less than –4.
 - They can either have a sprite say the message or use broadcast to change the sprites/stage to convey the message.

Resources:

- Age Project
- age.sb
- age solution.sb
- age – greater-equal.sb
- variable example.sb

Age Project

You are going to finish a program that will tell you what you can do depending on your age. Use the slider to set the age.

1. Currently, it only does the first condition. Your task is to finish the program so that the cat will tell you the rest:

If you are older than 2 "you don't need diapers"

If you are older than 15 "you can drive"

If you are older than 16 "you can see an R rated movie"

If you are older than 17 "you can vote"

If you are older than 20 "you can gamble"

If you are older than 24 "you can rent a car"

If you are older than 49 "you can retire"

2. If the age is less than 3, make the code tell you:
"Sorry, you are not old enough for anything yet"
3. Feel free to add more conditions.

Instructional Days: 16-17

Topic Description: This lesson introduces And, Or and randomness. Students have an opportunity to practice utilizing these features in the context of programs.

Objectives:

The students will be able to:

- Use conditionals with And and Or to write a grade program.
- Use a random number generator to write a dice program.

Outline of the Lesson:

- Journal Entry (5 minutes)
- And/Or discussion (15 minutes)
- Grades program (35 minutes)
- Random lecture (20 minutes)
- Dice (35 minutes)

Student Activities:

- Complete journal entry.
- Participate in And/Or discussion.
- Develop Grades program.
- Participate in discussion of Random.
- Complete Dice program.

Teaching/Learning Strategies:

- Journal Entry: What's the difference between And and Or? What does the word random mean in English?
 - Students should complete individually and then share with their elbow partners.
- And/Or Discussion
 - Start with a few journal entries about And and Or.
 - Kinesthetic And/Or Activity (Following is a possible set of conditions.)
 - Tell the students to stand up if the condition is true.
 - Say: If (you are a girl AND you are wearing blue) stand up.
 - Find a girl that is not wearing blue and is sitting. Ask her why she is sitting if she's a girl? (Answer: she's not wearing blue)
 - Ask: How many parts of the condition must be true for you to stand up if it is an AND? (Answer: both)
 - Say: If (you are a boy OR you are wearing blue) stand up.
 - Find a boy that is standing but is not wearing blue. Ask: Why are you standing if you are NOT wearing blue? (Answer: I'm a boy)
 - Ask: How many parts of the condition must be true for you to stand up if it is an OR? (Answer: at least one)
 - Ask: If both parts of the condition are true for an OR, is it ok to stand? (Answer: YES!)
 - Show the students the "and" and "or" blocks in Scratch.

- Emphasize how they are hexagon shaped and take two other hexagons.
 - Show the students Grades Project.
- Grades
 - Circulate and help students with projects.
 - If many students are stuck, build the “B” part of the code together as a class.
 - In the last minute, have students share their solutions with their elbow partners.
- Random lecture
 - Have a few students share their journal entries about what random means.
 - Ask: if I roll a pair of dice, will the numbers come out in order (2, then 3, then 4 the next roll, etc.) (Answer: Most likely not)
 - Roll a pair of dice a few times to prove it.
 - This unpredictability is called randomness.
 - Randomness can make games more exciting.
 - For example, how many spaces will I get to move this turn?
 - Randomness is also used in computer science for simulations and in scientific/statistical experiments.
 - Walk students through dice.sb showing them the “pick random _ to _” block.
 - Explain that the numbers create the range that the random integer can fall under. The block works inclusively. Therefore 1 to 6 will produce numbers 1,2,3,4,5,6.
- Dice
 - Instruct students to finish dice.sb so that it creates a pair of dice. They can create their own look for the dice.
 - Circulate and help students with projects.
 - In the last minute, have students share their solutions with their elbow partners.

Resources:

- Grades Project
- grades solution.sb
- dice.sb
- dice solution.sb

Grades Project

Your task is to make a Scratch program that will tell you the letter grade based on the percentage.

1. Create a variable grade.
2. Double click grade to display the scroll bar.
3. When the green flag is clicked, the program should look at the value of grade and the sprite should respond with a letter as follows:
 - A: greater than 90
 - B: greater than 79 and less than 90
 - C: greater than 69 and less than 80
 - D: greater than 59 and less than 70
 - F: less than 60

At Crazy High School, students only qualify for tutoring if they have a B OR a D. After it says the grade, make your program say “You qualify for tutoring” if the grade is a B or D.

Instructional Day: 18

Topic Description: This lesson requires students to apply their knowledge of conditionals to develop a Rock Paper Scissors program in Scratch.

Objectives:

The students will be able to:

- Apply knowledge of conditionals to complete a Rock Paper Scissors program.

Outline of the Lesson:

- Review of Rock Paper Scissors rules (5 minutes)
- Rock Paper Scissors discussion (10 minutes)
- Rock Paper Scissors project (40 minutes)

Student Activities:

- Review Rock Paper Scissors rules.
- Participate in Rock Paper Scissors discussion.
- Complete Rock Paper Scissors project

Teaching/Learning Strategies:

- Review of Rock Paper Scissors rules
 - Lead a class discussion—students volunteer to share the rules for Rock, Paper Scissors.
- Rock Paper Scissors discussion
 - Give students a tour of rps starter.sb.
 - Show them how there are variables for ROCK, PAPER and SCISSORS.
 - Ask: Why might it be easier to work with the variables instead of just using numbers? (Answer: It makes the code easier to read.)
 - Show students variables for player and computer.
 - Ask how does the computer determine if they will choose rock, paper, or scissors? (Answer: It randomly chooses one using “pick random 0 to 2”.)
 - Closely examine the computer’s “when I receive showPick” script
 - Explain how the else part works if the condition of the if is false.
 - Ask: Why don’t you need a statement that says “if computer = scissors”? (Answer: You asked if it was = to rock and that was false, then you asked if it was equal to paper and that was false so the only thing left was for it to equal scissors. Hence, you can just put the “switch to costume scissors” in the else.)
 - Instruct students that they only need to change the script that starts with “When I receive determine winner” under the computer sprite. (They may change more features if they have time.)
 - Facilitate them in writing some pseudo code to handle all the cases for the computer choosing ROCK. Remind students that this is an algorithm.
 - Create two versions, one like rps solution.sb and one like rps solution b.sb.; this way students can choose the method that they understand better.
 - Show students a working example in presentation mode (so they can’t see the blocks).

- Rock Paper Scissors project
 - Circulate room and help students with projects.
 - Allow students to try various approaches to solving the problem.
 - If students finish, offer them extra credit for keeping score of the wins for the computer and player.

Resources:

- rps starter.sb
- rps solution.sb
- rps solution b.sb

Instructional Day: 19

Topic Description: This lesson builds on previous concepts to create a timer.

Objectives:

The students will be able to:

- Create a timer.

Outline of the Lesson:

- Review of Rock Paper Scissors solutions (10 minutes)
- Creation of a timer (15 minutes)
- Review of Timer solutions (5 minutes)
- Introduction of Timing Game (15 minutes)
- Timing Game theme (10 minutes)

Student Activities:

- Review Rock Paper Scissors solutions.
- Create a timer.
- Review Timer solutions.
- Choose Timing Game theme.

Teaching/Learning Strategies:

- Review of Rock Paper Scissors solutions
 - Review rps solution.sb and rps solution b.sb.
 - Allow students to share their own unique solutions.
- Creation of a timer
 - Explain to students that they will make a timer that will count down from 10 to 0.
 - Show students Timer Project.
- Review of Timer solutions
 - Allow students to share their own unique solutions.
 - Review timer solution a.sb and timer solution b.sb.
- Introduction of Timing Game
 - Have students help build an example. (See timing.sb.)
 - Review Timing Game Sample Rubric.
- Timing Game theme
 - Circulate room and help students pick the theme of their timing game.

Resources:

- rps solution.sb (modified version of Jesse Moya’s solution)
- rps solution b.sb (modified version of Jesse Moya’s solution)
- Timer Project
- Timing Game Sample Rubric
- timer solution a.sb

- timer solution b.sb
- timing.sb

Timer Project

How to make a timer in Scratch:

1. Create a variable called timer.
2. When the flag is clicked, initialize the timer to 10.
3. Continually, wait 1 second and check if the timer = 0
 - a. output the current time either with a sprite or just show the variable
 - b. If the timer = 0 make either the background or a huge sprite say “Time’s Up”
4. When the flag is clicked, everything should start over.
5. Be creative as to what you want your program to look like.
6. Make sure the timer stops at 0 and does not continue into negatives.

Timing Game Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
The Game				
Have 3 or more “timed” sprites	10			
Have 4 or more “timed” sprites	5			
Use a timer for your game	5			
Keep score (points)	10			
Give the user feedback as to how well they timed their button pressing	10			
Have a help screen with directions	5			
Does the game reset when the flag is clicked	10			
Does the game stop when it is over	5			
Does the game notify the user when it is over	10			
Does the game keep track of how many “perfects” in a row	5			
Does the game get harder as you keep playing	5			
Peer Grading	20			
Extra Credit				
Have the best project as voted on by peers	Up to 10			
TOTAL:	100			

Instructional Day: 20-23

Topic Description: Students create a timing game in Scratch and participate in an Arcade Day during which they display their games.

Objectives:

The students will be able to:

- Create a timing game.
- Assess their peers to help them gauge progress.
- Complete their rubrics and submit their timing games.
- Prepare a presentation of a Scratch program.
- Evaluate their peers' timing games.

Outline of the Lesson:

- Timing game (95 minutes)
- Peer Review and discussion (15 minutes)
- Completion of timing game (70 minutes)
- Arcade walk (40 minutes)

Student Activities:

- Work on timing game.
- Participate in peer review and discussion.
- Continue working on and complete timing game.
- Participate in arcade walk.

Teaching/Learning Strategies:

- Work on timing game
 - Circulate room and help students with projects.
- Peer review and discussion
 - Circulate the room and make sure students understand the rubric and what they still need to accomplish to finish their project.
- Completion of timing game
 - Circulate room and help students with projects.
 - Collect projects and rubrics.
 - Help students prepare their presentations.
- Arcade Walk
 - Have students rotate through the room playing each other's games and giving each one a score on their Peer Grading sheet. Use a timer to indicate the amount of time that each student has at each computer.
 - Have students vote for the top two games out of the entire class. The vote should be based on both the content and adherence to the rubric.
 - Discuss features of games and how they conform to the rubric. What types of programming strategies did students use?

Resources:

- Timing Game Sample Rubric
- Peer Grading

Instructional Day: 24**Topic Description:** Investigating Games**Objectives:**

The students will be able to:

- Investigate two different types of games.
- Get ideas for their final projects.

Outline of the Lesson:

- Monkey game (25 minutes)
- Review of answers (5 minutes)
- Pinball game (25 minutes)

Student Activities:

- Complete Monkey game.
- Review answers.
- Complete Pinball game.

Teaching/Learning Strategies:

- Monkey game
 - Have students answer the questions in Monkey Game Project.
 - Have students enhance monkey game.sb.
- Review answers
 - See Monkey Game Project Solutions and monkey game solution.sb.
- Pinball game
 - Have students answer questions in Pinball Project.
 - Have students enhance pinball.sb.

Resources:

- Monkey Game Project
- Monkey Game Project Solutions
- monkey game solution.sb
- monkey game.sb
- Pinball Project
- Pinball Project Solutions
- Pinball.sb (An example that comes with Scratch)

Monkey Game Project

Answer these questions on paper:

1. Play the game by using the arrow keys. What blocks make the monkey respond to the keys?
2. Does the banana always appear in the same place?
3. What blocks do you think decide what x and y the banana should change to?
4. What are the names of the orange blocks under Variables?
5. What block(s) are used to change the score?

Make these changes to the file:

6. Customize the sprites in the game (make the characters be who you want).
7. Add another sprite that gives you 2 points if you touch it.
8. Get the game to stop at 10 points or more by telling you that you win.

Monkey Game Project Solutions

Answer these questions on paper:

9. Play the game by using the arrow keys. What blocks make the monkey respond to the keys? "when _ key pressed"
10. Does the banana always appear in the same place? No, it's random.
11. What blocks do you think decide what x and y the banana should change to? "set x to _" and "set y to _" combined with "pick random _ to _"
12. What are the names of the orange blocks under Variables? "change points by _", "set points to _", and "points"
13. What block(s) are used to change the score? "set points to 0" when the green flag is clicked and "change points by 1" when the monkey touches the banana.

Make these changes to the file: [see monkey game solution.sb](#)

14. Customize the sprites in the game (make the characters be who you want).
15. Add another sprite that gives you 2 points if you touch it.
16. Get the game to stop at 10 points or more by telling you that you win.

Pinball Project

Open pinball.sb and answer the questions below on paper:

1. Look at the scripts for the pinball. How did the author simulate gravity?
2. How does the ball know when to “bounce” off of something?
3. Does the ball always bounce the same way when it hits something?
4. How do you think the ball determines which direction to bounce?
5. What’s the purpose of the purple line at the very bottom of the game?
6. Modify the game to keep track of points and get it checked off. Write down what changes you made.
7. What other features do think would make this game better?

Pinball Project Solutions

Open pinball.sb and answer the questions below on paper:

1. Look at the scripts for the pinball. How did the author simulate gravity? **There is a variable called gravity that is constantly affecting the direction of the ball.**
2. How does the ball know when to “bounce” off of something? **If it touches something that is green, orange or red, it will “bounce” off of it.**
3. Does the ball always bounce the same way when it hits something? **No, the amount of the turn is random.**
4. How do you think the ball determines which direction to bounce? **It uses “pick random _ to _” to vary the amount of the turn.**
5. What’s the purpose of the purple line at the very bottom of the game? **If the ball touches purple, you lose.**
6. Modify the game to keep track of points and get it checked off. Write down what changes you made. **See pinball solution.sb**
7. What other features do think would make this game better? **Answers will vary.**

Instructional Day: 25**Topic Description:** Introduce the final project.**Objectives:**

The students will be able to:

- Make an appropriate choice of which final project they will do.

Outline of the Lesson:

- Review of answers for Pinball Project (5 minutes)
- Introduction of projects (15 minutes)
- Final projects (35 minutes)

Student Activities:

- Review answers for Pinball project.
- Participate in discussion of introduction to projects.
- Start Final Projects.

Teaching/Learning Strategies:

- Review of answers for Pinball project.
- Introduction of projects
 - Review both sets of descriptions and rubrics.
- Final projects
 - Circulate the room
 - Help students decide which project to do.
 - If students are making the game, tell them to start off small and add features as they go along to ensure that they will finish.

Resources:

- Pinball Project
- Pinball Project Solutions
- Game Project Sample Rubric
- My Community Project
- My Community Sample Rubric

Instructional Day: 26-28

Topic Description: Complete final projects.

Objectives:

The students will be able to:

- Incorporate all objectives in the unit into the final project.

Outline of the Lesson:

- Work on final project (110 minutes)
- Peer review and discussion (15 minutes)
- Completion of final project (40 minutes)

Student Activities:

- Work on final project.
- Participate in peer review and discussion.
- Complete final project.

Teaching/Learning Strategies:

- Work on final project
 - Circulate room and help students with projects.
- Peer Review and discussion
 - Pair students with someone that has a similar project if possible (communities together and games together).
 - In pairs, students take turns using a rubric to grade the other person's project. Students should be able to get an idea of where their grade stands now, and what else they need to do to complete their project.
 - Circulate the room and make sure students understand the rubric and what they still need to accomplish to finish their project.
- Completion of final project
 - Circulate room and help students with projects.

Resources:

- Game Project
- My Community Project
- Game Project Sample Rubric
- My Community Sample Rubric

Instructional Day: 29**Topic Description:** Complete final projects.**Objectives:**

The students will be able to:

- Finish their KWL charts.
- Complete their rubrics and turn in their final projects.
- Prepare their presentations.

Outline of the Lesson:

- KWL chart (15 minutes)
- Completion of final projects (40 minutes)

Student Activities:

- Complete KWL chart.
- Groups take turns sharing their L's orally.
- Finish final projects.

Teaching/Learning Strategies:

- KWL chart
 - Refresh memory on activity.
 - They already did the K (Know) and W (Want to learn), now they must fill in the L (Learned) section of their chart.
 - They should meet in their original groups.
 - Students fill in the L portion of their chart.
- Groups take turns sharing out their L's orally. Encourage them not to repeat anything that has already been said.
- Completion of final projects
 - Collect projects and rubrics.
 - Help students prepare their presentations.
 - Note: If time permits, the gallery walk of projects can be completed here to allow more time for presentations on the last day.

Resources:

- KWL Graphic Organizer Chart.pdf (UCLA SMP)

Instructional Day: 30

Topic Description: Complete presentations.

Objectives:

The students will be able to:

- Complete a presentation on the final project.

Outline of the Lesson:

- Gallery walk (10 minutes)
- Final Project Presentations (45 minutes)

Student Activities:

- Participate in gallery walk.
- Complete final project presentations.

Teaching/Learning Strategies:

- Have students do a gallery walk of the projects.
 - Complete the Peer Grading form for each project.
- Presentations
 - Have students take turns presenting. They will present their product as well as explain how they created the projects.
 - Help guide students by asking questions if the student does not fully explain how they wrote their program.
 - Have students in audience complete their Peer Grading forms.
 - Have students vote for first and second place.

Resources:

- Peer Grading
- My Community Project Sample Rubric
- Game Project Sample Rubric

Final Project

Choose one of the following:

My Community Project

You will use Scratch to make a project about your community. You will use blocks like broadcast in your project. You should have at least three different pages or scenes in this project.

Decide on one positive thing that you want to highlight and one thing you want to improve about your community. Then find at least one statistic to backup your conclusions. Also include at least one personal comment/recording and one picture. Lastly, you should have at least one observation from someone else in the class about the topic – this means you will have to ask them what they think and either record it or write it down.

Use these websites to find statistics about Los Angeles and California:

For Education Data go to:

<http://dq.cde.ca.gov/dataquest/>

<http://www.ed-data.k12.ca.us/welcome.asp>

For information about other concerns in LA, like hunger and homelessness, go to:

<http://www.unitedwayla.org/GETINFORMED/RR/Pages/default.aspx>

<http://www.unitedwayla.org/getinformed/rr/datalinks/Pages/default.aspx>

For information about health and health care go to (if you need a password, let me know):

<http://www.chis.ucla.edu/>

For Los Angeles Sponsored web sites go to:

<http://www.ci.la.ca.us/>

Create A Game Project

Create your own game that meets the specifications outlined in the rubric. Be creative!

Community Project Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
The Content				
3 or more scenes	10			
1 or more positive things in the community that you want to highlight	5			
1 or more things in the community that you would like to improve	5			
Statistics to back up your conclusions	5			
A personal comment/recording	5			
A personal picture	5			
A comment/recording from someone else in the class	5			
The Program				
Use broadcast	10			
Scenes only show sprites that are supposed to be in that scene	10			
Program starts and restarts when green flag is clicked	10			
The Presentation				
Show and explain the contents of each scene in your project	9			
Explain how each scene changes (how the program works)	9			
Peer Grading	12			
Extra Credit				
Have the best project as voted on by peers	Up to 10			
TOTAL:	100			

Game Project Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
The Game				
Let the player know if they win	10			
Keep score	10			
Have a timer	10			
Have a help screen with directions	10			
The game resets when the green flag is clicked	10			
The game stops when it is over	10			
Does the game get harder as you keep playing (more than one level)	10			
The Presentation				
Brief description of the game explaining how it was programmed using Scratch (what blocks you used, etc.)	10			
Peer Grading	20			
Extra Credit				
Have the best project as voted on by peers	Up to 10			
TOTAL:	100			

Unit 5:

Computing and Data Analysis



Introduction

Managing and interpreting large amounts of data is part of the foundation of our information society and the economy. The ability to analyze, visualize and draw conclusions from large data sets is critical to computing. This unit has been designed to allow students the opportunity to experience the process of data collection and analysis in real-world contexts. Although it is highlighted throughout the unit, it is important to make special note of the fact that R and the graphical data analysis system, Deducer, are just tools for the data analysis that the students will be doing throughout the unit. Students need to be able to use the commands in appropriate ways to manipulate the data, but the focus should be on conceptual understanding. This is especially important since R is only one of many tools that can be used for data analysis.

The unit is divided into three main sections.

- Overview of the final project and the data collection process (Days 1-6)
- Data analysis techniques (Days 7-23)
- Final project (Days 24-30)

The goal is to prepare students to collect rich data, formulate queries that will inform whatever campaign they choose, and to use that information to either make a case or facilitate a discovery.

Due to the possibility of technical difficulties at any point in the unit, extra time has been built in to “catch up” should these occur. By the time students reach the final project, much of the data analysis will be complete; therefore, fewer days can be allocated the project, if that seems reasonable. This extra time also provides flexibility to spend additional time on lessons where discussions are rich rather than cutting those short. For example, additional time could be allotted for students to get to know the members in their group and plan their campaign.

Teachers who do not have mobile technology available can still teach this unit. Some possible options for the days on which students use their own data include: Pose more questions related to the data sets provided in the curriculum or use the contextual data sets included in the Canonical Campaign Supplement to create additional activities for each analysis type. A possible option for the final project is to extend the project from Unit 2 to include more detailed analysis and then proceed with the design of a web page or Scratch program.

Specific topics for each instructional day are listed in the overview chart on the next page.

Note: Information on Deducer can be found in the Deducer Quick Start Guide. Information specific to the canonical phone applications can be found in the Canonical Campaigns Supplement. Both of these documents are included with the Supplemental Materials.

Daily Overview Chart	
Instructional Day	Topic
1	Review how data can be used for making a case/discovery and provide an overview of the final project.
2	Discuss photo ethics and student safety related to android phone use.
3-5	Distribute phones. Create groups. Discuss group roles and responsibilities. Navigate the android application. Navigate the online system.
6	Data check-in—Discuss issues that arise (aggregating data, etc.).
7-10	Introduce R/Deducer. Create maps using the latitude and longitude of a location and then create maps from a file of data.
11	Create maps with student data and related data set.
12-14	Discuss bar plots, categorical and continuous data, and mosaic plots as a vehicle for comparing categorical data, and looking at trends in data.
15	Create bar plots and mosaic plots with student data and related data set.
16-18	Review mean, median, minimum, maximum. Discuss various ways to subset data. Represent data with box plots and histograms.
19	Identify mean, median, minimum, maximum, create subsets, and create box plots and histograms with student data and related data set.
20-22	Use a variety of filters and queries to create subsets of text data. Create bar plots to graphically display the information.
23	Analyze text in student data and related data set.
24-26	Finalize data analysis for final project.
27-29	Develop website or Scratch program to present data analysis campaign.
30	Final project presentations

Daily Lesson Plans

Instructional Day: 1

Topic Description: This lesson sets the stage for the unit. It provides a review of the data collection and analysis that will be needed in order to complete the final project.

Objectives:

The student will be able to:

- Explain the possible themes for the final project.
- Explain the difference between data used for making a case and data that informs a discovery.

Outline of the Lesson:

- Collect permission slips (5 minutes)
- Review of data collection and making a case/discovery (30 minutes)
- Overview of final project (20 minutes)

Student Activities:

- Participate in discussion of data review.
- Participate in discussion of final project.

Teaching/Learning Strategies:

- Collect parent permission slips.
- Overview of the Final Project
 - If you have access to videos of sample projects, show a few of them.
 - Use the appropriate parts of the Canonical Campaigns Supplement as a guide for this discussion. The overview will help students understand why they are receiving phones, what they will be doing with the phones, what issues they should think about in data collection, and what questions they might choose to ask as part of their analysis. You may choose to provide students with a printed copy of this overview, but the most important thing is to encourage the discussion. Highlight each of the following in detail.
 - Context and examples
 - Data collection with the android application
 - Analysis phases
 - Making a case v. discovery

Resources:

- Canonical Campaigns Supplement

Instructional Day: 2

Topic Description: In this lesson, photo ethics and student safety related to android phone use are discussed.

Objectives:

The student will be able to:

- Explain why they are not allowed to take photos of people during their data collection.
- Describe situations in which they should not take any type of photos.

Outline of the Lesson:

- Journal Entry (10 minutes)
- Photo ethics (20 minutes)
- Safety of the photographer (10 minutes)
- How to represent people in photos (15 minutes)

Student Activities:

- Complete journal entry.
- Participate in photo ethics discussion.
- Participate in photographer safety discussion.
- Participate in how to represent people in photos discussion.

Teaching/Learning Strategies:

- Journal Entry: Have you ever felt uncomfortable when taking a photo of someone? Have you ever felt uncomfortable when someone took a photo of you? Why?
- Discuss photo ethics. (See Photoethics.ppt in the Supplemental Materials for the photographs to display.)
 - Project Dorothea Lange photograph of “Migrant Mother”.
 - Ask students if they recognize the photograph.
 - Ask questions such as: What does the image depict?
 - Probe as necessary—ask them other questions around the image itself. For example: What is this woman doing in the photograph? Who do you think she is? Where do you think she might be from? What do you think she does for a living and how can you tell? Who do you think the children are? Why do you think they are turned away from the camera?
 - Ask a question such as: How does this photo make you feel?
 - Most likely, students will say things such as “sad”, “depressed”, “hungry”, or maybe “nothing.” If students say that they don’t get any reaction from the image, ask them what they think the *woman* is feeling in the image.

- Ask questions such as: What in the picture makes you feel that way? (You might want to get them thinking about things like the lack of color, the lighting, the children’s faces turned away, the skinny woman with a worried look, and the wrinkles in her face, etc.)
- Finally, explain the history of the photo: From 1935-42, the US Farm Security Administration created to address agricultural problems and rural poverty during the Great Depression sent out photographers to help document what life was like in rural and poor areas of the country. This photograph is by Dorothea Lange and became world-famous for illustrating rural poverty in America.
- Next, show the picture from the BBC News website
 - Ask a question such as: What is happening in this photograph?
 - Probe as necessary—ask them other questions such as: Where do you think this photo was taken? Who do you think this woman is? What is she doing in the picture? Who else is in the photograph? What is the soldier doing? Why do you think he is laughing? How do you think the woman is feeling at this moment? Why would she be picking up rice?
 - It is possible that students won’t recognize anything wrong at first since they are probably used to seeing these types of images on television or in magazines/newspapers.
 - Then ask questions such as: How does this photograph make you feel? Why?
 - Next, show both photos (flipping back and forth between slides 1 and 2) and ask the students: Does anything seem wrong about these photos?
 - Try to get students thinking about the ethics of the photographer’s role. You could ask questions such as: Who took these pictures? Where were the photographers when these pictures were taken? What do you think happened to these people after the photographer took the picture? What did the photographer win out of these photos compared to the subjects of these photos? How do you think the people in these photos felt when the photographer took their pictures? Happy to be photographed? Embarrassed? Unsure of themselves? Uncomfortable?
 - Finally, explain the story of the BBC photograph: This picture was published on the BBC News Website. It is a photo taken in the Middle East of a woman who is picking up spilt rice grains to feed her family. The soldier in the background may be laughing at her or may be smiling at the camera—it isn’t really clear. Regardless, you can tell that the soldier doesn’t really care about the woman’s situation and that the woman is so starved and desperate that she is taking the time to pick up tiny grains of rice on the ground. Students may provide insights such as: There are many homeless people and people picking through the trash in the United States and they may have gotten so used to seeing these people that they don’t even realize they are there; this might be the situation for the soldier. As human beings we have the power to help people in need around us and we shouldn’t ignore others or take advantage of them by gaining a photograph out of their misery.

- Explain to students that, as photographers, they should be conscious of the lives, privacy, and experiences of people around them; that these photographs may feel disturbing or uncomfortable—and if they feel uncomfortable looking at the images, this is a good thing because they are beginning to recognize how a photograph can really take advantage of people and be unethical. Discuss their journal entries and ask them to think of times when they might have felt uncomfortable when someone they didn't trust took their photo or when somebody they didn't know very well posted their photo on MySpace or Facebook.
- Remind students that photographers have an important responsibility to respect the privacy of the things/people they take photos of (even if people aren't in the photograph) because people surrounding a person with a camera might feel uncomfortable too without the photographer even realizing it.
- When students receive the cell phones and start taking pictures of things, they should be aware of how people near the object of the photo might feel about the camera/phone being pointed at them.
- Safety of the Photographer
 - Next, show the war photograph.
 - Ask questions such as: What is happening in this photograph? Where do you think this is taking place? Where is the photographer? What feelings does the image evoke?
 - Discuss how the photographer is at risk in this picture—can be easily shot down just like the soldiers.
 - Explain that as they collect data for this unit they should **NEVER** put themselves at risk. If taking the photograph is going to hurt them in any way—because somebody doesn't like that they're taking a photo, because someone sees their phone and wants to steal it from them, because they're standing in the middle of a busy street and might get hit by a car, etc.—then they should NOT take the photo at that moment, but should wait and take a picture of something that REPRESENTS what they wanted to take a picture of.
- Photo ethics and safety related to cell phone use
 - Explain to students: When they receive the cell phones, they will not be allowed to take any pictures of people.
 - This will prevent people from feeling uncomfortable if they take pictures around them
 - This will prevent the students being put at risk if they take pictures in dangerous areas of their neighborhoods, etc.
- Summary:
 - Don't make people feel uncomfortable by taking pictures of things with the cell phones.
 - Don't put yourself in an awkward or dangerous position by taking a photo.
- How do you represent people in the photos?
 - Brainstorm photography alternatives with the students by doing the following:
 - Ask students to imagine their best friend. Ask students to make a list of things that remind them of that friend in their journal. These could be objects related to memories they share with that friend, experiences they have had with that friend, inside jokes, etc.

- Ask students to share what they have written.
- Explain to students that this is how they can take pictures of people—by taking photographs of things that represent the person instead of taking pictures of the person him/herself.
- Whenever they want to take a picture of a specific person, they should wait and take pictures of an object or scene or something else that represents that person.
- Ask students to write in their journals about what they would take photographs of to represent an important person in their life, for example their mother or grandmother.

Resources:

- Photoethics.ppt (Supplemental Materials)

Instructional Days: 3-5

Topic Description: Phones are distributed. Groups for the final project are created and roles and responsibilities for group members are assigned. Students navigate the chosen android application, navigate the online application, and develop a list of possible questions to use for the final project.

Objectives:

The student will be able to:

- Explain the rules for sharing the phones and why those rules are in place.
- Login and navigate through the basic features of the phone application.
- Login and navigate through the basic features of the online system.
- Develop a method for data collection.

Outline of the Lesson:

- Phone distribution (25 minutes)
- Online system (25 minutes)
- Phone application (25 minutes)
- Group roles and responsibilities (35 minutes)
- Data collection method for the final project (55 minutes)

Student Activities:

- Groups complete phone paperwork.
- Experiment with the online application.
- Experiment with the phone application.
- Groups discuss group roles and responsibilities.
- Groups discuss potential questions regarding advocacy/discovery and data collection.

Teaching/Learning Strategies:

- Distribute phones
 - Have groups complete all phone paperwork.
 - Sharing the phone—Remind students that they need to share the phones with their group. Tell them to coordinate with their group how they will be sharing the phone. Each phase will be long enough that everyone should have a few days of data collection on their own, even though they are sharing the phone.
 - Remind students to follow school-wide rules when using phones.
 - Explain limitations on phone functionality (data collection device only).
- Determine who in each group will be responsible for the phone on the first night.
- Navigate the chosen application both online and on the phone. (You can use the Canonical Campaigns Supplement as a resource.)

- Demo the online system.
- Explain how the phone application works.
- Give students time to explore both applications to become familiar with the features. Make sure that each student has an opportunity to use the phone application and understands how to enter data.
- Group roles and responsibilities
 - Explain to students that everyone in the group is accountable for the work on the in the various stages and that they need to alternate roles depending on who is the data collector for a given night, etc. They also need to come to consensus on decisions related to data collection, questions for analysis, etc.
 - Provide students with dates for the data check-ins/analysis with their own data. (These may need to be revised if sections take longer than anticipated.) Students should plan to collect some data prior to Day 6 data check-in to use as a base for discussion.
 - Review the examples that were provided on day 1 of the unit.
- Discuss campaigns
 - Have student groups consider the research questions in the Canonical Campaigns Supplement as they develop a method for data collection.
 - Answer questions and guide them as appropriate.

Resources:

- Phone distribution forms
- Canonical Campaigns Supplement

Instructional Day: 6

Topic Description: Data collection check-in.

Objectives:

The students will be able to:

- Identify issues related to the data collection process.
- Explain aggregation of data.

Outline of the Lesson:

- Data check-in (10 minutes)
- Journal Entry (5 minutes)
- Data collection issues and aggregation of data (40 minutes)

Student Activities:

- Groups discuss the data collected to date.
- Complete journal entry.
- Participate in discussion about data collection issues and aggregation of data.

Teaching/Learning Strategies:

- Data check-in
 - Have the person in charge of the first phase of data collection upload their data and share with the rest of their group.
- Journal Entry: Consider the data that your group collected. What issues did you have with collecting the data?
- Discuss issues that have arisen with data collection.
 - Have each group describe the data they have collected to date.
 - Clarify any misconceptions about what they should be collecting.
 - Did they understand the various prompts and possible responses?
 - How many entries did they collect? How does that compare with other groups?
 - Discuss aggregation of their data.
 - Why is it important for each member of the group to be a data collector for some period of time?
 - Why will we want to pool the data from all of the groups at the end of the unit even if each group is working on a different set of research questions?

Resources:

- No additional resources needed

Instructional Days: 7-10

Topic Description: This lesson introduces R/Deducer as a data analysis tool. The basic features of loading and saving files, sorting and creating subsets are explored. Maps are created by using the latitude and longitude of a location and then maps of points and bubble charts are created from a file of data.

Objectives:

The students will be able to:

- Translate addresses into latitude/longitude.
- Sort files of data.
- Create subsets of data.
- Read location data from a file and plot points on maps.
- Create bubble plots.

Outline of the Lesson:

- Journal Entry (5 minutes)
- Describing location (30 minutes)
- Exploring LA Bike Data and Deducer (30 minutes)
- LA Bike Activity (45 minutes)
- LA Bus Stops Activity (45 minutes)
- Bubble charts (20 minutes)
- Bubble Charts Activity (45 minutes)

Student Activities:

- Complete journal entry.
- Participate in discussion of location, LA Bike Data and Deducer.
- Complete LA Bike Activity.
- Complete LA Bus Stops Activity.
- Participate in discussion of bubble charts.
- Complete Bubble Chart Activity.

Teaching/Learning Strategies:

- Journal Entry: Consider the data that you have been collecting with the phone. How might seeing the data on a map help you analyze it?
- Install Deducer package. (See Deducer Quick Start Guide.)
 - You may want to do this installation yourself before the class to save time.
- Describing location
 - Use Walking and Biking in LA as an introduction to the LA bike data and describing location with latitude and longitude. (You may want to share a version of this resource with students.)

- Take the opportunity to point out that this is a campaign similar to what they will be doing for the final project—people like them concerned about a topic, collecting data to inform their cause.
- Load the Google Map with the locations of the 56 intersections for the survey.
- Ask questions about these locations: Are any of these locations in your neighborhood? Near the school? Near your home? Other questions that may be of interest.
- Go to <http://www.getlatlon.com/> and demonstrate how to translate a place on the map to latitude and longitude.
 - Type in an address. Use the example in Walking and Biking in LA Teacher Resource or another that you choose
 - The appropriate numbers are under the map inside the parenthesis next to WKT: POINT(-118.2796304, 34.0916803). That means the location is 34.0916803 north of the equator and 118.2796304 west of the Prime Meridian. (If the example provided is used.) Note: Latitude is often referred to first, but the coordinates of the point are (longitude, latitude).
 - Have students try finding locations for their house and/or school.
 - Point out that the system of longitude and latitude will allow them to draw spatial objects such as points (house, school). They can think of longitude as the x-direction (it runs along the equator from east to west) and latitude as the y-direction (it runs from the North Pole to the South Pole along the Prime Meridian) when making a plot from these coordinates.
- Exploring LA Bike data and Deducer
 - Load the labike data file.
 - Point out that this is what they will be doing with the data they collect with the phones.
 - Point out the following features as you discuss the layout of the table shown in the data viewer (You can use Exploring LACBC and Deducer as a reference.):
 - Header, number of rows, categories
 - Note what you see in the data viewer (many of rows of data).
 - The first line is the header and describes the names of each variable or column.
 - Each **row** refers to a different intersection, and so there are 38 intersections represented in the data set. Each **column** refers to the various data that were collected about the intersection.
 - Navigate through the survey to show the variables in the data
 - set:
 - **"name"** is the location of the intersection,
 - **"longitude"** is the longitude of the location
 - **"latitude"** is the latitude of the location
 - **"type"** is the type of bike transportation available at the intersection (bike lane, bike path, bike route, none)
 - **"bike_count_pm"** is the evening count of bikes
 - **"ped_count_pm"** is the evening count of pedestrians
 - Demo how to obtain a table of frequencies for type.
 - The table appears in the Console window.
 - Note that 20 of the intersections have nothing.

- There will be more discussion of frequencies later in the unit, but note that this file is small enough that the counting could be done by hand (as with the data collected in Unit 2), but that later data sets will be much larger.
- Demo how to sort by bike count and pedestrian count.
 - Which intersections have the most bike traffic/ pedestrian traffic? Are they the same?
- Demo how to create a subset of locations where the bike count is greater than or equal to the pedestrian count.
 - Point out that the system provides a default title for the subset, but it is better to create a new title.
 - Ask students what other subsets might be interesting to create (e.g., locations with bike routes).
 - Have students create a few subsets of their own and list the questions they might want to ask about those subsets.
- Demo how to plot the intersections on a map. Include a title, axes and background. Also demo the various sizes and shapes of points and how to zoom.
 - Before each plot feature is added, ask students questions that will guide them to the need for the feature.
 - Ask students questions about the plot such as: Are there any outliers? Are there clusters of points? Does the plot match the table?
- LA Bike Activity
 - Have students complete the LA Bike Activity on their own. Circulate the room and answer questions.
 - Allow sufficient time at the end of this part to ask students for their responses and lead a discussion to ensure that they understand each of the features of Deducer discussed so far.
- LA Bus Stops Activity
 - Have students complete the LA Bus Stops Activity on their own. Circulate the room and answer questions.
 - Allow sufficient time at the end of this part to ask students for their responses and lead a discussion to ensure that they understand each of the features of Deducer discussed so far.
- Bubble charts with LA Bike Count Data
 - Describe bubble charts. (You can use Bubble Charts as a resource.)
 - Ask questions such as: What is being described when longitude and latitude is plotted on a map? Is there a way to distinguish counts of pedestrians and bicyclists?
 - Demo how to create a bubble chart with the pedestrian counts in the labike file.
 - Demo how to change the size and color of the bubbles.
- Bubble Chart Activity
 - Have students complete the Bubble Charts Activity on their own. Circulate the room and answer questions.
 - Allow sufficient time at the end of this part to ask students for their responses and lead a discussion to ensure that they understand each of the features of Deducer discussed so far.

Resources:

- Deducer Quick Start Guide
- Walking and Biking in LA
- Exploring LACBC and Deducer
- LA Bike Activity
- LA Bus Stops Activity
- Bubble Charts
- Bubble Charts Activity

Walking and Biking in LA

**Survey Description

The data that will be considered first in this lesson were collected in September of 2009 by the Los Angeles County Bicycle Coalition (LACBC, <http://la-bike.org>), a non-profit organization that works to "make the entire L.A. region a safe and enjoyable place to ride." For two days in late September, the LACBC recruited volunteers to count the number of bicyclists and pedestrians that pass 56 different intersections within Los Angeles County. Some of the survey locations were chosen because they are known to be popular with cyclists and pedestrians, others because they are near locations where a traffic-related change is about to take place, and still others because they are the site of a large number of bike accidents each year.

LACBC volunteers surveyed each location in the morning (7:00-9:30 am) and evening (4:00-6:30 pm) on Tuesday September 22 and Wednesday September 23 of 2009. Data were also collected on Saturday the 26th, but will not be considered here. The volunteers produced a report summarizing their findings (<http://lacbc.files.wordpress.com/2010/06/labikecountreport.pdf>).

** Describing location

When someone asks you to describe your current location, you might respond informally by saying you are in class or at school. Friends and family will know where that is, but if a relative were visiting from out of town and were unfamiliar with the area, the street address for school and maybe a nearby intersection would be necessary. These descriptions are excellent for looking up a location on a map or for walking, biking, or driving somewhere new. Roads and intersections and street addresses create a network that we regularly navigate. This network may change as old roads and buildings are replaced by others. Also, some of the important places in our lives do not have a street address (like the peak of a mountain or a hiking trail in the Santa Monica Mountains). Finally, in order to draw a map of LA, it would be helpful to be able to specify positions in a more consistent way, tracking a road as it turns a corner or veers to the left. For all of these reasons, there is a need to associate positions with a fixed set of "coordinates" on the earth.

One of the most popular such coordinate systems involves specifying a point's latitude and longitude. These are two numbers that represent angles (in degrees) from the center of the earth to a point on its surface. Latitudes are angles from north to south—in this case the North Pole is assigned a value of 180 degrees, the equator is at 0 degrees and the South Pole is at -180 degrees. Longitudes are angles from east to west with 0 occurring at the Prime Meridian, a line running from the North Pole to the South Pole and crossing Greenwich, England. (Greenwich is also used in defining Greenwich Mean Time or GMT.) A description of longitude and latitude can be found at the following url.

http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=Georeferencing_and_coordinate_systems

These two ways of describing a location (an address versus latitude-longitude) can be compared. The LACBC produced a Google Map with the locations of the 56 intersections for their survey—this was actually a recruitment tool and there are references to shifts that are needed for people to lend a hand.

<http://tinyurl.com/LABikeECS>

To translate the location on the map for the intersection of Sunset and Hyperion into latitude and longitude, a service like Get Lat Lon (<http://www.getlatlon.com/>) can be used.

In the box at the top of the page enter

Sunset & Hyperion, Los Angeles, CA

and the map should center on the right intersection. The latitude and longitude of the point will appear at the bottom of the page. In this case the latitude and longitude of the point is 34.0916803, -118.2796304—that is, 34.0916803 north of the equator and 118.2796304 west of the Prime Meridian.

The system of longitude and latitude allows people to draw spatial objects like points (house, school) or lines (a street) or shapes (the grounds of a high school, a park). Longitude can be thought of as the x-direction (it runs along the equator from east to west) and latitude as the y-direction (it runs from the North Pole to the South Pole along the Prime Meridian) when making a plot from these coordinates.

Exploring LACBC and Deducer

Load the labike data file into the Deducer Data Viewer.

It should appear similar to the table below.

name	longitude	latitude	type	bike_count_pm	ped_count_pm
1st & Alameda	-118.238125	34.049175	none	62	241
4th & Wilton	-118.313441	34.06713	bike route	48	87
7th & Figueroa	-118.259883	34.049388	none	216	1979
8th & La Brea	-118.344641	34.060446	none	72	272
9th & Pacific	-118.287306	33.735118	none	58	160

The first line of the file is known as a header and describes the names of each variable or column (e.g., "name", "longitude", and "latitude"). Each row refers to a different location at which volunteers counted the bike and pedestrian counts during the evening rush hour. ("Objects" in this table are positions in Los Angeles and the "variables" measured for each include the name, the longitude and latitude, and the counts of pedestrians and cyclists.)

The data frame "labike" has 38 rows, each referring to a different location. The first column is the name of the locations, similar to the list in the LACBC Google Map. (The data here reproduce Table 14 from LACBC's report and there are only 38 of the 56 locations included.) The next two columns give the positions' longitude and latitude. These coordinates can be used to place the locations on a map. The fourth column describes the type of bike transportation available at the intersection (a bike lane, a bike path, a bike route or nothing) and the last two columns represent the evening counts of bikes (column 5) and pedestrians (column 6) crossing the intersection.

A variety of operations can be performed on the data. (e.g., create a table that shows the frequency of type, sort by bike_count_pm, sort by ped_count_pm, create a subset of locations where the bike count is greater than the pedestrian count (bike_count_pm ≥ ped_count_pm)).

There are several kinds of spatial data. Their structure is best described by their look—that is, positions or points on the map (a house, the Sandstone Peak in the Santa Monica Mountains); points that are connected to form paths or lines (the route of a walk to school or the driving route to a friend's house); and areas or regions (the footprint of a school's buildings or the area covered by Los Angeles County). Points, lines and regions are basic spatial structures that are used for computation.

In the case of the LA Bike Count data, there are intersections where survey takers stood (points). The transportation system in Los Angeles can also be consulted for bus routes (lines), and the U.S. Census Bureau can provide demographic data about people living in different Census blocks (small geographic areas or regions).

Since the data set includes the longitude and latitude in columns 2 and 3, the intersections can be plotted on a map.

LA Bike Activity

Load the labike data file into the Deducer Data Viewer.

1. Create a subset of the locations with no special routes for bikes.
 - How many locations are in the subset?
 - Sort by `bike_count_pm`. Which intersection has the greatest count? Which has the least count?
 - Plot this subset on a map. Include a title, axes and a background.
2. Create a subset of the locations with special routes for bikes.
 - Use a different color and shape to plot this subset on the same map.
 - Describe any patterns you see.

LA Bus Stops Activity

Load the bus_stops data file into the Deducer Data Viewer.

1. What are the variables in this survey?
2. Form a frequency table to see the number of stops along each street. Which street has the most stops? What might be a reason for this?
3. How many total stops are there?
4. Look at the data for the 6000th row—a bus stop on Sunset Boulevard at Anita Avenue. Go to <http://getlatlon.com> and type in Sunset & Anita, Los Angeles, CA and check that the longitude and latitude listed in the data file are the same as from Get Lat Lon. What do you notice? Why might this be the case?
5. Create a subset of the bus stops that are along Sunset or Vermont. How many stops are there?
6. Create a plot of bus stops that are along Sunset or Vermont. Include a title, axes, and a background. Describe what you see in the plot.
7. Create a plot of bus stops that are along Myrtle or Mulholland. Include a title, axes, and a background. Describe what you see in the plot. How does this compare to the plot of Sunset or Vermont.
8. Create a plot of bus stops that are along Gayley or Hilgard. Include a title, axes, and a background. Describe what you see in the plot. How does this compare to the previous plots?
9. Create a plot of bus stops that are along a few streets in your neighborhood. Include a title, axes, and a background. Describe what you see in the plot. Why might this data be useful for someone to have?
10. What is an advantage to plotting the data on a map instead of just looking at the latitude/longitude numbers?
11. What would happen to the map if you had less data in the file? More data in the file? How would that affect your interpretation of the map?
12. If you were trying to make a case that you needed more bus stops in your neighborhood would it be enough to show that the count of bus stops is less than those along Sunset? Explain your answer.
13. How could you use what you learned about plotting points on a map with the data collected on the phone?

Bubble Charts

The points at which LA Bike volunteers stood and the position of bus stops exhibit the geometry of these things. However, spatial objects can have other data associated with them. The LA Bike Counts are associated with counts of pedestrians and bicyclists. When the intersections at which the LA Bike volunteers stood were plotted there was nothing that could be determined about the number of bikers or pedestrians.

A bubble chart uses numerical values to scale the diameter of circles located at a given spatial location. Consider the pedestrian totals from the LA Bike data.

In a bubble plot of the pedestrian counts each intersection where volunteers collected data is the center of a circle—the larger the circle, the greater the number of pedestrians counted there. If this plot were drawn by hand, a number of choices would need to be made. First, the size of the circles relative to each other is fixed by the data. If a volunteer at one intersection saw twice as many pedestrians as another volunteer saw at a different intersection, the first circle should be twice as big as the second. The relationship between the circles and the map, over which they are plotted, however, is not fixed and can be changed (again, assuming the relative sizes of the circles remains the same).

Bubble Chart Activity

Load the labike data file into the Deducer Data Viewer.

1. Create a bubble chart of the pedestrian counts.
2. Create a bubble chart of bike counts and add it to the pedestrian counts. What happens to the chart?
3. Change the color for bike counts. Describe what you see now.
4. What else might you change to get an even clearer visual picture of bike and pedestrian counts? Try these ideas. Explain how this changes the chart.
5. Based on your graph, what questions might you ask?
6. Try zooming in on a part of the graph. Describe what you see.
7. Does your graph make sense based on the counts in the table? Explain why or why not.
8. Create a subset of all locations that have a special route for bikes. Create a bubble chart with the counts for the subset. Describe what you see.
9. Create a subset of all locations that have no special route for bikes. Add a bubble chart of the counts for the subset to the previous chart in a different color. Describe what you see now. What conclusions might you draw? Justify your answer.
10. Create another pair of subsets that are of interest to you. Create a bubble chart. Describe the story you see.

Instructional Day: 11

Topic Description: In this lesson students use the data they have collected and additional contextual data sets to do spatial analysis for use in the final project.

Objectives:

The students will be able to:

- Analyze the data they have collected using spatial analysis techniques.

Outline of the Lesson:

- Create spatial plots with student generated and contextual data sets (55 minutes)

Student Activities:

- Groups create spatial plots and use spatial analysis techniques with the data they have collected and additional contextual data sets.

Teaching/Learning Strategies:

- Students work in their groups to analyze the spatial aspects of the data they have collected pulling in the additional contextual data sets as appropriate.

Resources:

- Student data
- Additional contextual data sets

Instructional Days: 12-14

Topic Description: Bar plots and the differences between categorical and continuous data are explored. Mosaic plots are introduced as a vehicle for comparing categorical data and looking for trends in data.

Objectives:

The students will be able to:

- Read and interpret a bar plot.
- Create bar plots.
- Differentiate between categorical and continuous data.
- Compare two categorical sources with mosaic plots.
- Look for trends by analyzing various plots.

Outline of the Lesson:

- Journal Entry (5 minutes)
- Birth Month Bar Plot (15 minutes)
- Experiment with bar plot commands (30 minutes)
- Public Agenda Bar Plot Activity (45 minutes)
- Journal Entry (5 minutes)
- Public Agenda data and mosaic plots (60 minutes)
- Wrap up Question (5 minutes)

Student Activities:

- Complete journal entry.
- Participate in Birth Month Bar Plot discussion.
- Experiment with bar plot commands.
- Complete Public Agenda Bar Plot Activity.
- Complete journal entry.
- Respond to questions during guided discussion.
- Complete questions in Public Agenda Data and Mosaic Plots Activity.
- Provide responses to the wrap up question and participate in discussion.

Teaching/Learning Strategies:

- Journal Entry: If everyone were going to be put in a different group based on the MONTH in which they were born, how many groups would there be? Which group do you think would have the most people?
- Birth Month Bar Plot
 - Tell students that you are going to create a bar plot (also called a bar graph or bar chart) of everyone's birth month to answer the journal question.

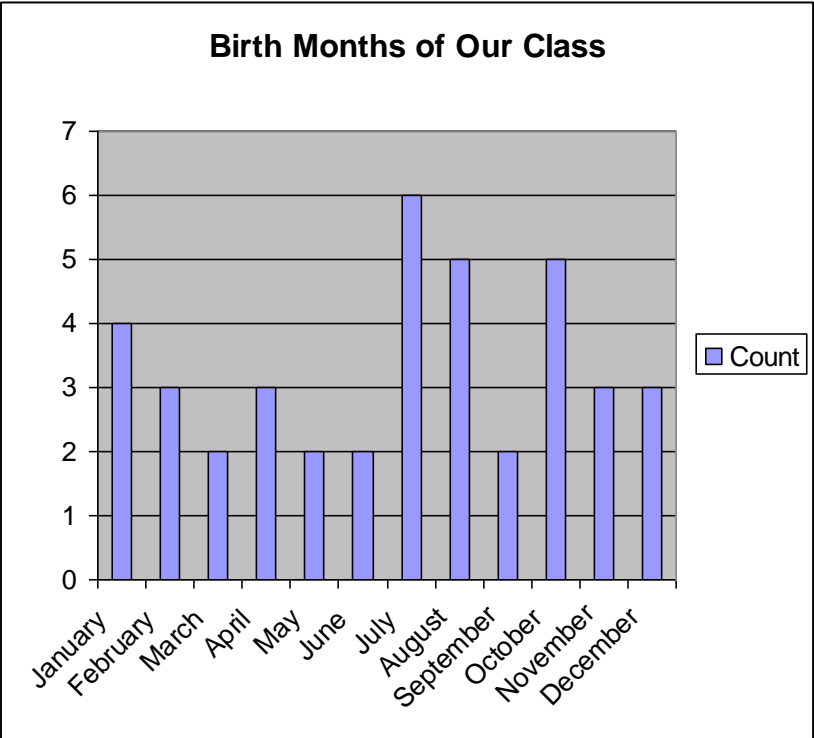
- Have students help you create the skeleton of a bar plot like Sample Birth Month Bar Plot. You should end up with a similar chart, but without any counts (bars).
- Ask each student what their birth month is. Increase the height of the corresponding bar by one until the entire class has responded.
- The bar plot should be used for categorical data only.
- Categorical data is expressed in terms of specific category values or labels (e.g., days of the week, answers to a multiple choice survey).
- Explain to the students that if we tried to do a bar plot of every student's exact height (example of 68.901 inches), we would most likely end up with a bar plot with a bar for every student each with a height of one. This type of data is quantitative (e.g., decimal numbers).
- Public Agenda Data and Bar Plots
 - Explain that the survey data file holds data collected by a private research group called Public Agenda (www.publicagenda.org). It is a survey of high school students and their parents designed to see if both groups have the same view of what's working (or not) with our schools. The people in the survey were identified by random selection from a list of all high school students in the United States. Respondents were asked over 100 questions—the file that will be used is a small subset.
 - Have students load the survey data file. Ask questions such as: How many different students are represented? (1293) How many different questions were asked of a student. (*Survey contains four (4) of the over 100 questions.*)
 - Ask: What are the variables?
 - "year" is their year in school
 - "effort" describes how hard they are working at doing well in school
 - "homework" describes their view of the amount of homework they are getting, and
 - "grades" records how well they said they are doing in school
 - Look at the Variable View in the data viewer. Here you see the type of each variable and possible values that are assigned to each variable. Factor is the type for categorical variables.
 - Demonstrate how to use the plot tool to create a bar plot. Point out that a bar plot is a graphical representation of the table and each bar should correspond to the count in the table.
 - Have students complete Public Agenda Bar Plot Activity individually.
 - Lead a discussion of the answers to Public Agenda Bar Plot Activity.
 - Each of the responses should generate a discussion beyond the simple solutions.
- Journal Entry: Do you think there is a relationship between grades and effort? If so, what type of relationship do you think grades and effort might have?
- Public Agenda Data and Mosaic Plots
 - Reload the survey data file, if necessary.
 - Demo looking at two variables at once with mosaic plots and guide a discussion with students.
 - Note that in the previous section bar plots about grades and effort were looked at separately.

- A good question to ask is “are the two related?”
 - Discuss the journal entry.
- Create a contingency table with data to show the relationship between the answers to the two questions.
 - The table will appear in the Console window. Ask students to explain what the items in the table mean. For example, there are 311 students that earned an A and are trying their best to do well in school. To represent this graphically, we can use a mosaic plot.
- Demo how to create a mosaic plot to graphically compare the 2 categorical variables grade and effort.
- How to interpret the mosaic plot:
 - The wider the columns, the more responses there are in that category.
 - Point out that the labels may not line up correctly.
 - Allow students time to respond individually to questions such as the following before discussing them as a group.
 - What grade is the most common?
 - What grade is the least common?
 - Does that reflect the numbers in the table?
 - Within each column, the taller the row, the more responses there are in that category.
 - Allow students time to respond individually to questions such as the following before discussing them as a group.
 - Within those students with A’s, are most of them trying their best or could they try harder?
 - Within those students with B’s, are most of them trying their best or could they try harder?
 - All the sizes are proportional to the numbers in the tables. So if twice as many respond a certain way, then the height would be twice as tall in the mosaic plot.
 - Looking at the mosaic plot as a whole, is there a trend? What story does it tell?
- Have students complete the Public Agenda Data and Mosaic Plots Activity individually.
- Discuss student responses and ask probing questions that will lead to discussion of the data.
- Wrap up question: Which items used when tagging events with phones are categorical?
 - Ask students to provide a response. Discuss their responses to make sure they understand the difference between categorical and quantitative data.

Resources:

- Sample Birth Month Bar Plot
- Public Agenda Bar Plot Activity
- Public Agenda Data and Mosaic Plots Activity
- Deducer Quick Start Guide

Sample Birth Month Bar Plot



Public Agenda Bar Plot Activity

1. Create a bar plot for effort.
 - Copy the plot to a document.
 - How does the effort of the students that responded compare?
 2. Create a bar plot for homework.
 - Copy the plot to a document.
 - How much homework did most students respond that they have?
 - How do you think that compares with students at your school?
 - If you think responses about homework are different from those at your school, why do you think students in this survey might have responded as they did? How could you test your assumption?
 3. Create a bar plot for grades.
 - Copy the plot to a document.
 - What grade is most common?
 - How do you think that compares with students at your school?
- If you think grades are different from those at your school, why do you think students in this survey might have responded as they did? How could you test your assumption?

Public Agenda Data and Mosaic Plots Activity

1. Create a contingency table with effort as the row and grade as the column.
 - How does this table compare to the one with grade as the row and effort as the column?
2. Create a mosaic plot with (effort, grades)
 - What do you see in this plot?
 - Compare your plot to the one done previously. Does it tell a different story? Justify your answer with features of the plot.
3. Try making mosaic plots with three different combinations of the available variables: year, effort, homework, grades. Choose one of these other plots, describe what you see, and explain what story it tells.

Instructional Day: 15

Topic Description: In this lesson students use the data they have collected and additional contextual data sets to create and analyze bar and mosaic plots for use in the final project.

Objectives:

The students will be able to:

- Analyze the data they have collected using bar and mosaic plots.

Outline of the Lesson:

- Bar and mosaic plots with student generated and contextual data sets (55 minutes)

Student Activities:

- Groups create bar and mosaic plots with the data they have collected and additional contextual data sets.

Teaching/Learning Strategies:

- Students work in their groups to analyze the data they have collected with bar and mosaic plots pulling in the additional contextual data sets as appropriate.

Resources:

- Student data
- Additional contextual data sets

Instructional Days: 16-18

Topic Description: In this lesson, the statistical measures of mean, median, minimum, and maximum are reviewed. Various ways to subset data are discussed and data is represented using box plots and histograms.

Objectives:

The students will be able to:

- Read and interpret a histogram.
- Create a histogram.
- Read and interpret a box plot.
- Create box plots.
- Explain mean, median, minimum, maximum.
- Create and query subsets of a data set.

Outline of the Lesson:

- Journal Entry (5 minutes)
- Sleep Histogram Activity (15 minutes)
- Quantitative Data and the CDC Survey Activity, Parts I-III (140 minutes)
- Wrap up Question (5 minutes)

Student Activities:

- Complete journal entry.
- Participate in Sleep Histogram Activity.
- Complete Parts I-III of the Quantitative Data and the CDC Survey Activity and participate in the discussions associated with the activity.
- Provide responses to the wrap up question and participate in discussion.

Teaching/Learning Strategies:

- Journal Entry: Thinking back to the bar and mosaic plots, why are we making graphical representations of our data instead of just using the numbers? Are their advantages to the different representations of data?
- Quantitative vs. Categorical revisited
 - Review the differences between these two types of data before starting on the Hours of Sleep Histogram.
- Sleep Histogram Activity
 - Create a histogram based on the amount of sleep each of the students got last night. (See Sample Hours of Sleep Histogram.)
 - Construct the bottom part of the histogram on the board, chart paper, or other display.
 - The lines should be right at the number of hours.

- Ask each student how much sleep they got last night.
 - For the most part, even if someone says they slept 8 hours that is an estimate. They really slept close to that amount (7.9 hours, etc).
 - The histogram creates groupings of the data.
 - To reinforce this idea, students cannot answer that they slept exactly 7 hours. They will have to choose either the 6-7 group or the 7-8 group.
 - As each student picks a group, increase the size of the corresponding column.
 - Emphasize:
 - Histogram is for quantitative data and not categorical.
 - Histogram is similar to the bar plot, but notice that there is no space between columns.
 - Histogram is similar to a bar plot, but counts are within a pre-constructed range of numbers.
 - In the CDC data they are about to explore, the hours of sleep question was asked in a multiple-choice format so the resulting data is categorical rather than quantitative. Therefore, they will need to use a bar plot rather than a histogram.
 - CDC Data
 - Explain to students that for the next few days they will be working with a survey of students by the Centers for Disease Control and Prevention (CDC). The CDC is part of the Department of Health and Human Services and addresses health issues facing students in the United States. (For more information: <http://www.cdc.gov/healthyouth/yrbs/>). CDC Data Subset Description provides the survey questions that explain the columns in the data set.
 - Part I—Getting familiar with the survey
 - Students should complete this on their own as teacher circulates.
 - Before going on to Part II, discuss Part I, demo how to find frequencies and descriptives, and conduct the discussion as indicated below.
 - Remind students that frequencies provides the number of responses for each option of a categorical variable. Show the frequencies for gender.
 - How many responses are there for each gender?
 - Is the sum of those the same as the total?
 - Point out N/A's.
 - Descriptives provide a numerical summary for a quantitative variable. You can include the minimum value (the shortest student's height), the maximum value (the tallest student's height), two measures of the "center" of the distribution, the mean (exact average) and median (if everyone was standing in order by height, the person in the middle) and the number of NA's or missing values. You can also include the 25th and 75th percentiles. Show the descriptive for height.
 - Explain that the heights are currently in meters, but that they will be converted in a later section. Discuss mean, median, maximum, and minimum. Make sure students understand what those terms mean in a general sense (They don't need to do the calculations.). In particular, ask

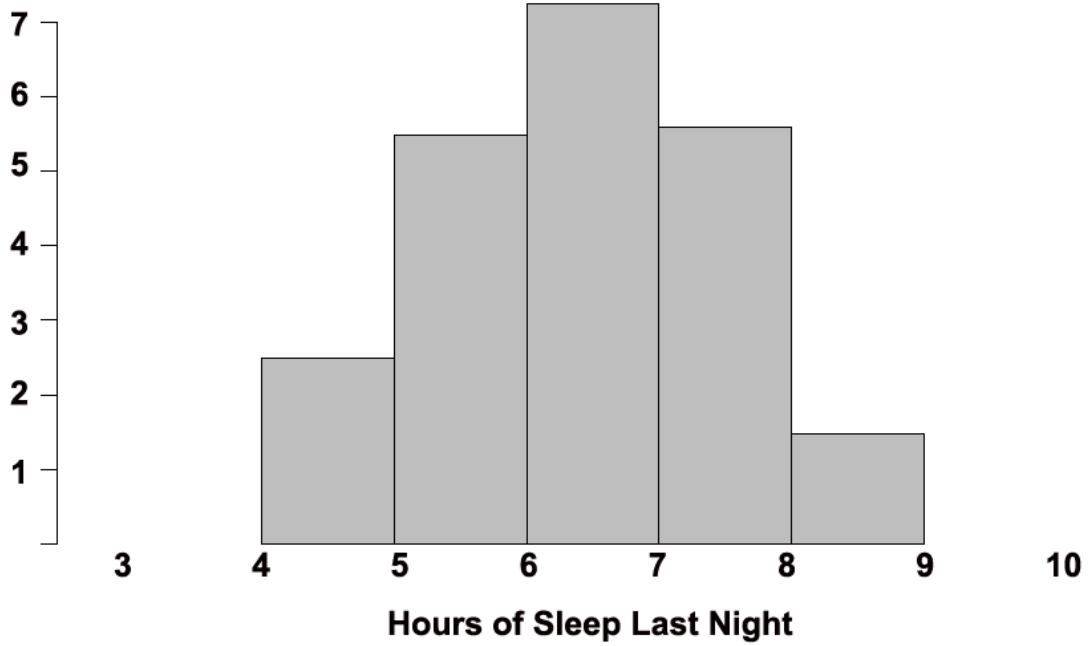
them when one measure might be better than another. For example, the mean is more sensitive to outlying data. Students may ask what 1st Qu. and 3rd Qu. refer to. Explain that it will be easier to discuss that when you start doing graphical representations.

- Part II—Subsets
 - Remind students how to create a subset and how to use more complicated conditions to subset by creating a subset of students that are “Female” **AND** “16 years old”.
 - Students should complete Part II of the activity on their own as teacher circulates.
 - Before going on to Part III, discuss Part II.
- Part III—Graphical Representations
 - Demo how to create a histogram of heights.
 - Discuss the plot.
 - Demo how to create a box plot of men’s heights.
 - Explain each piece of the box plot (median, middle half of people, maximum, minimum). Discuss each statistic and have students label their graph.
 - Box Plots can be placed side by side broken up by answers to a category. Demo a box plot of only the female’s weight on the left and only the male’s weights on the right. This shows a relationship between a quantitative variable (weight) and a categorical variable (gender).
 - Ask questions such as: According to the box plots, on average are females or males heavier?
 - Demo how to transform data. Show the transformation from meters to inches for height.
 - Note that the transformed variable appears at the end of the columns.
 - Students should complete Part III of the activity on their own as teacher circulates.
 - Distribute Different Types of Plots as a reference after completing all three parts so students can see the types of plots they have learned up to this point.
- Wrap up question: Which items used when tagging events with the phones are quantitative?
 - Ask students to provide a response.

Resources:

- Sample Hours of Sleep Histogram
- CDC Data Subset Description
- Different Types of Plots
- Quantitative Data and the CDC Survey Activity
- Deducer Quick Start Guide

Sample Hours of Sleep Histogram



CDC Data Subset Description

Data Names	Data Type	Question
Age	categorical	12 years old or younger; 13 years old; 14 years old; 15 years old; 16 years old; 17 years old; 18 years old or older
Gender	categorical	Male, Female
Grade	categorical	
Hisp_latino	categorical	Are you Hispanic or Latino?
Race	categorical	What is your race?
Height	double	Height in meters
Weight	double	Weight in kilograms
Helmet	categorical	When you rode a bicycle in the past 12 months, how often did you wear a helmet?
Seat_belt	categorical	How often do you wear a seat belt when riding in a car driven by someone else?
Fights	categorical	During the past 12 months, how many times were you in a physical fight?
Depressed	categorical	During the past 12 months, did you ever feel so sad or hopeless almost every day for two weeks or more in a row that you stopped doing some usual activities?
Days_Smoking	categorical	During the past 30 days, on how many days did you smoke cigarettes?
Describe_Weight	categorical	How do you describe your weight? (Very underweight, slightly underweight, about the right weight, slightly overweight, very overweight)
Eat_Fruit	categorical	During the past 7 days, how many times did you eat fruit? (Do not count fruit juice.)
Eat_Salad	categorical	During the past 7 days, how many times did you eat green salad?
Drink_Soda	categorical	During the past 7 days, how many times did you drink a can, bottle, or glass of soda or pop, such as Coke, Pepsi, or Sprite? (Do not include diet soda or diet pop.)
Drink_Milk	categorical	During the past 7 days, how many glasses of milk did you drink? (Include milk you drank in a glass or cup, from a carton, or with cereal. Count the half pint of milk served at school as equal to one glass.)
Days_Exercise_60	categorical	During the past 7 days, on how many days were you physically active for a total of at least 60 minutes per day? (Add up all the time you spend in any kind of physical activity that increases your heart rate and makes you breathe hard some of the time.)
Hours_TV	categorical	On an average school day, how many hours do you watch TV?
Hours_Videogame	categorical	On an average school day, how many hours do you play video or computer games or use a computer for something that is not school work? (Include activities such as Nintendo, Game Boy, PlayStation, Xbox, computer games and the Internet.)
Number_Teams	categorical	During the past 12 months, on how many sports teams did you play? (Include any teams run by your school or community groups.)

Asthma	categorical	Has a doctor or nurse ever told you that you have asthma?
Days_Exercise_30	categorical	On how many of the past 7 days did you participate in physical activity for at least 30 minute that did not make you sweat or breathe hard, such as fast walking, slow bicycling, skating, pushing a lawnmower, or mopping floors?
Days_Exercise_20	categorical	On how many of the past 7 days did you exercise or participate in physical activity for at least 20 minutes that made you sweat and breathe hard, such as basketball, soccer, running, swimming laps, fast bicycling, fast dancing, or similar aerobic activities?
Sunscreen	categorical	When you are outside for more than an hour on a sunny day, how often do you wear sunscreen with an SPF of 15 or higher?
Hours_Sleep	categorical	On an average school night, how many hours of sleep do you get?
General_Health	categorical	How do you describe your health in general?

Note: only two data types are “numerical”—height, weight

Reminder: R/Deducer describes categorical data with the word “factor” and a decimal value with the word “double”.

Source:

<http://www.cdc.gov/HealthyYouth/yrbs/data/index.htm>

2007_National_YRBS_Data_Users_Manual.pdf

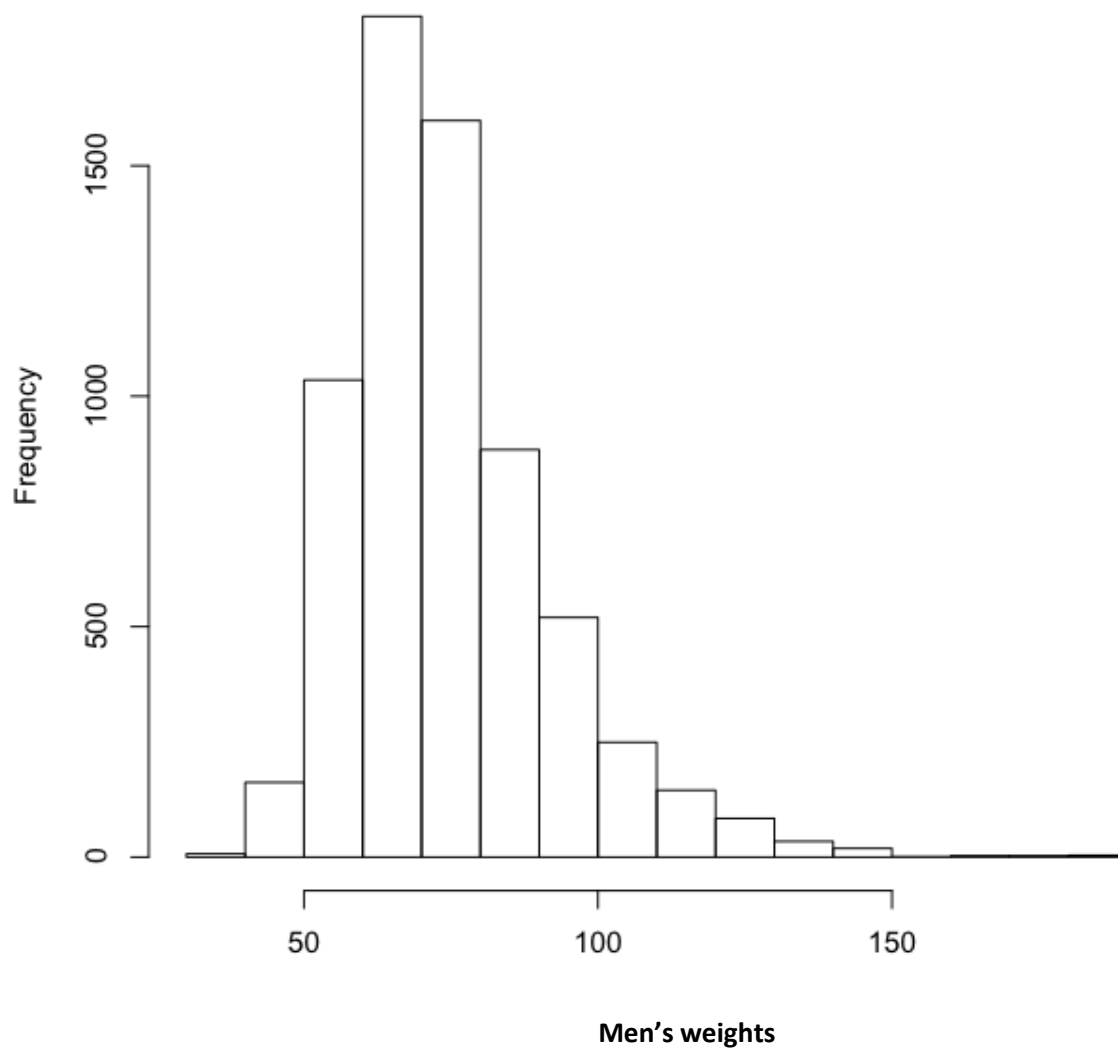
2007 National YRBS Data Users Manual

Different Types of Plots

Histogram (quantitative data)

What plot shows: How often a group of numbers occurs in a dataset. Each segment also represents its percentage of the entire data set.

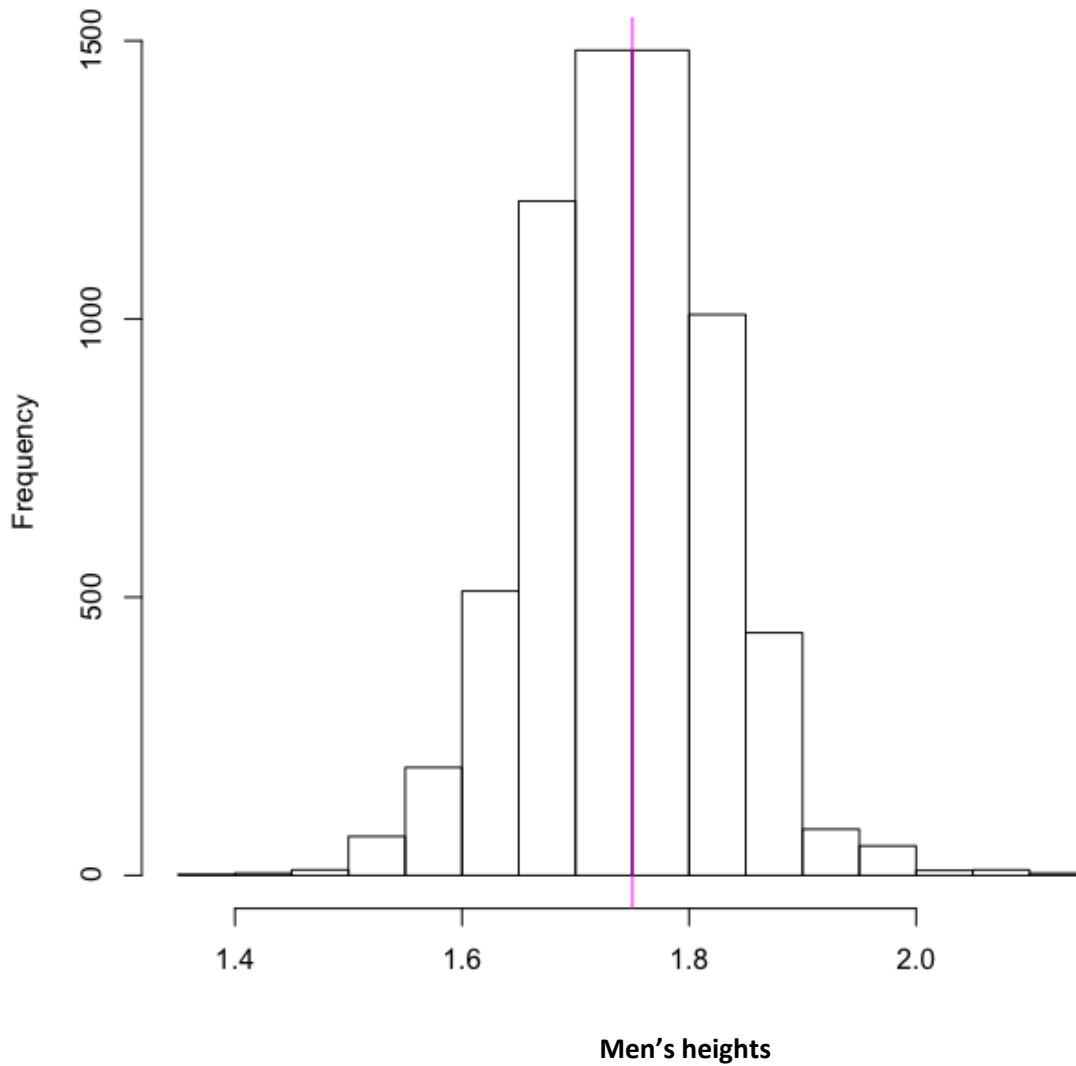
Example: Histogram of men's weights



Median (quantitative data)

What the plot shows: Median is the value that is literally in the middle—the point where half the data have larger values and half the data have smaller values.

Example: Median indicated on histogram of men's heights



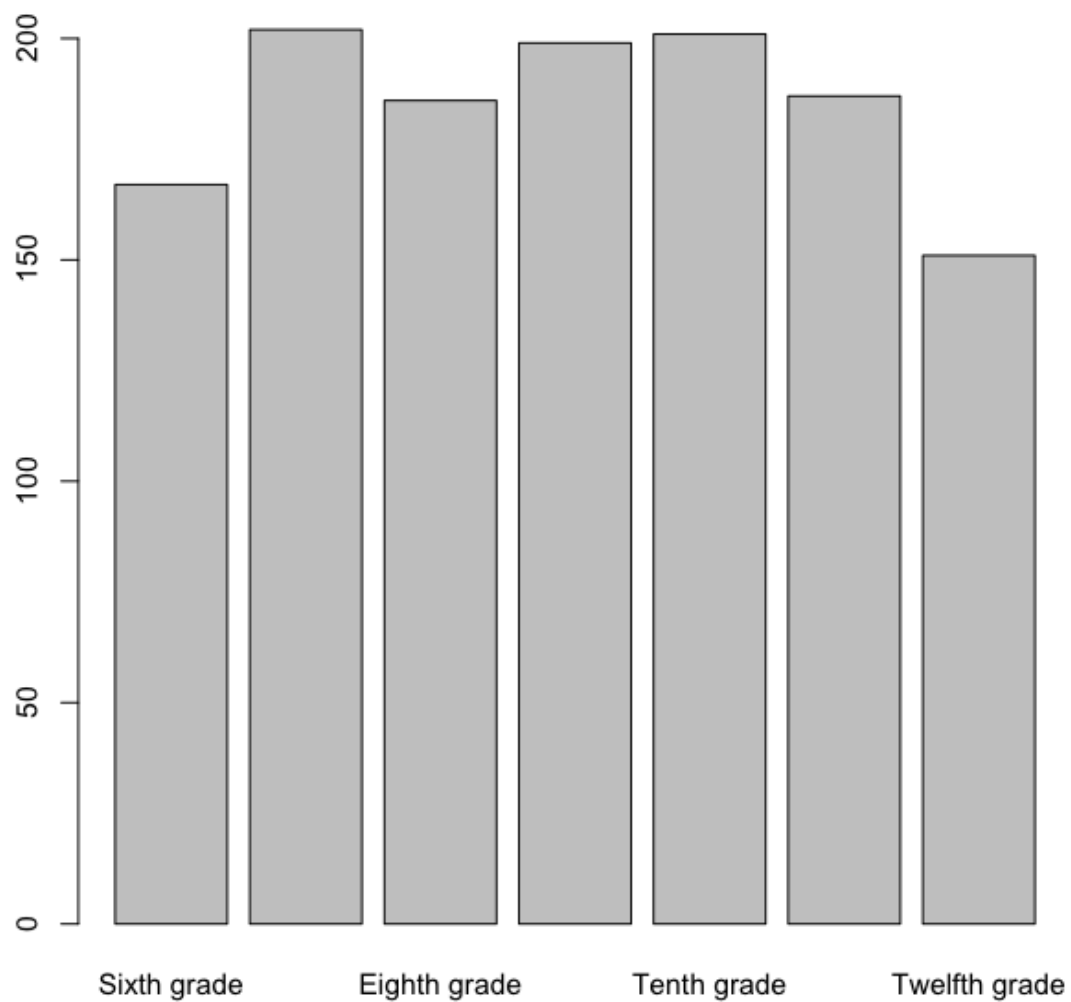
Bar Plot

What the plot shows: The number of occurrences within a given category

The data are quantitative (number of students) and categorical (grade level)

Ideal when categories are roughly the same size.

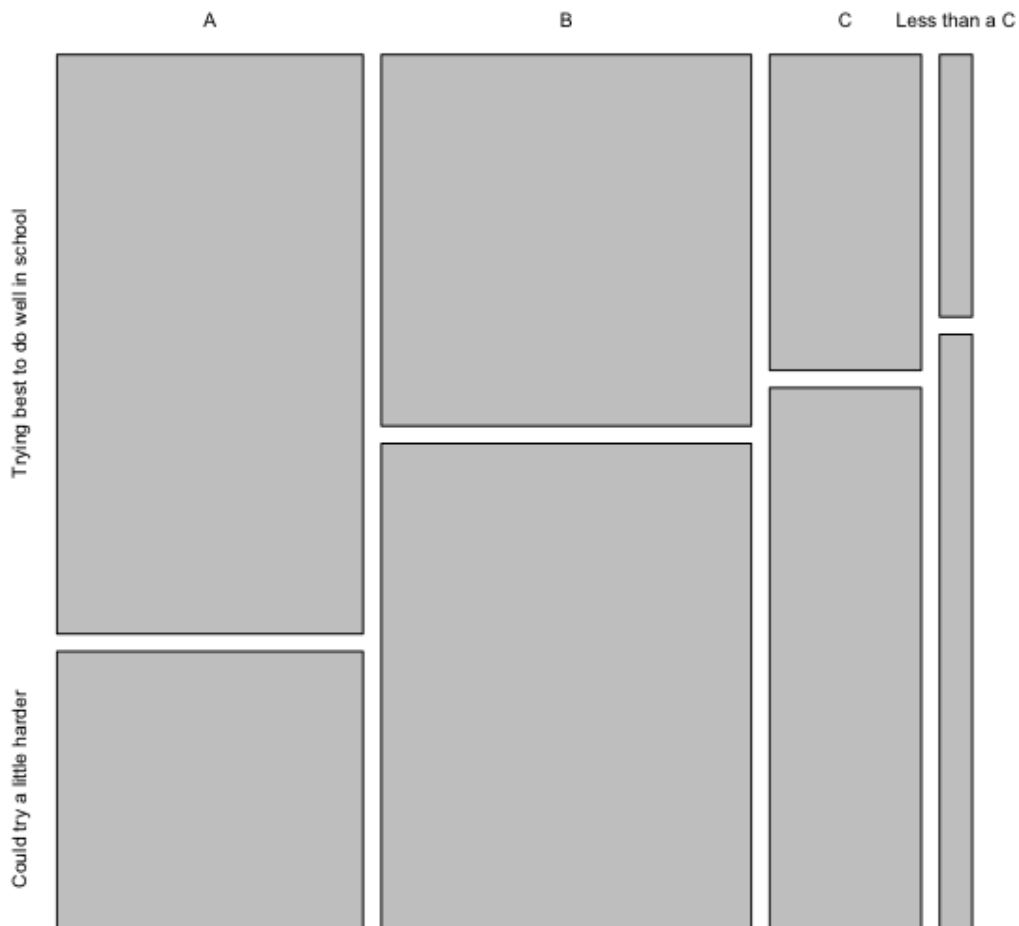
Example: Bar plot of years



Mosaic Plot – Sample 1 (categorical data)

What the plot shows: The possible relationships among categorical data.

Example: Mosaic plot of grades compared to effort



Mosaic Plot – Sample 2 (categorical data)

Sometimes interchanging the relationship of the values helps clarify or present different points of view.

Example: Mosaic plot with effort compared with grades

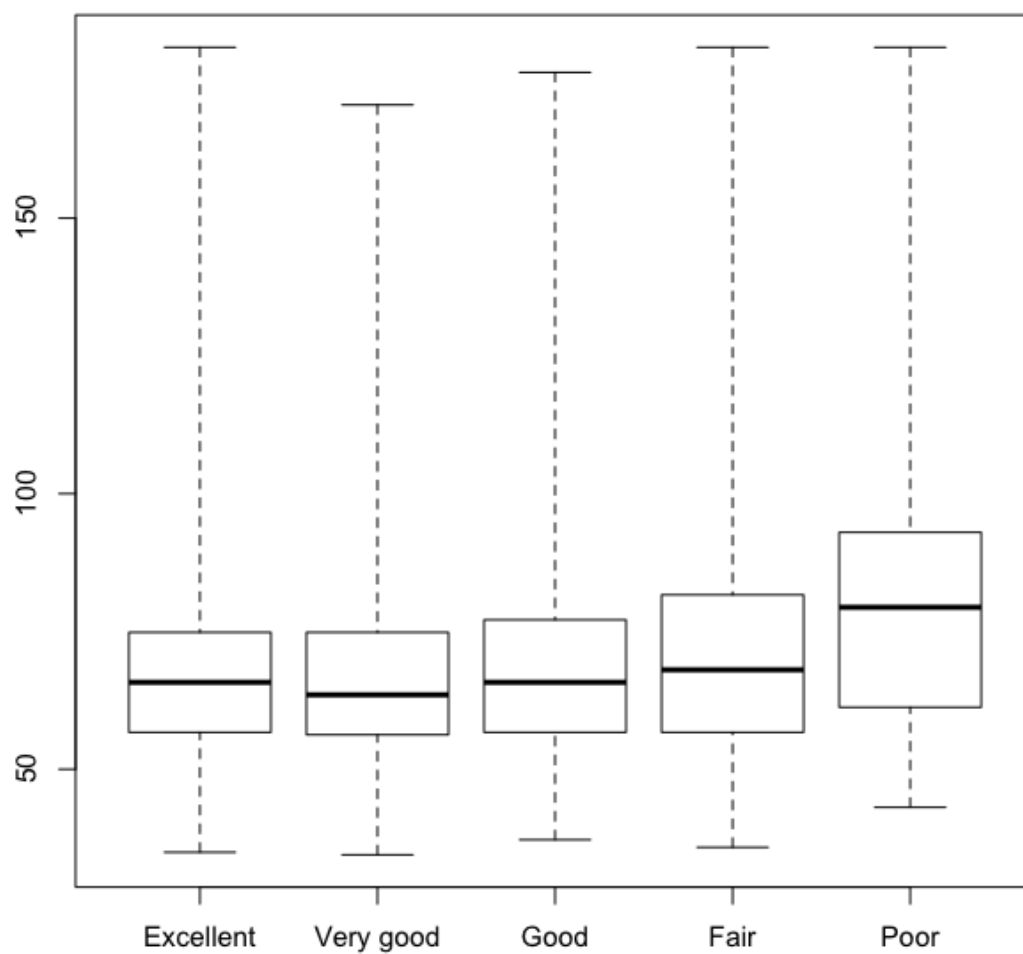


Box Plot

What the plot shows: A visualization of a range of values: the median (the thick black line), the upper and lower bounds of the values (the ends of the dotted lines), the 25th quartile (the lower part of the box), and the 75th quartile (the upper part of the box)

Data are categorical (general_health) and numerical (weight)

Example: Box plot of weight compared to general health



Quantitative Data and the CDC Survey Activity

Part I—Getting familiar with the survey

1. Load the CDC data file.
2. How many students responded to the survey?
3. What type of variable is gender?
4. What are the possible responses for gender?
5. What kind of data do you see for height?
6. What type of variable is height?
7. What type of variable is weight?

Part II—Creating subsets

Reminder that subset allows you to make a copy of just the data that meets certain requirements. For instance, you could separate out the responses for female students or students of a specific height.

1. Look at the age, gender, height and weight of everyone that has a height of 1.27
2. How many students are 1.27 m tall?
3. Save the subset of all Females as women.
4. Save the subset of all Males as men.
5. How many men are there?
6. How would you subset out the males that are 17 years old? How many men are in this subset?
7. What are the possible answers to general_health?
8. Create a subset of students that have a general_health of “Fair” **OR** “Poor”
9. How would you subset out students with excellent or very good health?
10. Why might subsetting be useful for your final project?

Part III—Graphical Representations

1. Make a histogram of men's weight. What command did you use? Paste the resulting plot into a document.
2. Add a line at the mean in magenta and a line at the median in cyan.
3. Why would you want to identify both the mean and the median on the same graph?
4. Create side by side box plots to compare weight and general_health? Paste the resulting box plots in a document.
5. What story do you see?

**** Transforming data**

1. Create a plot of men's height in inches.
2. What is the new mean height?
3. Transform weight from kg to pounds (1kg=2.2 lbs).
4. What is the mean weight now?
5. Write three questions that are interesting to you related to this data. Create plots of each and paste them into a document. At least one of them should be a Mosaic Plot comparing 2 categorical variables. For each plot write a description of the story it tells and how the data support the story.

Instructional Day: 19

Topic Description: In this lesson students use the data they have collected and contextual data sets to do statistical analysis with mean, median, maximum, minimum, and display information with a variety of plots for use in the final project.

Objectives:

The students will be able to:

- Analyze the data they have collected using statistical analysis and a variety of plots.

Outline of the Lesson:

- Statistical analysis with mean, median, maximum, and minimum with student generated data and contextual data sets (55 minutes)

Student Activities:

- Groups do statistical analysis with mean, median, maximum, and minimum using the data they have collected and additional contextual data sets.

Teaching/Learning Strategies:

- Students work in their groups to analyze the data they have collected pulling in the additional contextual data sets as appropriate.

Resources:

- Student data
- Additional contextual data sets

Instructional Days: 20-22

Topic Description: In this lesson, computation with text is explored. A variety of filters and queries are used to create subsets of text data. Bar charts are used for graphical display.

Objectives:

The students will be able to:

- Read in a file containing text as data.
- Filter a text data set (remove punctuation, remove case, remove stop words, stemming).
- Create a bar chart as one method of analyzing text.
- Create and query subsets of a text data set.

Outline of the Lesson:

- Introduction to text data (25 minutes)
- Introduction to Text Activity (30 minutes)
- Basic Analytics (20 minutes)
- Computing with Text Activity Part I (30 minutes)
- Journal Entry (5 minutes)
- Focusing on the words (20 minutes)
- Computing with Text Activity Part II (30 minutes)
- Wrap up Question (5 minutes)

Student Activities:

- Participate in discussion of text data.
- Complete Introduction to Text Data Activity.
- Complete journal entry.
- Complete Parts I and II of the Computing with Text Activity and participate in the discussions associated with the activity.
- Provide responses to the wrap up question and participate in discussion.

Teaching/Learning Strategies:

- Introduction to text data (You can use Introduction to Text Data as a resource.)
 - Explain secondary uses of data.
 - Provide Twitter background.
 - Ask students what words or phrases people may use to describe the first warm day after winter.
 - Look at Jillamore pdf.
 - Have students plot her location on a map.
 - Look at Weather Underground site.

- Have students search for Jillamore’s location.
 - Load the weather data file.
 - Navigate through the file.
 - Ask students to provide an explanation of what the variables are, etc.
 - Load the twitter data file. Navigate through the file.
 - Ask students to explain the variables, etc. Ask questions such as: What terms were searched for? Which had the most tweets?
 - Point out that this original file is very large, so to improve performance of the computers and make it easier for them to work with the file they will be using a subset. This is a good opportunity to remind students that computing is a powerful tool and allows working with large data sets, but is limited by things like processor speed and memory. (This can be linked back to the lessons in Unit 1.)
 - Demo how to create a subset of the data for a specific region of the country by picking latitude and longitude boundaries.
 - Load the twitterwithdate data file.
 - Point out the created date variable.
 - Have students complete the Introduction to Text Activity.
 - Discuss results.
 - Note that students may have some difficulty creating the subset expressions.
- Text Analysis
 - Part I—Basic Analytics
 - Text can be analyzed many different ways. Research areas like "stylometrics" attempt to say something quantitative about an author's work; e.g., by computing the average number of words per sentence or the average number of letters per word written by an author. Analytics can also be used to find patterns in other types of text. (You can use Computing with Text Background as a resource.) In the first part, the basics of counting words in a file and creating bar charts based on those counts will be addressed by working with the California subset of the twitter data file. Demonstrate how to do the following:
 - Load the CATwitter data file.
 - Look at the tweets.
 - Change the size of the column in order to view the entire tweet.
 - Scroll and look at the variety of tweets.
 - Text mining—analyzing word counts. Demonstrate how to do the following:
 - Turn the vector of tweets into a "corpus". A corpus is the term used to describe a collection of writings. This is necessary in order to do some more sophisticated analysis.
 - Demo creating the corpus.
 - This is a good opportunity to explain that the tweets are stored in an array (or vector) where the numbers in front indicate the place the tweet is in the vector. Arrays are an important concept in computer science. Storing items in an array allows us to access particular elements, search and sort.

- Demo how to view the corpus and point out that each of the array elements of the corpus matches the corresponding tweet in the data file.
- Create a frequency table that separates out each word and counts how many times it appears in all the tweets.
- Ask questions such as: What is the word that appears least frequently? What is the word that appears most frequently?
- Demo how to produce frequency tables that show only the most frequently appearing words and the different sorting options.
- Demo how to produce a bar chart of frequently occurring words.
- Journal Entry: What do you think would happen if you did all of these same things on the NJ subset?
 - Have students complete Part I of Computing with Text Activity.
 - Discuss results before going on to Part II.
- Part II—Focusing on the Words
 - Demonstrate how to do the following:
 - Remove case. Make “Spring” and “spring” be the same thing by making everything lowercase. (Note: each new corpus that is created should be assigned a new name.)
 - Removing "stop" words. Some words like “a” and “the” are probably always going to appear frequently because they are common parts of speech. Those words can be removed to emphasize the other less common words. Demo the method for removing stop words.
 - Deleting punctuation. Notice that many of the captions include symbols other than numbers and letters. Demo the method for deleting punctuation.
 - Stemming. It might be useful to ignore the ending of words such as “s”, “ing”, etc. In other words, change words like “boats” and “boating” to just “boat”. This is called stemming. Demo stemming.
 - Have students complete Part II of Computing with Text Activity.
 - Discuss results.
- Wrap up question: What is the source of the words that will be analyzed for your final project?
 - Ask students to provide a response. Make sure they understand that the answer is any of the text they enter that is “free text”.

Resources:

- Introduction to Text Data
- Jillamore.pdf
- Introduction to Text Activity
- Computing with Text Background
- Computing with Text Activity
- Deducer Quick Start Guide

Introduction to Text Data

****Secondary uses of data**

Data that are publicly available on the web are subject to a host of secondary uses. As the consumer of these data, there are a few questions to ask: Who collected the data and why were they collected? When was the data collected and how old is the data? What was their original purpose? What are the strengths and limitations of these data for your problem? How are the data organized? How do you access them? Is there someone who you can ask for help if you have questions?

****Twitter and the Jillamore file**

Twitter is a micro-blogging site that handled 4 billion messages or tweets in the first three months of 2010 alone. As a social network, Twitter culls activities from millions of people and there have been several studies of what people are posting to Twitter.

It is possible to look to Twitter and its users for signs of spring. Somewhere in the daily observations of millions of people it should be possible to find comments about the changing season.

The easiest place to start is with "Spring is here". On April 5, 2010 at about 1 pm a search for the phrase "spring is here" was submitted. *Jillamore.pdf* is a screen shot of the search results.

The last tweet was from the user "jillamore" who comments that it is a beautiful day in her part of the country, with temperatures in the upper 70s. She declares "Spring is here!" A bit more about this person can be learned from the last page in the *Jillamore.pdf* file. She lives at 40.360171 latitude-74.079609 longitude. The point can be plotted on a map. (Enter latitude and longitude into Google Maps, for example.)

This point is Red Bank, New Jersey. To get a sense of what the weather has been like in her part of the world, a service like the Weather Underground can be examined.

<http://www.wunderground.com>

It allows searches for weather anywhere in the country. This site is interesting to us both because it is possible to see what the weather has been like for jillamore and also because of where the data to make this judgment comes from. The Weather Underground culls data from about 10,000 officially run weather stations (e.g., the National Oceanic and Atmospheric Administration or NOAA) and 8,500 that are privately run but subject to strict data quality controls. The idea that citizens would install sensors and volunteer their data is very much in the spirit of the phone applications being used.

Search the Weather Underground for Red Bank, NJ. It shows historical weather data for this city. A file starting from January 1, 2010 to April 5, 2010, when the tweet about spring was posted was created from this information.

Load the weather data file and look at it in the Data Viewer.

Each row represents a different day and the names of the different variables recorded for each day appear in the first row.

**** Some historical data**

A group of researchers made hourly requests from the Twitter API using several different phrases in addition to "spring is here"—some relating to things turning green or trees beginning to bud. The researchers were also interested in the beginning of fall, so they also collected data on phrases like "fall is here" and comments about leaves turning colors.

Load the twitter data file and look at it in the Data Viewer.

Each row represents a single tweet. The variables include "created" which is a timestamp; Twitter "username" from the person who wrote the tweet; the "longitude" and "latitude" of their location (either their home or, if they are using a smart phone, the place where they typed in their tweet); which of the researcher's "search_term"s the tweet matched; and then the "message" itself. The variable "search_term" is a factor (categorical).

Load the twitter with date data file and look at it in the Data Viewer.

This file was created from the original twitter file and in addition to being a small subset; it includes a variable that indicates the created date in date format. (Note: to sort by date, the numerical created date needs to be used.) This file can be subsetted further by choosing latitude and longitude boundaries. For example, the New Jersey area would be approximately bounded by latitude between 38.5 and 41.5 and longitude between -75.5 and -73.5.

Introduction to Text Activity

Load the twitterwithdate data file and the weather data file into the Deducer Data Viewer.

1. Create a subset of the twitter data that includes only the tweets that contain "Spring is here".
 - How many tweets contain "Spring is here"?
 - What other search terms could you include that might indicate spring?
 - Create a subset that includes "Spring is here" and at least one other search term. How many tweets were added?
2. Create a subset of all tweets from approximately the New Jersey area.
 - Look at locations near where Jillamore lives. How many tweets are from that area? Do they match Jillamore's description? What other ways could you use to verify this?
 - Plot the New Jersey subset on a map. Experiment with different point sizes and zoom levels. What inferences can you make from the plot?
 - Sort by created. Do the dates and search terms correspond correctly?
 - Who has the most tweets? How does that impact the total number of tweets from New Jersey? How does that impact the number of tweets that include "Spring is here"?
3. Create a subset of California.
 - How many tweets are from that area?
 - Plot the California subset on a map.
4. What reasons can you think of to explain the difference in the number of tweets between New Jersey and California? How might you test your reasoning?

Computing with Text Background

A book usually has a fairly predictable structure. There are chapters which are made up of paragraphs which are made up of sentences which are made up of words. Research areas with names like "stylometrics" attempt to say something quantitative about an author's work. It is possible to compute the average number of words per sentence or the average number of letters per word written by an author. Some authors write in short, choppy sentences, while others craft sentences that are over a page long, adding phrase after phrase. Some authors choose simple vocabulary, while others prefer long, complex words. Statistics of this kind can not only point out interesting ways to think about the differences between authors, but they can even be used to help us figure out who wrote texts if their author is unknown or uncertain. One of the earliest analyses of this kind was of the famed Federalist Papers, a collection of documents describing the philosophy motivation behind our system of government. The papers are thought to be written by Alexander Hamilton, James Madison and/or John Jay. In the mid 1960s, a group of statisticians considered a number of novel statistics to differentiate the writing styles of the three men.

The counts of the different words in a document have also been used to characterize something about the document's subject.

The idea that the frequency with which words appear in a document might reflect something of its content has real-world applications. For example, the spam filter that intercepts junk e-mail is working on the frequency of words in each message. If a message makes too many references to "sales" or "won" or "Visa", there is a strong suspicion that the e-mail is spam.

The goal of this section is not to develop any of the topics above in any great depth. Instead, it will provide some basic tools for simple analysis on text.

Computing with Text Activity

****Part I—Basic Analytics**

Load the NJTwitter data file into the Deducer Data Viewer.

1. Create a corpus from NJTwitter.
2. View the corpus. Do the elements of the corpus match the messages in the NJTwitter data file?
3. Create a frequency table that counts how many times each word appears in all the tweets. What is the word that appears least frequently? What is the word that appears most frequently?
4. Create a frequency table that shows only the top 1% of frequently appearing words sorted by ascending frequency. What are the most frequently occurring words?
5. Create a bar chart of the most frequently occurring words. Does this match your frequency table?
6. Experiment with different percentages to determine the greatest percentage that allows you to read all of the words on the chart. Describe your process and the reasoning for your final answer.
7. Notice that there is a “spring” and a “spring!” (with an exclamation point). Do you think we should include those counts together or keep them separate? Why?

****Part II—Focusing on the Words**

Another data set was collected by using the API (Application Programming Interface) to conduct a search on Flickr for images that were tagged with the word "chill". The first 3,000 image captions in the list of search results were downloaded. Those captions containing words that were inappropriate were removed. In order to make it easier to work with the file, a subset was created.

Load the smallcaptions data file into the Deducer Data Viewer. Enlarge the column so that the entire caption can be viewed.

1. Change the list into a corpus. View several of the elements of the corpus. Do they match the file?
2. Create a frequency table for the entire list. What is the most frequently occurring word? Scroll to look through the entire file.
3. Run the frequency for only the top 10%. What does this do? Describe anything that you notice.
4. Run the frequency for the top 1%. Make a list of the words and their counts.
5. Create a bar chart for the top 1%. Describe what you see. Save the chart to a document.
6. Experiment with a few more % choices. Which gives the most information? Explain your choice.
7. Create a frequency table of the top 1% of words after making them all lower case and without punctuation. Make a list of the words and their counts.
8. Create a bar chart for the top 1%. Describe what you see. Save the chart to a document.
9. Experiment with a few more % choices. Which gives the most information? Explain your choice.
10. Create a frequency table of the previous corpus after deleting stop words. How did the file change?
11. Create a bar chart of the file without stop words. What are some of the words that disappeared from your bar chart? Why might it be useful to delete these stop words?
12. Create a frequency table of the previous corpus after deleting stems. How did the file change?
13. Create a bar chart of the file without stems. What are some of the words that disappeared from your bar chart? Why might it be useful to delete these stems?

Instructional Day: 23

Topic Description: In this lesson students use the data they have collected and additional contextual data sets to do text analysis for use on the final project.

Objectives:

The students will be able to:

- Analyze the data they have collected using text analysis techniques.

Outline of the Lesson:

- Analyze text in student generated and contextual data sets (55 minutes)

Student Activities:

- Groups do text analysis with the data they have collected and the additional contextual data sets.

Teaching/Learning Strategies:

- Students work in their groups to analyze the data they have collected with text analysis techniques pulling in the additional contextual data set as appropriate.

Resources:

- Student data
- Additional data set

Instructional Days: 24-26

Topic Description: Students complete final projects.

Objectives:

The students will be able to:

- Incorporate all objectives of the unit into the final project.

Outline of the Lesson:

- Review of final project expectations (20 minutes)
- Overview of rubric (15 minutes)
- Final project (~3 days)

Student Activities:

- Teams complete final projects.

Teaching/Learning Strategies:

- Review of expectations and overview of rubric
 - Discuss the rubric and answer questions.
- Final project
 - Teams work on final projects.
 - Help student teams with projects as necessary.

Resources:

- Final Project
- Final Project Sample Rubric

Instructional Days: 27-29

Topic Description: Students complete Scratch projects or websites to use with the presentation of their final projects.

Objectives:

The students will be able to:

- Incorporate all objectives of the unit into the final project.

Outline of the Lesson:

- Review of final project expectations (10 minutes)
- Overview of rubric (10 minutes)
- Final project presentation development (~3 days)

Student Activities:

- Teams complete final project presentation.

Teaching/Learning Strategies:

- Review of expectations and overview of rubric
 - Remind students of project expectations.
 - Discuss the rubric and answer questions.
- Final project presentation development
 - Teams work on final projects.
 - Help student teams with projects as necessary.

Resources:

- Final Project
- Final Project Sample Rubric

Instructional Day: 30

Topic Description: Students present final projects.

Objectives:

The students will be able to:

- Incorporate all objectives of the unit into the final project.

Outline of the Lesson:

- Final project presentations (55 minutes)

Student Activities:

- Teams present final projects.

Teaching/Learning Strategies:

- Final project presentations
 - Student teams present their findings to the class.
 - Other teams ask questions and participate in the discussion.

Resources:

- Final Project
- Final Project Sample Rubric

Final Project

Analyzing Your Data

In the course of the past few weeks, your group has collected data using the phone application. Now it is your turn to tell an interesting story based on the data. You will present your story to the class (it can be a series of web pages or a Scratch program). You must include plots/graphics that support the story.

You may include data from any of the other data sets you've seen in the lessons. However, these data cannot be the primary source of your story.

You will have access to data from your classmates as well as students at other schools that have also collected data. Keep in mind that you have already done some analysis on your data. This is your opportunity to pull it all together, modify as necessary and tell a compelling story that makes a case or highlights a discovery.

Final Project Sample Rubric

	Points Possible	Yes/No	Points Earned
Does your web page or Scratch program:			
1. Have a title with your group members' names?	5		
2. Tell a story based on your data? (Why does this data support your story?)	10		
3. Have 2 or more descriptive plots?	10		
4. Have other types of visuals?	10		
5. Have a description of why you chose the visuals that you did?	10		
6. Address how you can use this data to make a difference?	10		
8. Bring in data from an outside source that supports your story?	10		
Does your presentation include:			
How many items are in your data sets? Based on that how valid is your story? An explanation of this?	10		
An explanation of what you learned in this unit (analysis techniques, etc)?	10		
Web Page or Scratch program uses appropriate features for the medium	15		
Total	100		

Unit 6: Robotics



Introduction

Robotics provides a physical application of the programming and problem solving skills acquired in the previous units. The LEGO® Mindstorms NXT software uses drag and drop programming which will provide a natural transition from Scratch. Robots are shared by several students which will emphasize the collaborative nature of computing. In order to design, build and improve their robots, students will need to apply effective team practices and understand the different roles that are important for success.

Discussing the features of robots provides an opportunity to emphasize how computing has far-reaching effects on society and has led to significant innovation. Students can discuss such topics as:

- The effects innovations in robotics have had on people.
- The significance of processes that have been automated because of robots.
- How innovations in robotics have spurred additional innovations.

The unit consists of three main sections:

- The features of robots (Days 1-3)
- Familiarization with the robot and the software (Days 4-13)
- Robotics projects (Days 14-33)

Throughout the unit the similarities and differences between Scratch and the programming needed to move the robot can be highlighted.

Specific topics for each instructional day are listed in the overview chart on the next page.

Daily Overview Chart	
Instructional Day	Topic
1	What is a robot? Identify the criteria that make an item a robot.
2-3	Evaluate robot body designs and create algorithms to control robot behavior.
4	Set up LEGO® Mindstorms® NXT® kit.
5	Build robot base.
6-7	Introduce the features of NXT Brick—the “brain” of the robot.
8-9	Introduce the features of the Mindstorms NXT software.
10-13	Program the robot using the Mindstorm Robot Educator Software tutorials.
14	Introduce RoboCup real life robotic competition and write instructions for tic-tac-toe.
15	RoboTic-Tac-Toe Tournament and introduction to RoboCupJunior Dance Challenge.
16-18	Build, program, and present a dancing robot.
19-23	Build program and present a rescue robot.
24-33	Final projects and presentations

Daily Lesson Plans

Instructional Day: 1

Topic Description: “What is a Robot”? Identify the criteria that make an item a robot.

Objectives:

Students will be able to:

- List and explain the criteria that describe a robot.
- Determine if something is a robot, using the criteria.

Outline of the Lesson:

- Brainstorm about robot definition (10 minutes)
- “Kismet” video (5 minutes)
- Elements of a robot (10 minutes)
- Am I a Robot? Activity 1 (15 minutes)
- Student group work—Are we Robots? (15 minutes)

Student Activities:

- Brainstorm what they think of when they hear “robot” and then identify common features of robots.
- Participate in whole class activity determining if common items are robots.
- Work in small groups to complete “Are we Robots?” activity.

Teaching/Learning Strategies:

- Brainstorm: Ask students what they think of when they hear “robot”. Display responses. Responses may include the following:
 - Movie and TV robots such as Wall-E, iRobot, Robots, Rosie from The Jetsons
 - Modern industrial robots such as those involved in assembly-line factory work
 - Mars Rovers
 - iRobot robots, both the vacuum cleaner and the robots built for military use, other robots such as bomb detection and detonation
- View the video “Kismet” from Teachers Domain.
- Ask students if they can identify common features of the robots they have identified. What do all those robots have in common? What tasks are easy for robots? What tasks are hard for robots? (Answers: robots are often used for dangerous or repetitive tasks such as recovering bombs, search and rescue operations in dangerous conditions where the robots search and the humans rescue, factory work. They are replaceable, unlike humans, and don’t get bored or make

mistakes when doing the same thing over and over. Tasks that require judgment or human-like interaction such as recognizing when there is a problem or walking and seeing like humans are hard for robots. The two articles listed in the resource section provide more information and would be interesting for students to read.)

- Use the What is a Robot? handout to guide a discussion of robots.
- Hand out copies of Am I a Robot? activity, with the pictures of a basic stove and a fancy microwave. Check with students to make sure they recognize the items in the two pictures. Based on student input, display the five criteria for whether something is a robot: body, input, program, output, behavior. Note that what distinguishes a robot from a programmable device is the ability to respond to changes in the environment and adapt; robots respond to. Explain to the class that as a group you will figure out whether each of the two machines shown is a robot.
 - Go through the stove first. Ask students to figure out whether the stove meets the criteria for a robot:
 - Body—yes
 - Input—yes (dials to turn the burners off and on, set oven temp)
 - Programmable—yes, in the sense that oven temperature tells a sensor what temperature the oven needs to be heated.
 - Remind students that they programmed in Scratch and that the programmable aspect of the robot will require a language to provide the robot with instructions
 - Output—yes (heat!)
 - Behavior—yes, the oven responds by stopping at the desired temperature. It also adapts to changes as in opening the oven door, adding a frozen item, etc. by adding more heat to get back to the desired temperature.
 - Next go through the microwave in a similar way:
 - Body—yes
 - Input—yes (buttons)
 - Programmable—yes (buttons set time, set mode, microwave can be programmed by the user, for example “cook 3 minutes 50% power, hold 1 minute, cook 1 minute 90% power)
 - Output—yes (microwaves in chamber, light comes on)
 - Behavior—yes (cooks food, makes popcorn, boils water...)
 - Question: Does a microwave adapt?
- Hand out copies of Are we Robots? activity two. Explain the directions. Either have students brainstorm machines as a group to complete the table or have them think of machines on their own. Have students work in small groups to complete the table, determining whether each machine is a robot according to the criteria.
- Optional Extra Credit—have students research Isaac Asimov’s three Laws of Robotics. What are the three laws? What is law Zero? Why did he come up with these laws and how do they think

these laws affect our thinking about robots today?

Law Zero: A robot may not injure humanity, or through inaction, allow humanity to come to harm.

Law One: A robot may not injure a human being, or through inaction, allow a human being to come to harm.

Law Two: A robot must obey the orders given to it by human beings, except where such orders conflict with Law One.

Law Three: A robot must protect its own existence, as long as such protection does not conflict with Laws One and Two.

Resources:

- Jennifer Casper and Robin Murphy, Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center, IEEE Transactions on Systems, Man and Cybernetics 33:3, 2003, pp. 367-85.
- Robin Murphy, J. Kravitz, S. Stover and R. Shoureshi, Mobile robots in mine rescue and recovery, IEEE Robotics & Automation Magazine 16:2, June 2009, pp. 91-2003.
- What is a Robot? Handout (Based on handouts from The Big Picture “Robotics Teacher Guide 1” (Item #29852 from LEGO Dacta))
- Am I a Robot? Activity
- Are we Robots? Activity (Based on handouts from The Big Picture, “Robotics Teacher Guide 1” (Item #29852 from LEGO Dacta))
- http://www.teachersdomain.org/resources/eng06/sci/engin/design/lp_robot/index.html specifically
<http://www.teachersdomain.org/resources/eng06/sci/engin/design/kismet/index.html> (may require free registration)
- Asimov’s three laws of robotics: http://en.wikipedia.org/wiki/Three_Laws_of_Robotics,
http://www.asimovonline.com/asimov_FAQ.html#series13 , essay at
<http://www.sfwriter.com/rmasilaw.htm>

What is a Robot? Handout

There are many different kinds of robots, from ones designed to build cars to ones that vacuum to ones that explore other planets. To be a robot, a machine must meet certain criteria. A machine is only a robot if it has all the elements listed below:

Body

The body is a physical substance and shape of some type. The body will be designed based on the function—some look like vehicles, some like an arm, and some like a person. If you can touch it, that's the body.

Control

Control is a program to control the robot. Robots must be told what to do. To control a robot we need:

Input

Input is the information that comes from the robot's sensors. Robots have sensors that they use to get information from the robot's environment. For example, a smoke detector can detect smoke. (In other words, sensing the robot's environment). Robots typically have external and internal sensors.

Programmable

The program is a set of instructions or rules that the programmer gives the robot. For example, a smoke detector has a program to make a sound if it senses smoke. To be a robot, a machine must be programmable.

Output

The output is the action a robot takes, often involving motors, lights, or sounds. For example, a smoke detector makes a loud sound and might flash lights. (In other words, effecting change in the robot's environment—adapting.)

Behavior

Behavior is the combination of outputs that result in the task or job the robot does. For example, the behavior of a smoke detector is to "go off" in the presence of smoke. "Going off" is a combination of making noise and flashing lights, and may also involve calling the fire department.

Am I a Robot? Activity

Image 1: Basic Stove

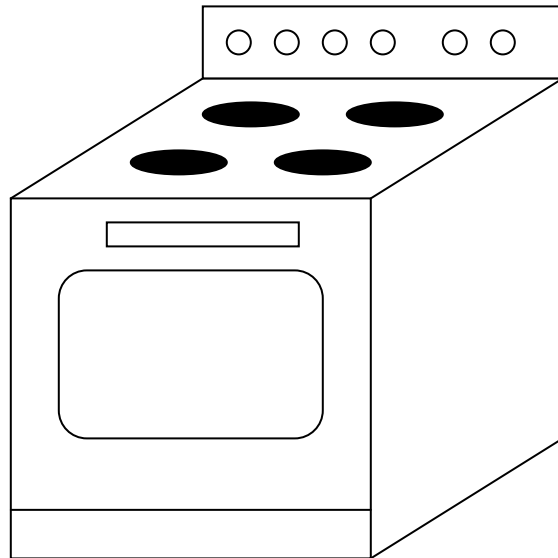
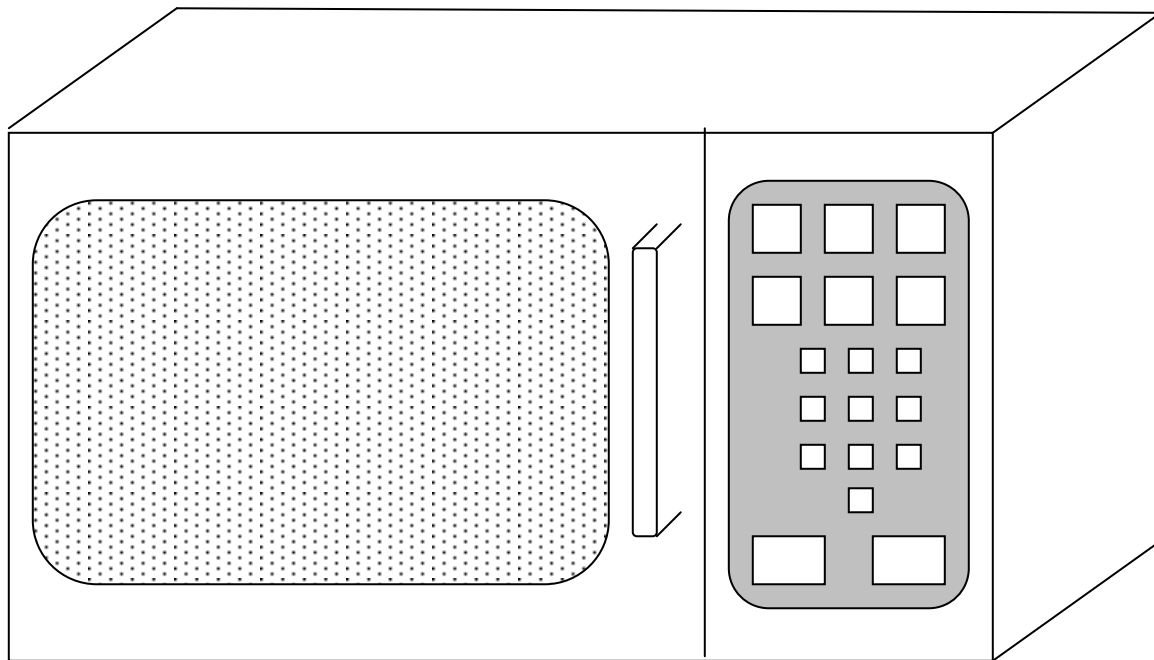


Image 2: New Microwave



Are we Robots? Activity

Instructions: Below is a list of machines that you may encounter in your daily life. Add machines to the bottom. Complete the table by deciding if the machine meets the criteria for being a robot. Then determine if the machine is a robot.

Body—physical form of some kind

Control—

Input—gets information from sensors, buttons, etc.

Program—Is programmable, follows a set of instructions you give it

Output—an action it takes

Behavior—what it does; the function it performs

	Body	Input	Program	Output	Behavior	Is it a robot?
Stove						
Microwave						
Radio						
iPod						
Flashlight						
Bicycle						
Car						
Alarm clock						
Traffic light						
Photocopier						
Computer						
Mars Rover						

Instructional Days: 2-3

Topic Description: Evaluate robot body designs and create algorithms to control robot behavior.

Objectives:

Students will be able to:

- Evaluate how the design of a robot’s body affects its behavior.
- Create an algorithm to direct a human “robot” from one part of the room to another.

Outline of the Lesson:

- “Are we Robots” activity (15 minutes)
- Journal Entry (5 minutes)
- The effect of changing design (15 minutes)
- Student group work—Can a robot tie your shoes? (40 minutes)
- Student group work—Walk like a robot (35 minutes)

Student Activities:

- Participate in discussion of “Are we robots” activity.
- Complete journal entry.
- Discuss how changing the design of an item affects the item.
- Students work in pairs to try tying a shoe in several robot-esque situations including with closed eyes, with tongue depressors, pliers, and with another person.
- Students work in small groups to direct a person to move along a path given a limited list of commands.

Teaching/Learning Strategies:

- Revisit “Are we robots” activity. Go through the list of items, asking students to indicate if they thought each item was a robot or not. Occasionally, especially if there is disagreement, ask students to defend their answer.
- Journal Entry: What happens when you change the design of a robot?
 - Have students share their responses
 - Ask students, “If you could change the body of the printer (or another device in the room) what would you change? How would that affect other things like the behavior or function of the printer, price, cost to build, or popularity? Have students share their ideas.
- Explain that there are limits to what robots can do because robots are limited by their bodies. For example, it is difficult to create a robotic hand that can grasp small or delicate items—it

would require many motors (simulating all the muscles in the hand) and many sensors to detect the item (simulating the neurons in the hand).

- Make sure each pair of students has a shoe that can be tied.
- Direct students to first try tying the shoe blindfolded or with eyes shut. Discuss how it went—Was it hard? What was hard about it? How was it like a robot tying the shoe?
- Direct students to tie the shoe with heavy gloves on. Discuss the experience. How was it like a robot tying the shoe? What made it hard?
- Direct students to tie the shoe with tongue depressors taped onto thumbs and forefingers or just holding tongue depressors. Discuss the experience. How was it like a robot tying the shoe? What made it hard?
- Direct students to tie the shoe with pliers. How was it like a robot tying the shoe? What made it hard?
- Direct the students to work with their partner to tie the shoes using the pliers, each person holding one pair. Discuss the experience. How was it like two robots working together? What made it hard?
- **Activity: Walk like a robot**
 - Choose one student to be a “robot” or tell students that you will be the robot. Choose a starting point and an ending point between which the “robot” must navigate. Make sure the path is not direct.
 - Tell the class that they must direct the robot from the starting point to the ending point using only five commands:
 - Turn left 90 deg.
 - Turn right 90 deg.
 - Take a step forward with the left foot.
 - Take a step forward with the right foot.
 - Stop.
 - Students can take turns or work as a group. The robot should only follow those five commands and not respond to other commands. Tell students to be careful with the robot and not walk it into walls or barriers. (The robot should stop before it hits a barrier such as a wall.)
 - At some point, remind students about loops. They can tell the robot to repeat a command or a block of commands such as “repeat: take a step forward with the left foot, take a step forward with the right foot until you are at the wall”
 - Point out that this is frequently what is done in dancing and choreography—sequences of steps are repeated.
 - If there is time, show the video of the “macarena” referenced in the resource section.
 - Conclude by pointing out that these kinds of commands are what they will be programming their robots to execute.

Resources:

- Activity: Can a robot tie your shoes? (From www.thetech.org/robotics/activities/page05.html)
- Materials: shoes that tie, tongue depressors, masking tape, heavy gloves, pairs of pliers, blind folds
- Walk like a robot activity from LEGO Materials.
- <http://www.cs.colorado.edu/~lizb/chaotic-dance/macarena-orig.mpeg.gz>
- Explanation of video: <http://www.cs.colorado.edu/~lizb/chaotic-dance.html>

Instructional Day: 4

Topic Description: Set up LEGO® trays.

Objectives:

Students will be able to

- Distinguish between the LEGO parts for building a robot.

Outline of the Lesson:

- Distribution of LEGO kits (10 minutes)
- Separation of LEGO parts into the appropriate compartments of the trays (45 minutes)

Student Activities:

- Student groups work together to set up their LEGO kits for use in building robots.

Teaching/Learning Strategies:

- Give each pair (or group of three) a LEGO® Mindstorms® NXT® kit. Point out the picture that shows where each item should be placed in the tray.
- Ask students to set up their trays so that they will be ready for use in building robots.

Resources:

- LEGO Mindstorms NXT kit

Instructional Day: 5

Topic Description: Build the base of the robot.

Objectives:

Students will be able to

- Assemble the base of the robot.

Outline of the Lesson:

- Explanation of LEGO Mindstorms manual (10 minutes)
- Assembly of base of robot (45 minutes)

Student Activities:

- Student groups assemble the base of the robot.

Teaching/Learning Strategies:

- Have students get out their kits and the manual that comes with the kit. Go through step 1 on p. 8 with the students to make sure they understand the format of the manual.
- Ask student groups to assemble the base of their robot according to the instructions on pp. 8-21. (Batteries should be charged in advance.)

Resources:

- LEGO Mindstorms manual

Instructional Days: 6-7

Topic Description: Introduce the features of the NXT Brick—the “brain” of the robot.

Objectives:

Students will be able to

- Distinguish between the parts of the NXT brick.
- Hook up input and output devices correctly.
- Use built-in NXT Brick programs.

Outline of the Lesson:

- Observation of the NXT brick (20 minutes)
- Plug in sensors, motors, and light, and run “View” programs (30 minutes)
- “Try Me” built in programs (40 minutes)
- NXT Brick programs (20 minutes)

Student Activities:

- Articulate what they observe about the the NXT brick while the teacher explains each part.
- Test sensor data using the ‘View’ programs and report observations.
- Run ‘try me’ programs and describe what the programs do.

Teaching/Learning Strategies:

- Have students get out their robot base, sensors, lights, motors, and three wheels. Explain that the NXT is the brain of the robot. Have students describe the parts they see and make sure the following parts get identified. (The NXT User Guide pp. 9-12 can be used as support.)
 - Ports number 1-4: these are input ports. You use wires to plug sensors into the NXT brick. There are four kinds of sensors: touch sensors (detect touch/obstacles), sound sensor (detect the sound levels), light sensor (detects light level), ultrasonic sensor (detects movement and distance to an object). Reminder: input means sensing something in the robot’s environment.
 - Ports A-C: these are output ports. You use wires to connect devices for output. The devices are lamps and motors. Also, note that the speaker is an output port. The output is that the light can go on or off and that the motor can turn or stop turning. Reminder: output means effecting change in the robot’s environment.
 - Buttons:
 - Orange button: On/Enter
 - Light grey arrows: Navigation, left and right

- Dark Grey button: Clear/Go back. Keep pressing this to turn off until prompt, and then hit orange
- Lines in the right side: speaker. This is where noise comes out of the robot.
 - If the rechargeable battery is in, there will be a power plug and LCD lights.
 - Tell students to turn on the robot by pressing the orange button. What happens? (It makes a happy little song. LOUD.) What do they see now?
 - NXT at the top—name of the brick. This can be changed in the software
 - Battery level top right.
 - Running icon—next to the battery icon. As long as it is spinning, the NXT is turned on and working correctly. If it freezes, the NXT has frozen and must be reset.
 - There are three icons on the screen. The one that is highlighted by default looks like two floppy disks and has the label above “My Files”. If they start hitting buttons, they can scroll through several menu options by using the arrow keys or go into My Files by hitting the orange button. The menu options are:
 - My Files—where programs will be stored.
 - NXT Program—allows you to build small programs using only the NXT without the need for a computer.
 - View—you can do a quick test of your sensors and motors and see the current data for each. You have to select the test you want to do and which port the sensor or motor is on. Only one test can be run at a time. The data will display on the screen.
 - Bluetooth—you can set a wireless connection between the NXT and other Bluetooth devices including other NXTs, phones, and computers.
 - Settings—you can change settings such as the speaker volume and the sleep time.
 - Try Me—built-in programs.
 - Explain that in order for the robot to really do anything, you have to hook up input and output devices. Ask students to try to identify the devices in the kit. Make sure they can identify the touch sensor, sound sensor, light sensor, ultrasonic sensor, servo motor and lamps. Reinforce that the sensors are all input devices and the motors and lamps are output devices.
 - Demonstrate and have students carefully plug the devices into the NXT. Sensors can be plugged into any input port numbered 1-4 but these default settings are used for the test and sample programs. See pp. 5-6 and 9 of the NXT User Guide for more information. Make sure students know to support the weight of the devices and the NXT brick without pulling on the wires.
 - Port 1: Touch sensor
 - Port 2: Sound Sensor
 - Port 3: Light Sensor
 - Port 4: Ultrasonic Sensor
 - Port A: Light

- Port B & C: Servo motor
- Have students navigate to the View menu. They should test each of the sensors and see what the displays do. Make sure they also use the motor rotations and motor degrees program. See NXT User Guide pp. 23-33 for more information. After a few minutes with students experimenting, ask what they noticed. What kind of data does each of the sensors provide? How could a robot use this in a program? Remind them that what they are doing is testing and debugging.
- Have students navigate to the Try Me menu by using the dark gray button to move up the menus and using the light gray arrows and orange button to enter the Try Me menu.
 - Select one of the programs and have all the students try it. Once they have tested it, ask them what it did. See if they can flowchart what the program does.
 - Try sound—moves the motors faster as more sound is detected.
 - Try touch—changes display and makes noise when button is touched.
 - Try light—makes noise based on how much light is detected.
 - Try ultrasonic—changes noise based on distance detected.
 - Try motor—changes sound based on motion of motor on port A.
- Finally, have students navigate to the Program menu and follow the directions in the LEGO Mindstorms manual on pp. 22-45, trying the programs indicated. They should then test the programs and make sure each one works as expected. (Also see the NXT User guide pp. 15-16 for more information.)

Resources:

- NXT User Guide

Instructional Days: 8-9

Topic Description: Introduce the features of the Mindstorms NXT Software.

Objectives:

The students will be able to

- Recognize the parts of the Mindstorms NXT software.
- Explain the different types of icons in the common palette and how to use them.
- Explain the different types of icons in the complete palette and how to use them.
- Explain the difference between software errors and hardware errors.
- Explain the difference between logical errors and syntax errors.

Outline of the Lesson:

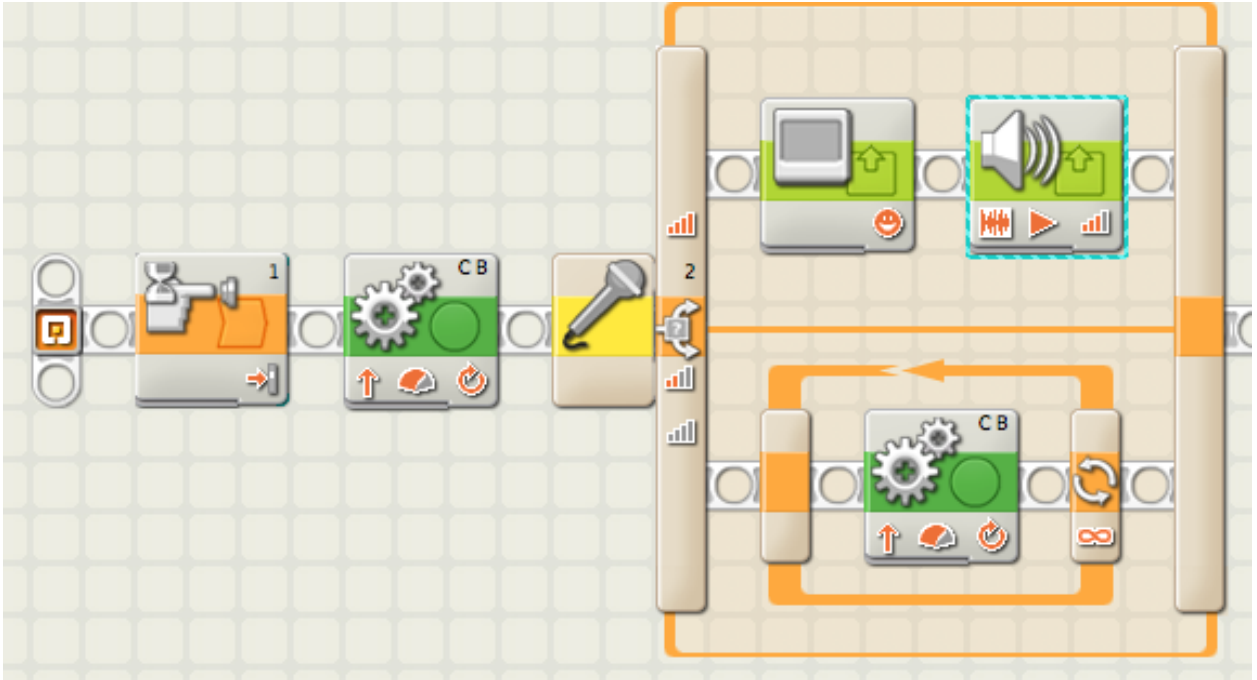
- Review of Program activity from Day 7 (20 minutes)
- Interface: the parts of the Mindstorms NXT software (10 minutes)
- A simple program from the common palette (30 minutes)
- A simple program from the complete palette (40 minutes)
- How to use the tutorials (10 minutes)

Student Activities:

- Discuss how the programs were created in the NXT brick and how they behaved compared to expectations.
- Listen to explanation of Mindstorms NXT software and respond to questions.
- Give ideas to teacher as s/he writes small programs in the software.
- Listen to explanation of how to use the tutorials.

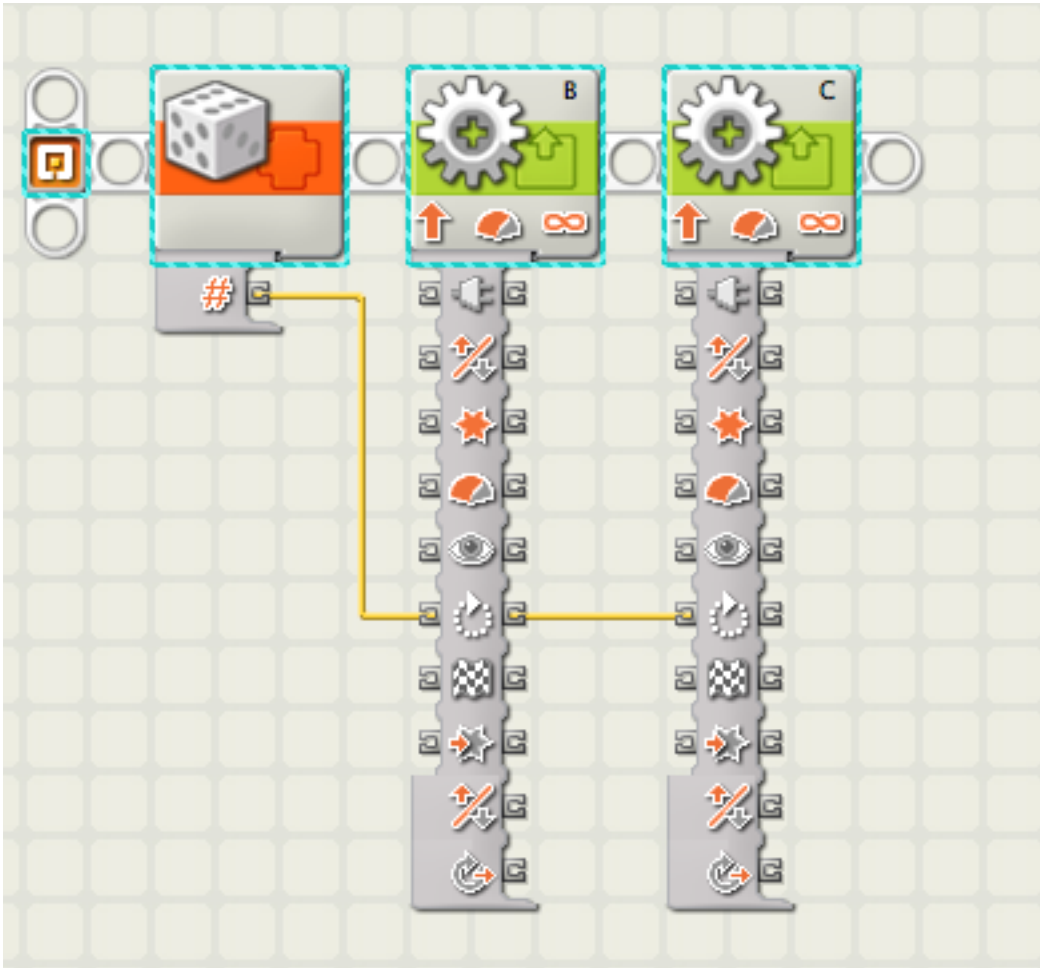
Teaching/Learning Strategies:

- Ask students what they programmed the robot to do. Get several answers. Did it do what they expected? Why or why not? Would it be a good idea to use the NXT Program interface to write all their programs? Why not? (It can only take 5 commands in a program.)
- Projecting the teacher's screen, launch the Mindstorms NXT software. Show the students where the tutorials are in the Robot Educator section and how to open a new program. Using the User Guide pp. 48-49, describe all the parts of the interface.
- With student input, use the common palette to build a small program. Ideally, use a variety of the blocks of the common palette, explaining what each one does as you use it. For example, if you wanted to build a program that told the robot to wait until the touch sensor was touched, then move forward for one rotation then listen and if a loud sound occurs, then display a smiley face and play a sound otherwise move forward, it would look like this:



- Save the program and download it to an NXT brick. Make sure the brick is set up to do the actions—have one built with the driving base and any necessary sensors. Demonstrate the running of the program.
- Modify the program and download it again. Try to make mistakes during this period and show how to debug the program by frequently testing it, downloading extra blocks, and also making mistakes such as having disconnected blocks. During this part have students try to work with the software themselves and follow along with you.
- Open a new program and switch to the complete palette. Show the differences in the two palettes. With student input, write a new program using the blocks of the complete palette. Show the differences in controlling the program. Make sure to show how to wire things in the data hub. For example, a program that runs the motors for a random amount of time would

look like this:



- Make sure to make mistakes and demonstrate how to solve problems with the software such as mis-wiring ports. Have students try these features at their seats as you do it. Point out the similarities between programming the NXT software and what they did in the last unit with Scratch.
- This is a good point at which to discuss
 - Software vs. hardware errors—in robotics the programming may be correct, but the robot configured incorrectly.
 - Syntax vs. logical errors—the program may compile, but the logic can still be incorrect.
- Tell students that the next five days will be spent going through the tutorials in order to learn how to build and program the NXT system.

Resources:

- NXT Robot Educator

Instructional Days: 10-13

Topic Description: Program the robot using the Mindstorm Robot Educator Software tutorials.

Objectives:

The students will be able to:

- Use the building blocks of the common palette to program the robot.
- Build robots that can execute the functions programmed through the Robot Educator Software.
- Program the robot using some or all of the complete palette of blocks.

Outline of the Lesson:

- Description of the assessment model (10 minutes)
- How to use the tutorials (10 minutes)
- Build and program robots according to tutorials (255 minutes)

Student Activities:

- In groups of 2-4, students follow tutorials to build and program small robots.

Teaching/Learning Strategies:

- Explain assessment model for tutorials. (Recommended: observe some but not all robots, such as those for tutorials 8, 16, and 20 in the common palette along with several from the complete palette; look at robot construction and the program as well as execution to determine grade.)
- All students should complete the tutorials for the common palette before moving on to the complete palette. It will be helpful for the future projects if students complete most, if not all, of the tutorials for the complete palette as well.
- Circulate throughout class to answer questions, help troubleshoot, and assess robots.
- If some groups finish early, have them assist other groups.

Resources:

- NXT User guide pp. 50-53 explain the tutorials

Instructional Day: 14

Topic Description: Introduce RoboCup real life robotic competition and write instructions for tic-tac-toe.

Objectives:

Students will be able to

- Explain how a sequence of game moves can be expressed in simple statements.
- Describe the RoboCup challenge and examine how robots have been programmed to play soccer.
- Develop if-then statements and use Boolean operators to direct a human “robot” to play tic-tac-toe.

Outline of the Lesson:

- Tic-tac-toe (10 minutes)
- “Robot Competitors Meet on a Soccer Field of Dreams” (25 minutes)
- Instructions for a “robot” to play tic-tac-toe. (20 minutes)

Student Activities:

- In pairs, students play a game of tic-tac-toe; then they discuss and write answers to the posted questions.
- Read and discuss the article, “Robot Competitors Meet on a Soccer Field of Dreams”.
- In pairs, students write a series of clear instructions for a “robot” to play tic-tac-toe.

Teaching/Learning Strategies:

- Before students enter the classroom, write the following on the board or chart paper: “Play a game of tic-tac-toe with your partner. Then think about these questions together, and write your answers: What are the rules of tic-tac-toe? What decisions does a player need to make before taking a turn? How would you verbally describe each of these decisions? What is the action a robot would need to take based on the decisions?”
- After a few minutes, have students share some of their responses. Make a list of the rules of tic-tac-toe on the board. Ensure students remember that if statements and conditionals are required to describe the moves of the game. Collect the written responses to the warm up activity.
- Distribute the article “Robot Competitors Meet on a Soccer Field of Dreams” and have students read it.
- Lead a discussion about the article.

- Explain to students that they will be working in pairs to write an application for human “robots” (students will act as the robots) to enable them to play tic-tac-toe. The following day will be the RoboTicTacToe Challenge. Remind them of the earlier discussion of tic-tac-toe. What goals does each player have? Who starts the game? Is there a “best place” to put the first X? What are some winning strategies for the next move? For example, If the X is in the center, then where should an O be placed? Why is “if-then” logic a good way to explain strategy for a simple game like tic-tac-toe? How can Boolean operators, and/or/not, help simplify the commands?
- Demonstrate the opening move for a game of tic-tac-toe on the board. Draw a nine-space grid and label the squares one through nine. Then ask students where to place the first X. Depending on where it is placed, have students create an if-then statement that determines the next move. For example, “If the first X is in the center, place an O in a corner square.”
- Ask students to complete the instructions. Each instruction in the entire sequence will cover every possible combination of moves the students can think of until a game is completed. Students need to remember that there are multiple options for each move (including the beginning move). They should consider all of the possibilities in developing their code. They also need to consider what the behavior the robot will exhibit based on the instructions provided.
- Note that the focus in this lesson is really a reinforcement of programming as a set of instructions in the context of something most students understand. The game of tic-tac-toe is not a natural example of robotics because robot environments are generally dynamic with infinite possible states of the environment.

Resources:

- Lesson plan from NY Times Lesson Plan Archive:
<http://www.nytimes.com/learning/teachers/lessons/20010802thursday.html>
- Copy of article
http://www.nytimes.com/learning/teachers/featured_articles/20010802thursday.html
- Dictionary

Instructional Day: 15

Topic Description: RoboTic-Tac-Toe Challenge and Introduction to RoboCupJunior Dance Challenge.

Objectives:

Students will be able to

- Debug conditional statements by testing them and compete as teams in a RoboTic-Tac-Toe Challenge.
- Describe dancing robots that have competed in the RoboCupJunior Dance Challenge.

Outline of the Lesson:

- Debugging of robotic-tac-toe statements (5 minutes)
- RoboTic-Tac-Toe challenge (35 minutes)
- Introduction to RoboCupJunior Dance challenge (15 minutes)

Student Activities:

- Complete debugging tic-tac-toe statements by testing that they work correctly in several games.
- Participate in RoboTic-Tac-Toe challenge.
- Listen to an explanation of RoboCupJunior Dance Challenge and watch videos of dancing robots from RoboCupJunior challenges.

Teaching/Learning Strategies:

- Ask students to quickly test their tic-tac-toe instructions to make sure they are complete and correct. They should play tic-tac-toe following only the instructions they have written.
- Explain the challenge: each team will be acting as a single robot “programmed” by the application they developed. One student will read a command from their application and the other student will execute the command. Teams play against each other, testing how successful their code is. Each game should be observed by the rest of the class and monitored to ensure the teams only execute the commands read.
- At the conclusion of the challenge, celebrate the winning team. Ask the students to describe why that team won? What have they learned? How would they improve their programs? (Remind students that precise instructions are required in programming.)
- Explain that RoboCup is a research initiative founded in 1997 by an international group of scientists interested in defining a common problem that could be addressed by researchers in robotics, engineering, and artificial intelligence. Most participants are university and industry research labs. RoboCupJunior (RCJ) was founded in 2000, with a focus on education. The RCJ Rescue challenge was piloted in 2001 and adapted in 2003. RCJ is open to students up to age 19. There are two divisions: primary, which is up to age 14, and secondary, which is age 14 to

19. The first two robot projects will be based on the RoboCupJunior program. The first one is the dancing robot which is the introductory level of the RoboCupJunior program. Students will build and program a robot that dances. Show videos of dancing robots in competition.

Resources:

- RoboCupJunior videos: <http://rcj.robocup.org/videos.html>
- More videos available through YouTube such as <http://www.youtube.com/watch?v=25sZr3u-WwU>

Instructional Day: 16-18

Topic Description: Build, program, and present a dancing robot.

Objectives:

The students will be able to:

- Use the NXT and output devices to build and program a robot that dances in time to music.

Outline of the Lesson:

- Explanation of project guidelines and show dance floor (15 minutes)
- Design, build, and program dancing robot (150 minutes)
- Dance challenge (30 minutes)
- Reflection and Clean up (25 minutes)

Student Activities:

- Agree on ideas and music for robot.
- Build robot.
- Write a program in Robot Educator software.
- Test robot and refine program and hardware.
- Participate in dance challenge and discussion.
- Complete project reflection. Take robots apart and put materials away.

Teaching/Learning Strategies:

- Hand out requirements and rubric. Explain guidelines and answer questions.
 - A dance floor can be made out of large square of one or more pieces of butcher paper.
- Circulate and make sure students are on task; answer questions as needed.
- Before the dance challenge, assign one student as timekeeper and another as DJ. Collect each group's program as they compete and immediately assess the robot using the rubric, while the next group gets set up. You may declare a winner or have the students vote for the best robot.
- Discuss the various dance routines and the features of each. Have students provide comments.
- At the end of the challenge, have each student complete the project reflection and submit it, then clean up the robots.

Resources:

- Official RoboCupJunior Dance Challenge rules (2008): <http://rcj.robocup.org/dance.html>
- Dancing Robot Activity
- Dancing Robot Sample Rubric

- Project Reflection

Dancing Robot Activity

The dancing robot assignment is based on the first level of RoboCupJunior, an international competition. More information about RoboCupJunior is available at <http://rcj.robotcup.org>.

Task:

Build a robot that dances to music for 1-2 minutes.

Requirements:

- The robot should not take any input, only have output in the form of various dance moves.
- Dance must be 1-2 minutes long. You have a total of 5 minutes to get set up, have the robot dance, and get out of the way for the next group.
- The robot must stay in the marked space.
- The robot must be autonomous. Other than hitting the start button, no human can touch it while it performs.
- The dance should be choreographed to the music you provide. The music must be appropriate for playing at school—no obscenities, etc.
- Teams may restart the robot up to 2 times at the discretion of the teacher. Any re-started, unless due to a problem not the fault of the team, will result in a grade penalty.
- Teams are encouraged to be as creative and entertaining as possible! Props, costumes, and varied dance moves are encouraged. You may dance alongside your robot.
- Each team must print out its program and hand it in at the same time that they compete.
- Fair play is an important part of the RoboCup competition. Teams are expected to help other teams as needed and not deliberately interfere with or damage other teams' work. All students are expected to respectfully watch all other teams compete.

Process:

1. Brainstorm ideas about how your robot should look, how it should work (wheels? Arms?) and how you'll build it. Select music.
2. Start building your robot.
3. Build a program that directs the robot to do your dance moves.
4. Test and revise the program. Make sure it runs for 1-2 minutes. Make sure it matches the music. Make sure it won't fall apart!
5. Show off the robot during the dance in class.

You will have two class periods to build and program the robot, and then you will present it on the third day.

Performance will be judged on

- Programming (e.g.: use of loops, jumps, conditionals, etc)

- **Choreography** (e.g.: robots to move in time with music, and change actions as music changes tempo or rhythm. Choreography of humans and robots will be scored separately, etc)
- **Construction** (i.e., robots should be of sound construction, components should not fall off , appropriate use of gearing, smooth and reliable operation, interesting movements, effective use of mechanics to achieve a purpose, etc.)
- **Entertainment Value** (i.e., How much does the performance entertain or delight the audience? Originality and creativity of the presentation, etc.)
- **Costume** (Costume of humans and robots will be scored separately.)
- **Cooperation between teams**

Dance stage will be a flat area. Official RobocupJunior stage size is 6X4 m.

Dancing Robot Sample Rubric

	Extra Credit	A	B	C	F
Programming	Program uses advanced techniques including blocks from the complete palette, flow blocks, etc.	Program is straightforward and efficient, using loops and parallel sequences as necessary. Program directs attached output devices to dance.	Program is straightforward and easy to understand. Program is inefficient and could use constructs such as loops.	Program is poorly written or difficult to understand. Program has unused parts or does not correctly control robot.	Program does not work.
Choreography	Dance has at least 10 different dance moves. Dance matched music precisely. Robot changed actions as music changed tempo or rhythm	Dance has at least 6 different dance moves. Dance is varied and entertaining. Dance is choreographed to match music	Dance has at least 4 different dance moves. Dance is repetitive. Dance lasted for 45-60 seconds or 120-150 seconds.	Dance has 3 different dance moves. Dance lasted for 30-45 seconds or 150-210 seconds. Dance did not match music.	Robot did not move or did not appear to dance.
Construction	Robot constructed using advanced gearing or other advanced construction techniques. Robot demonstrates extraordinary creativity.	Robot is of sound construction: nothing falls off, robot works as intended. Mechanics used well to achieve dance moves desired.	Robot dances as intended, but some extraneous parts fall off.	Robot does not work as intended, but does move. Robot falls apart. Very simple construction – mechanics not used well.	Robot falls apart or does not move at all. Construction appears careless or haphazard.
Entertainment Value	Presentation is unusually creative. Humans dance with robot. Costume, props, etc enhance robot.	Audience is entertained by robot, presentation, etc. Robot runs correctly the first time.	Presentation is not smooth: robot must be restarted.	Problems occur but robot does eventually run mostly correctly.	Robot does not compete.
Cooperation	Student(s) helped other groups	Student worked well with group. Student participated actively in all parts of project.	Student worked somewhat well with group. Student participated in most parts of project.	Student had trouble working with group. Student participated in few parts of project.	Student did not participate in project. Student sabotaged others' work.

Robot Project Reflection

For each member of your group, evaluate their performance as a team member:

Name: _____ Circle one word to describe his/her performance

Excellent Good Average Poor

Why?

_____ Excellent Good Average Poor

Why?

_____ Excellent Good Average Poor

Why?

What was your favorite thing about this project?

If you could do this project over, what would you do differently?

Instructional Days: 19-23

Topic Description: Build, program and present a rescue robot.

Objectives:

Students will be able to:

- Build and program a robot that uses input and output devices to count simulated people by following a black line and counting “people” on the path.

Outline of the Lesson:

- Explanation of project guidelines and floor (15 minutes)
- Design, build, program robot (195 minutes)
- Rescue Robot challenge (50 minutes)
- Reflection and clean up (15 minutes)

Student Activities:

- Brainstorm how to build and program the robot.
- Build the robot.
- Write a program in Robot Educator software.
- Test the robot frequently and refine program and hardware.
- Participate in rescue challenge.
- Complete project reflection. Take robots apart and put materials away.

Teaching/Learning Strategies:

- Hand out requirements and rubric. Explain guidelines and answer questions. Show students the arena with the victims laid out. Explain that they must use sensors so that the robot will follow the black line and will sense when it has encountered a victim or a gap.
- Circulate the room and make sure students are on task; answer questions as needed.
- During the rescue challenge, assign one student as timekeeper and one to keep track of victims found. Collect each group’s program as they compete and immediately assess the robot using the rubric, while the next group gets set up.
- At the end of the challenge, have each student complete the project reflection and submit it, then clean up the robots.

Resources:

- Rescue Robot Activity
- Rescue Robot Sample Rubric

- Project Reflection
- Official RoboCup Jr Rescue Competition Rules (2008): <http://rcj.robocup.org/rescue.html>
- Instructions for building modules are available at http://rcj.sci.brooklyn.cuny.edu/rcj2010/rescue_suggestedbuildinginstructions.pdf.

Alternatively use white butcher paper on the floor with black electrical tape as a path. Use green electrical tape to indicate victims.

Rescue Robot Activity

The rescue robot assignment is based on the second level of RoboCupJunior, an international competition. More information about RoboCupJunior is available at <http://rcj.robocup.org>. This robot simulates robots sent to rescue people during natural disasters. It must find “victims” along the path through each “room” and avoid obstacles. The goal is to program a robot that uses sensors to respond to different stimuli.

Task:

Build a robot that follows a black line on a white background, counts green or metallic “people” and avoids obstacles.

Requirements:

- The robot must follow the black line and attempt to complete the course through the entire arena. The robot will begin at the starting location in the doorway of the first “room”.
- The robot should stop and flash a light for at least two seconds to indicate it has found a victim. For extra credit, count the number of victims and display the count.
- The robot should be able to avoid items of debris blocking the black line.
- If a robot has been stuck or lost the black line for more than 20 seconds, the teacher may pick it up and put it back onto the black line a little beyond where it ran into problems. The 20-second rule allows it to try to find its way back to the line without intervention. A team may decide to quit if the robot is faulty or repeatedly loses the line.
- Robots must be controlled autonomously except for being started by a member of the team.
- The robot will have 10 minutes to complete the course and identify all victims.
- Each team must print out its program and hand it in at the same time that they compete.
- Fair play is an important part of the RoboCup challenge. Teams are expected to help other teams as needed and not deliberately interfere with or damage other teams’ work. All students are expected to respectfully watch all other teams compete.

Process:

6. Brainstorm ideas about how your robot should work: what sensors will you need? What motors and lights? What programming constructs will you need?
7. Start building your robot.
8. Build a program that controls the robot
9. Test frequently and revise the program. Make sure it correctly detects victims and that it can follow the line. Check if it can navigate gaps.

You will have three and a half class periods to build and program the robot; then you will present it in class.

Official Rules available <http://rcj.robocup.org/rescue.html>

Official RoboCupJunior Rescue Challenge

5.1. Victims:

5.1.1. Ten (10) points are awarded for each victim located by the robot. The robot indicates that it has found a victim by stopping and flashing a lamp for at least two (2) seconds.

5.1.2. Extra points are NOT awarded for the same victim being located more than once.

5.2. Gaps in the black line:

5.2.1. Ten (10) points are awarded for each gap in the black line that the robot successfully negotiates (i.e. recovers the line on the far side of the gap).

5.3. Debris blocking the black line:

5.3.1. Ten (10) points are awarded for each item of debris blocking the black line that the robot successfully avoids (i.e. moves around the debris and recovers the line).

5.4. Rooms:

5.4.1. Ten (10) points are awarded for each room that the robot navigates successfully (i.e. enters through one doorway and exits through the other doorway).

5.5. Ramp:

5.5.1. Thirty (30) points are awarded for the robot successfully negotiating a ramp without any assistance.

5.6. Penalties:

5.6.1. Two (2) points are deducted for each false victim identification (i.e. whenever a robot indicates that it has found a victim at a location where there isn't one).

5.6.2. Five (5) points are deducted for each lack of progress (i.e. whenever human intervention is required to enable a robot to resume progress along the black line).

Official Rules available

<http://rcj.robocup.org/rcj2008/china-rescue-rules-page.pdf> (Note: This references the RoboCupJunior 2008 Rescue rules. The committee members were Ashley Green, Maverick Luk, Eli Kolberg and Bill Freitas. You may wish to work with the most up to date version.)

Rescue Robot Rubric

	Extra Credit	A	B	C	F
Victims	Found victims are counted and count is displayed	All victims correctly identified	Most victims correctly identified	Some victims correctly identified	No victims correctly identified
Gaps		All gaps navigated correctly	Most gaps navigated correctly	Some gaps navigated correctly	No gaps navigated correctly
Debris		Robot avoided all debris	Robot avoided most debris	Robot avoided some debris	Robot unable to avoid debris
Rooms		Robot entered all rooms through one door and exited through the other	Robot entered most rooms through one door and exited through the other	Robot entered one room and was unable to exit	Robot did not enter the first room
Construction	Robot constructed using advanced gearing or other advanced construction techniques. Robot demonstrates extraordinary creativity.	Robot is of sound construction: nothing falls off, robot works as intended.	Parts of robot fall off. Very simple construction – mechanics not used well.	Robot does not work as intended, but does move. Robot falls apart. Robot is unable to navigate due to construction	Robot falls apart or does not move at all. Construction appears careless or haphazard.
Programming	Program uses advanced techniques including blocks from the complete palette, flow blocks, etc.	Program is straightforward and efficient, using loops and parallel sequences as necessary. Program uses sensors and strong logic to navigate challenges and find victims.	Program is straightforward and easy to understand. Program uses inefficient logic to navigate challenges and find victims.	Program is poorly written or difficult to understand. Program has unused parts or does not correctly control robot. Program does not correctly use sensors to control motion.	Program does not work.
Cooperation	Student(s) helped other groups. Managed own role & helped group members.	Student worked well with group. Student participated actively in all parts of project.	Student worked somewhat well with group. Student participated in most parts of project.	Student had trouble working with group. Student participated in few parts of project.	Student did not participate in project. Student sabotaged others' work. Made it difficult for group to work.

Instructional Days: 24-33

Topic Description: Complete Design Challenge final project.

Objectives:

Students will be able to:

- Design, build, and program a robot that solves a stated problem.

Outline of the Lesson:

- Explanation of project guidelines (15 minutes)
- Distribution of challenges (10 minutes)
- Design, build, and program robot (~7.5 class periods)
- Design challenge gallery walk and discussion (1 class period)
- Clean up (1 class period)

Student Activities:

- In groups, determine who will complete each of the four roles.
- Use the planning document to plan the robot.
- Design, build, program, and refine a robot which meets the challenge.
- Set up their robot and participate in a gallery walk.
- Disassemble the robots and carefully organize all the robotics equipment.

Teaching/Learning Strategies:

- Hand out requirements, planning document, and rubric. Explain guidelines and answer questions.
- Hand out challenges. Allow students to trade challenges as necessary. You may choose to have each group working on a different challenge or have them overlap.
- Approve planning documents as students finish plan and prepare to build and program robot.
- Circulate and make sure students are on task; answer questions as needed. At the end of each day, remind information specialists to fill out paperwork and remind groups to clean up the space. Optionally, have students fill out the daily group evaluation.
- During the design challenge, fill out each rubric as you observe the robot. If possible, videotape (or have a volunteer videotape) the running of each robot. Discuss the features of the various robots and designs.
- On the final day of the unit have students disassemble the robots and organize the equipment.

Resources:

- Design Challenge Sample Rubric
- Information Specialist Report
- Project-Reflection
- Daily Group Evaluation
- Challenges:
 - Option 1: Challenges from Design Challenges for computer-controlled LEGO products by Len Litowitz. (Litowitz-challenges.doc) Some of these challenges are more appropriate than others.
 - Option 2: Gary Stager’s LEGO Challenges available from <http://www.stager.org/LEGO/challenges.pdf> (stager-challenges.pdf) Not all of these challenges are appropriate.
 - Option 3: Webquest

Final Project

Design Challenge Planning

STEP #1 TASK DEFINITION

Determine the purpose of your challenge—What are we supposed to do?

Criteria—list the specifications the robot needs to meet

1.

2.

3.

4.

5.

STEP #2 TASK BREAK-DOWN

List the steps the robot will need to go through to accomplish the task.

1.

2.

3.

4.

5.

6.

7.

8.

9.

STEP #3 BRAINSTORMING

List some possible solutions to the challenge.

1.

2.

3.

4.

5.

6.

7.

8.

STEP #4 ROBOT DESIGN

Use scratch paper to sketch ideas for the robot, and then choose the “best” design idea and illustrate it NEATLY below. Include any labels or explanations necessary to make your design understandable.

STEP #5 PROGRAM FLOWCHARTING

Outline the programming steps for your robot to accomplish the task. This can be in the form of a chart or graph.

STOP!!! – GET TEACHER APPROVAL BEFORE MOVING ON: _____

STEP #6 ROBOT BUILDING AND PROGRAMMING

Build the robot and program it according to your plan!

Design Challenge Rubric

	Extra Credit	A	B	C	F
Successful Solution	Meets criteria and one or more super challenge criteria	Solution clearly solves the problem but not super challenges.	Solution solves problem inelegantly or inefficiently.	Solution does not completely solve problem.	No reasonable attempt made to solve problem.
Programming	Program uses advanced techniques including Boolean logic, Complete palette blocks, etc. Program demonstrates extraordinary creativity or unique way of solving problem	Program is straightforward and efficient, and uses appropriate programming constructs. Program has a reasonable algorithm for solving problem and uses good logic.	Program is straightforward and easy to understand. Program is inefficient. Program has a reasonable algorithm for solving problem.	Program is poorly written or difficult to understand. Program has unused parts or does not correctly control robot. Algorithm is strained.	Program does not work. Program does not solve problem effectively.
Construction	Robot constructed using advanced gearing or other advanced construction techniques. Robot demonstrates extraordinary creativity.	Robot is of sound construction: nothing falls off, robot works as intended. Mechanics used well to achieve desired outcome. Robot can solve problem repeatedly.	Robot works as intended, but some extraneous parts fall off. Moderate degree of repeatability: robot will run again but must be adjusted or fixed.	Robot does not work as intended, but does move. Robot falls apart. Very simple construction – mechanics not used well. Robot cannot run repeatedly.	Robot falls apart or does not move at all. Construction appears careless or haphazard.
Documentation	Documentation goes beyond required paperwork.	Ample and accurate documentation. Documentation kept consistently and thoroughly.	Good documentation: documentation kept consistently but not as thorough as it could be.	Fair documentation: documentation kept inconsistently and missing parts.	Little or no documentation
Cooperation	Student(s) helped other groups	Student worked well with group. Student participated actively in all parts of project.	Student worked somewhat well with group. Student participated in most parts of project.	Student had trouble working with group. Student participated in few parts of project.	Student did not participate in project. Student sabotaged others' work.

Daily Group Evaluation

Date: _____

List each member of your group (including yourself) and assess each area with:

3 = strongly agree (s/he was very good at this)

2 = agree (about right)

1 = disagree (this was a problem)

<i>Name</i>	<i>Listened respectfully to group members</i>	<i>Was focused and on-task</i>	<i>Did his/her share of work</i>
(self)			

Comments:

Information Specialist Report

You are responsible for reporting the status of the project to the Team Manager every day. How has the team progressed? Address the following questions:

1. What did your team accomplish today?
2. What problems did the team find today?
3. What solutions did the team try?
4. Other comments?

	Tasks	Report
Per 1	Get Challenge Begin brainstorming & Designing	
Per 2	Finish Design & get approval Begin building test parts—try different ideas	
Per 3	Finish building test parts & begin assembling robot from successfully tested parts	
Per 4	Continue assembling robot from parts Create program for robot	
Per 5	Continue building & programming robot—test regularly	
Per 6	Continue to refine robot—test regularly with the program	
Per 7	Finish refining robot—make sure it completes challenge!	
Per 8	Finish or enhance robot	
Per 9	Design Challenge: Show off robot!	
Per 10	Clean up: Take apart robot, return materials to original state	

Names: _____