BALL STATE UNIVERSITY – CS4MS

Indiana State Computer Science Academic Standards

4/9/19

http://www.cs.bsu.edu/cs4ms/docs/StandardsResources.pdf

Introduction (Scope)

CS4MS (Computer Science for Middle Schoolers) is an immersive learning class at Ball State University in Muncie, Indiana, held twice per week throughout the Fall 2017 and 2018 semesters. CS4MS aims to find answers to the essential question: “How can we increase exposure and access to computer science and computational thinking for ALL middle school students?” The following compilation contains resources and explanations of Indiana state standards to assist teachers to successfully implement those standards.

Contributors

Fall 2017 Class

Monica Appel
Meghan Duffy
Rachel Harvey
Anna Hawkins
Ben Lawson
Ryan Magley
JR Pegg
Ian Pemberton
Jordan Reidy
Tim Skinner
Cody York

Fall 2018 Class

Ryan Ahler
Luke Betts
Austin Bolles
Michael Bratton
Will English
Josh Passey
Alexander Perry
Sarah Phipps
Adam Wessel
Morgan Williams

David Largent, Faculty Mentor

References

This document builds upon information that is located on the Indiana Department of Education’s website, which is located at the following address:
https://www.doe.in.gov/standards/computer-science-resources-6-8

Table of Contents

This page has been intentionally left blank (except for this statement).

6-8.DI.1 - Algorithmic Problem-Solving

Short Description
Use the basic steps in algorithmic problem-solving to design solutions (e.g., problem statement and exploration, examination of sample instances, design, implementing a solution, testing, and evaluation).

Long Description
The students should be able to use algorithmic problem-solving skills to design a solution for a given problem. This strategy consists of five big steps: read and comprehend the problem statement, select theoretical concepts that may be applied, qualitative description of the problem, formalization of a solution strategy, and test and description of the solution. They should then be able to test and examine their proposed solution and be able discuss it with their classmates.

Explanation For Teachers:
An algorithm is essentially a series of instructions used to complete a task. People use algorithms to complete a wide range of tasks, specifically as the basis for writing computer programs. Students should learn how to use symbols to represent algorithms.

Purpose
Students will benefit from recognizing patterns that will lead to success across all disciplines and implement problem-solving processes in everyday life.

Objectives

- Concerning algorithms:
    - Design a solution
    - Explore the problem
    - Examine sample instances of the problem
    - Implement the solution
    - Test a solution
    - Evaluate the solution

Proposed Activities

- Steiner Trees (1)
- Computational Thinking (2)
- How To TEACH "Algorithmic Thinking" (3)
- Sorting Socks Algorithm Complexity (4)
- Board Game Instructions (Additional Activities)
- Lego Structures (Additional Activities)
- Gym Activities (13)

Alternative Approaches

- Without computers, you could make worksheets with a simple problem on it and have the students write an algorithm for it.
- *My Robotic Friends* This game involves group work and problem solving and can be done without computers. (It could also apply to CD.2, CD.4, PA.2, PA.3, and/or NC.1) (5)
- Explore the Codebreakers of Bletchley Park and the Enigma Machine used during World War II. (This also applies to IC.2).
- Use Alice Programming as part of story-telling in English or Social Studies courses (Also applies to DI.5, PA.3, NC.1, NC.2)

Resources

- Algorithmic Problem Solving (6)
- Basic Strategy for Algorithmic Problem Solving (7)
- Alice Programming
- Compound Conditionals Powerpoint (12)

Hyperlinks

1. http://csunplugged.org/steiner-trees/
2. https://studio.code.org/s/course3/stage/1/puzzle/1
3. https://codinginmathclass.wordpress.com/2014/09/10/how-to-teach-algorithmic-thinking/
4. http://tryengineering.org/sites/default/files/lessons/AlgorithmsComplexity%20-%20Sorting%20Socks-%20MS_0.pdf
5. https://csedweek.org/files/CSEDrobotics.pdf
6. http://www.comp.nus.edu.sg/~cs1010/4_misc/jumpstart/chap3.pdf
7. https://www.cs.jhu.edu/~jorgev/cs106/ProblemSolving.html
8. https://bletchleypark.org.uk/
9. https://www.rocketcenter.com/forms/edu/101Inventions/101Enigma.pdf
10. http://enigmaco.de/enigma/enigma.html
11. https://www.alice.org/
12. http://www.cs.bsu.edu/cs4ms/docs/CompoundConditionalsSlides.pptx
13. http://www.cs.bsu.edu/cs4ms/docs/GymActivities.pdf

6-8.DI.2 - Parallelization

Short Description

Describe the process of parallelization as it relates to problem solving.

Long Description

Students should understand the concepts behind parallelization, the use of two or more processors (e.g., cores, computers) in combination to solve a single problem. Normally a computer must complete tasks in order and one at a time. To increase the speed at which processes can be completed, parallel computing can be implemented. Parallel computing aims to break down processes into discrete parts that different cores of a processor can work on at the same time.

Explanation For Teachers:

Parallel computing is when we have a process that needs to do multiple tasks at the same time. We can use the different cores of a processor to work on different tasks at the same time. Normally, instructions would be carried out in sequence, one at a time. Parallelization is the process of dividing the process up into tasks that can be accomplished at the same time, so that the computer does not need to wait as long to move on.

Purpose

Students will benefit from an understanding of breaking down a problem into different parts and distributing the work evenly amongst a team. This will increase teamwork and understanding of similarities between how humans process tasks and how computer process tasks.

Objectives

- Understand the process of solving a problem through parallelization
- Use parallelization to solve a problem

Proposed Activities

- Hamburger Cooking Activity*********
- Hamburger Cooking Worksheet
- Sorting Networks (1)
- Sorting With Parallelism Activity (2)
- Multi Core Processors (3)
- How To TEACH “Algorithmic Thinking”? (4)
- Animal Classification (Additional Activities)

Resources

- "Definition: Parallel Programming" (5)

Hyperlinks

1. http://csunplugged.org/sorting-networks/
2. http://csunplugged.org/wp-content/uploads/2015/03/unplugged-08-sorting_networks-original.pdf
3. https://www.nsf.gov/cise/csbytes/newsletter/vol2/vol2i17.html
4. https://codinginmathclass.wordpress.com/2014/09/10/how-to-teach-algorithmic-thinking/
5. http://searchdatacenter.techtarget.com/definition/parallel-processing

6-8.DI.3 - Data

Short Description

Represent data in various ways (e.g., numbers, text, pictures) and use visual representations of problems and data.

Long Description

Students should be able to read and represent data in a variety of ways (e.g., text, sounds, pictures, and numbers), and use different visual representations of problems, structures, and data (e.g., graphs, charts, network diagrams, flowcharts).

Explanation for Teachers

We collect data from samples, organize this data using graphs and charts, and then utilize statistical reasoning to make conclusions about an entire population. Data is an important part of statistics.

Purpose

Students will benefit from an ability to express ideas in a multitude of *visual representations*. Students will also gain the ability to interpret various visual representations. Students can make and interpret graphs and flowcharts to integrate in math, science, and English courses. Students will also gain an appreciation for evaluation of data.

Objectives

- Convert decimal numbers into binary
- Make visual representations of data such as graphs and charts on computer (or on paper)

Proposed Activities

- Songwriting with Parameters (1)
- Functional Suncatchers (2)
- Tune Trace (teacher's handbook included) (3)
- How Computers Think (4)

Alternative Approaches

- Use a more hands on approach to teach (e.g., physical objects to represent data)
- Binary Bracelets (Additional Activities)
- Flow charts, graphs, and such could also fit well with a science curriculum or business class

Resources

- “Why Kids Need Data Literacy, and How You Can Teach It” (5)
- How Computers Think (4)
- Binary Powerpoint (6)

Hyperlinks

1. https://studio.code.org/s/course3/stage/9/puzzle/1
2. https://studio.code.org/s/course3/stage/4/puzzle/1
3. http://www.cs4fn.org/teachers/activities/tunetrace/
4. http://www.cs.bsu.edu/cs4ms/docs/HowComputersThinkLessonPlan.pdf
   http://www.cs.bsu.edu/cs4ms/docs/HowComputersThinkSlides.pptx
   http://www.cs.bsu.edu/cs4ms/docs/WaveActivity.pdf
   http://www.cs.bsu.edu/cs4ms/docs/PixelGridActivity.pdf
5. http://www.slj.com/2017/06/technology/why-kids-need-data-literacy-and-how-you-can-teach-it/#_
6. http://www.cs.bsu.edu/cs4ms/docs/BinarySlides.pptx

6-8.DI.4 - Abstraction and Hierarchy

Short Description
Understand the notion of hierarchy and abstraction in computing including high-level languages, translation, instruction set, and logic circuits.

Long Description
Students should understand that building things in hierarchies is very common in computer software. For example, file systems provided by operating systems have a top-level directory that goes to sub-directories like "Program Files" and "Documents and Settings" and under these can be even more sub-directories. Abstraction is a new representation of a thing, a system, or a problem that helpfully reframes a problem by hiding details irrelevant to the question at hand. (source [1])

Explanation for Teachers

Abstraction: Computers at a hardware level are only able to understand binary expressions and all data (e.g., numbers, characters) has to be represented using binary. The way in which a computer stores this data does not affect the user, therefore it is hidden from them. This is called abstraction. Computer scientists often use abstraction when writing a program; for example, a programmer can create a program that pulls a file off the internet and gives it to another programmer who can use that program as a portion of their own. The implementation of the program to pull the file from the internet is abstracted, meaning the second programmer does not need to know the technical portions of the first program in order to access files on the internet.

Hierarchy: Many modern programming languages allow the program to be built from objects, which are collections of pieces of data and functions to be able to interact with that data. An example class could be called **motor-vehicle**, which would contain all things common to motor vehicles such as an *engine* and functions to interact with the motor vehicle, such as *start()* and *drive()*. The issue is that **motor-vehicle** will not contain variables not common to all motor vehicles such as doors or the number of wheels; to fix this programming, languages allow for the creation of objects such as **car** or **motorcycle** which inherit all of the attributes of **motor-vehicle** but also include data specific to themselves. The relationship between these types of objects is hierarchy.

Hierarchy:
We will use a Math teacher as an example of Abstraction and Hierarchy. All Math teachers at a school are teachers, and all teachers are employees. The hierarchy would look like Employee → Teacher → Math Teacher. The school might have a requirement that says all employees need to have a bank account set up so that they can get their paychecks from the school. However, the requirements for a Math teacher might only include "Prepare students for the Math portion of the standardised tests". Even though all Math Teachers will need to have a bank account set up, we leave those requirements out from the Math teacher's requirements, and allow it to be handled by the Employee level of the hierarchy. And we do not put the Math teacher's requirements in the Teacher or Employee level, because not all Employees need to prepare students for the Math portion of the Standardised tests.

Abstraction:
If someone asked you to go see a movie with them, and you told them that you could not because you had to grade papers, you would be using a form of abstraction. When you said you had to grade papers, you left out all of the small details. Instead of saying that you needed to check every answer on the quizzes against the answer key, mark down the score for each quiz, and enter those scores into the gradebook. You simply abstracted all of those actions into a single action called "grading" because all of the details were not of importance to the "end user" or, your friend who asked you to go see a movie.

Purpose
Students are able to determine small steps in a large process. This could be applied to large projects or homework assignments as students will be able to recognize small progresses leading to a goal. Students will also be able to visualize the complexities of a large system via its parts.

Objectives

- Explain how abstraction is the different ways something can be perceived. For example, typing "2+2 = 4" in a computer, but the computer may process that as something differently (binary)
- Demonstrate an understanding of hierarchy by sorting out a file system correctly
- Demonstrate an understanding of how the computer processes information is a hierarchy by sorting out a graph correctly

Proposed Activities

- Abstraction of People (Additional Activities)
- Computational Differences (Additional Activities)
- How Computers Think (4)

Alternative Approaches

- Binary bracelets (Same concept. The same thing is perceived differently by the machine and person)
- Activities on hierarchy can be integrated into science classes, as hierarchy plays a role in this topic

Resources

- Introduction to Computer Programming: Hierarchy (2)
- The Computer Hierarchy (3)
- How Computers Think (5)
- Abstraction and Hierarchy (4)

Hyperlinks

1. https://www.doe.in.gov/sites/default/files/standards/6-8csresourceguide.pdf
2. http://guyhaas.com/bfoit/itp/Hierarchy.html
3. http://alanclements.org/1computerhierarchy.html
4. http://www.cs.bsu.edu/cs4ms/docs/AbstractionHierarchySlides.pptx
5. http://www.cs.bsu.edu/cs4ms/docs/HowComputersThinkLessonPlan.pdf
   http://www.cs.bsu.edu/cs4ms/docs/HowComputersThinkSlides.pptx
   http://www.cs.bsu.edu/cs4ms/docs/WaveActivity.pdf
   http://www.cs.bsu.edu/cs4ms/docs/PixelGridActivity.pdf

6-8.DI.5 - Computational Thinking

Short Description
Demonstrate interdisciplinary applications of computational thinking and interact with content-specific models and simulations to support learning and research.

Long Description
Students should be able to apply the principles of computational thinking to interdisciplinary subjects. They should be able to interact with content-specific models and simulations to support learning and research of any subject.

Explanation for Teachers
Computational thinking is a set of problem-solving methods that involve expressing problems and their solutions in ways that a computer could execute. Even something as simple as reflecting back on an experience multiple times can be considered computational thinking because it is both retrospective and recursive.

Purpose
Students will gain the knowledge of how computers are involved in almost every discipline. This will be helpful to students in using computers in other courses to create reports or projects. Computational thinking can make other areas of STEM more approachable.

Objectives

- How to include computational thinking into problem solving for other disciplines
- Showing how computational thinking is already in ways students learn
- Presenting computational thinking in scenarios not related to computer science
- The thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent, for example: a computer

Proposed Activities

- Classify animals (1)
- Developing a tree to show the flow of information in a game of 20 questions (2)
- Using Ozobot to teach children about different states of matter by using different colors to make a robot move differently (3)
- Lessons on pattern matching and decomposing (4)
- Teaching techniques in understanding decomposition (5)
- Knight's tour: using algorithmic and computational thinking to move a knight around a grid. (6)
- Educational Video Games (10)
- Gym Activities (11)

Alternative Approaches

- Teaching trial and error techniques.
- Explaining that an algorithm is just steps to complete a task.

Resources

- "How one school district works computational thinking into every grade and class" (7)
- "Teaching Computational Thinking Is the First Step to Bridging STEM Skills Gap" (8)
- Free teaching resources for computational thinking. (9)

Hyperlinks

1. http://csunplugged.org/wp-content/uploads/2014/12/PhylogeneticsUnplugged.pdf
2. http://csunplugged.org/wp-content/uploads/2014/12/unplugged-05-information_theory.pdf
3. https://www.iste.org/explore/articleDetail?articleid=894&category=In-the-classroom&article=
4. https://code.org/curriculum/course3/1/Teacher
5. https://educators.brainpop.com/lesson-plan/computational-thinking-lesson-plan-decompose/
6. https://teachinglondoncomputing.org/resources/inspiring-computing-stories/computational-thinking-knightstour/
7. http://hechingerreport.org/how-one-school-district-works-computational-thinking-into-every-grade-and-class/
8. https://www.google.com/amp/s/edtechmagazine.com/k12/article/2016/11/teaching-computational-thinking-first-step-bridging-stem-skills-gap%3famp
9. https://www.iste.org/explore/articleDetail?articleid=152&category=Solutions&article=Computational-thinking-for-all
10. http://www.cs.bsu.edu/cs4ms/docs/EduVideoGames.pdf
11. http://www.cs.bsu.edu/cs4ms/docs/GymActivities.pdf

6-8.CD.1 - Hardware and Software

Short Description

Demonstrate an understanding of the relationship between hardware and software.

Long Description

Students should understand the difference between software and hardware as well as how they interact:

- Hardware – the physical components that make up a computing system, computer, or computing device
- Software – programs that run on a computer system, computer, or other computing device

Explanation for Teachers

When most people think of a computer, they think of computer hardware. The mouse, the monitor, the keyboard - these are all pieces of hardware. Software, on the other hand, cannot be seen. Software is the code that the computer runs on. Machine code, assembly code, and written code are examples of software.

Purpose

Students will be able to discern between a computer's hardware and software. This will allow them to further their understanding of computers and isolate issues that may arise with everyday computer usage.

Objectives

- Describe necessary components to a computer system
    - CPU, RAM, HD, Motherboard
- Describe extra components (peripherals) that can be added to the computer
    - Keyboard, Mouse, Speakers
- Describe how software is stored on a computer
- Know the differences between operating systems and applications
- Demonstrate how hardware does the work for the software
- Demonstrate how software tells the hardware what to do
- Describe the interactions between software and hardware

Proposed Activities

- Computer Component Sorting (Additional Activities)
- Program Data Path (Additional Activities)
- How Computers Think (http://www.cs.bsu.edu/cs4ms/docs/HowComputersThinkLessonPlan.pdf http://www.cs.bsu.edu/cs4ms/docs/HowComputersThinkSlides.pptx http://www.cs.bsu.edu/cs4ms/docs/WaveActivity.pdf http://www.cs.bsu.edu/cs4ms/docs/PixelGridActivity.pdf)

Alternative Approaches

- Teach students how another "system" works without talking about hardware
  - CPU is the brain or controller
  - RAM is the ideas that the brain is thinking about
  - HD is all the memories the system has
  - Motherboard is the body that let's all the parts connect
  - Keyboard is the senses that we have
  - Software is an action we can perform

Resources

- Introduction to Computers (1)
- Hardware vs Software (2)
- What is Software? (3)
- How Computers Think (4)

Hyperlinks

1. http://cs.sru.edu/~mullins/cpsc100book/module02_introduction/module02-03_introduction.html
2. http://www.diffen.com/difference/Hardware_vs_Software
3. http://www.bbc.co.uk/education/guides/zcxgr82/revision
4. http://www.cs.bsu.edu/cs4ms/docs/HowComputersThinkLessonPlan.pdf
   http://www.cs.bsu.edu/cs4ms/docs/HowComputersThinkSlides.pptx
   http://www.cs.bsu.edu/cs4ms/docs/WaveActivity.pdf
   http://www.cs.bsu.edu/cs4ms/docs/PixelGridActivity.pdf

6-8.CD.2 - Troubleshooting

Short Description
Apply troubleshooting strategies to identify and solve routine hardware and software problems that occur during everyday computer use.

Long Description
Students should understand troubleshooting, a systematic approach to problem solving that is often used to find and resolve a problem, error, or fault within software or a computer system. Students should be able to apply basic troubleshooting skills for common problems that they may encounter.

Explanation for Teachers
Troubleshooting is a logical thinking process/method that helps discover the source of an error.

Purpose
Students will be able to learn troubleshooting techniques that they can use in computers and their everyday lives. This will enable students to understand how a problem arises and not just the symptoms of this problem. Troubleshooting will also help with critical thinking as students must understand a deeper level of knowledge to an issue.

Objectives

- Apply basic troubleshooting strategies such as:
    - Always check cables
    - Restart the device
    - Watch for error messages
        - Write down error messages
        - Do research to find out what online resources say about the issue
    - Use process of elimination to rule out problems
    - Start from the beginning and work on checking the entire system
- Understand key shortcuts to important computer processes
    - Ctrl + Alt + Delete opens the Task Manager
    - Alt + f4 closes the current window
- Find trustworthy troubleshooting solutions with Internet searches
- Run anti-virus & anti-malware software

Proposed Activities

- Troubleshooting Solutions (Additional Activities)
- Hands On Troubleshooting (Additional Activities)

Alternative Approaches

- Replicated Troubleshooting (Additional Activities)

Resources

- Basic Troubleshooting Techniques (1)
- Understanding Troubleshooting (2)
- Troubleshooting Powerpoint (4)

Hyperlinks

1. https://www.gcflearnfree.org/computerbasics/basic-troubleshooting-techniques/1/
2. https://technet.microsoft.com/en-us/library/bb457121.aspx
3. https://technet.microsoft.com/en-us/library/bb457121.f27zs01_big(l=en-us).jpg
4. http://www.cs.bsu.edu/cs4ms/docs/Troubleshooting.pdf

6-8.CD.3 - Networks

Short Description
Describe the major components and functions of computer systems and network.

Long Description
Students should be able to to understand how a network passes information through routers, and how information is passed from one node to another inside a network. They should have a basic framework of how a server facilitates a connection. Students should also understand the difference between data and information. They should understand how a computer chooses the best specific path(s) along a network.

Explanation for Teachers
A computer network is like a tree or web that connects multiple computers with each other. There are both wired and wireless connections.

Purpose
Students are able to understand how information is communicated through a network, applicable to all networks. They will be able to use this information to build their own networks, as well as use already existing networks to their greatest capability. They will also understand the problems that may occur in already existing communication networks.

Objectives

- Understand the relationship between the 'server' (or 'host') and the 'client' computer, and the verification of information that happens between them for a communication channel to be established
- Understand how networks communicate through packet switching
- Understand how servers distribute information to an end point
- Understand the difference between an 'internet' and an 'intranet'
- Understand routing and deadlock
- Understand minimal spanning trees
- Understand that the computer sorts the information into different categories depending on their relevance to the information
- A computer sorts data by using these categories in relation to each other, creating an intersecting grid that allows users to specify what category the information falls into. For example, a user can tell a computer to find all cities that start with a vowel, but not the vowel 'A'
- Understand the difference between data and information

    - Raw data is merely the actual knowledge: "It will rain on the 25th," for instance
    - Information requires context: "In the city of Muncie, in the month of May, on the 25th day, it will rain"
- Be able to understand the concept of a 'network'

Proposed Activities

- Computing Unplugged (1)
- Minimal Spanning Trees (2)
- Routing and Deadlock (3)
- Internet (4)
- Human Network (Additional Activities)

Alternative Approaches

- Physically represent packet-passing by passing a paper with a message on it between students and naming what part of the process each person represents
- Play Telephone and talk about how the "data" changed and became corrupted

Resources

- Basic Computer Network Components (5)
- Internet Safety (6)

Hyperlinks

1. http://csunplugged.org/databases/
2. http://csunplugged.org/minimal-spanning-trees/
3. http://csunplugged.org/routing-and-deadlock/
4. https://studio.code.org/s/course3/stage/18/puzzle/1
5. https://en.wikiversity.org/wiki/Basic_computer_network_components
6. http://www.cs.bsu.edu/cs4ms/docs/InternetSafetyLessonPlan.pdf
   http://www.cs.bsu.edu/cs4ms/docs/InternetSafetySlides.pptx

6-8.CD.4 - Differences between Human and Machine

Short Description
Describe what distinguishes humans from machines focusing on human intelligence versus machine intelligence and ways we can communicate. As well as ways in which computers use models of intelligent behavior (e.g., robot motion, speech and language understanding, and computer vision).

Long Description
Students should understand that computers differ from people in several meaningful, important, and relatively easy-to-distinguish ways. They should have a basic understanding of the main difference between a neural network, and know that computers don't normally use these processes to handle information. Students should have a basic understanding of how computers learn, and they should understand that a computer cannot feel emotion: a computer can only simulate emotion that human programmers code into it.

Explanation for Teachers
Artificial intelligence is built by humans to mimic human behavior not the structure of the brain, which is analog not digital like a computer. Artificial intelligence is built upon pattern recognition over lots and lots of data. Artificial intelligence is much less pliable than human intelligence, being designed to learn just one thing. Artificial intelligence also lacks the bias attached to human decision making.

Purpose
Students will understand the fundamental differences between humans and machines, as well as the ways that machines can manipulate their emotions. They should understand that emotions are the most valuable asset of humans, and that they are fundamentally unquantifiable as far as we know.

Objectives

- Understand the general idea of the Turing Test and how/why human communication is complex compared to the generated messages of a computer
- Understand the fundamental difference between binary and richly conveyed information, as well as human 'randomness' compared to computer randomness
- Simulate the basic functionalities of a neural network, and how computers use, store, and access information
- Be able to differentiate between legitimate human emotion and simulated emotion

- Understand that the computers of today cannot feel emotion, and that emotion is a biological function that cannot be replicated through digital methods
- Understand that computer 'sense' is similar, but not the same as human senses
- Describe the basic concept of learning through failure, and how computers use failure as a learning method

Proposed Activities

- The Turing Test (1)
- Brain-in-a-Bag (2)
- Emotional Robot Video (3)
- Sweet Learning Computer (4)
- Objects and Emotions (Additional Activities)

Resources

"10 Important Differences Between Brains and Computers" (5)
"Machines Can't Flow: The Difference Between Mechanical and Human Productivity" (6)
"Difference Between Human and Machine" (7)

Hyperlinks

1. http://csunplugged.org/the-turing-test/
2. http://www.cs4fn.org/teachers/activities/braininabag/
3. http://www.cs4fn.org/teachers/activities/emotionalrobot/
4. http://www.cs4fn.org/teachers/activities/sweetcomputer/
5. http://scienceblogs.com/developingintelligence/2007/03/27/why-the-brain-is-not-like-a-co/
6. https://www.theatlantic.com/technology/archive/2013/06/machines-cant-flow-the-difference-between-mechanical-and-human-productivity/276969/
7. http://www.differencebetween.net/miscellaneous/difference-between-human-and-machine/

6-8.PA.1 - Tools & Resources

Short Description

Select appropriate tools and technology resources to support learning and personal productivity, publish individual products, design, develop, and publish data, accomplish a variety of tasks, and solve problems.

Long Description

Students should learn how to select applications for productivity, such as spreadsheet, powerpoint, word processing, etc., and how to use them efficiently. Students should also learn how to design data visualizations ethically and with effective communication.

Explanation for Teachers

Students should be able to pick the right method of processing to do an activity.

Purpose

Students will encounter data in science and math courses that may need to be manipulated. This standard will help students understand computer applications for presentations that could be used in all courses, including presentations for English courses. Tools presented throughout this standard will help students gain valuable communication skills that will assist with delivering information concisely and in a visually appealing manner.

Objectives

- Primary programs for data manipulation / graphing
- Good practices for data graphing, such as when to use certain forms, ethical color schemes, making charts as legible as possible, labelling axes, etc.
- Importance of visually presenting data over a table of numerical data

Proposed Activities

- Human Bar Graph (Additional Activities)
- Representing Data (Additional Activities)
- M&M Graph (1)

Alternative Approaches

- Students with dyscalculia/acalculia (difficulty comprehending numbers, mathematics, arithmetic) may benefit from more visual representations of data
    - Extend visual representation to physical objects - blocks, weights, etc.
- Students can possibly learn programmatic/algorithmic approaches for data input, etc. rather than manipulating input data directly - bysteps potential attention disorder issues by providing an engaging, puzzle-solving task rather than rote data entry
- Use color-blind friendly colors for presenting graphical data. Commonly, orange and blue combinations are easier to visualize than red and green combinations.

Resources

- Anscombe's Quartet (2)
- Zombie-Town USA (3)

Hyperlinks

1. http://www.brighthubeducation.com/middle-school-math-lessons/3352-mm-graphing-lesson/
2. https://blog.heapanalytics.com/anscombes-quartet-and-why-summary-statistics-dont-tell-the-whole-story/
3. https://mattbierbaum.github.io/zombies-usa/

6-8.PA.2 - Problem Solving with Programming

Short Description
Implement problem solutions using a programming language that includes looping behavior, conditional statements, logic, expressions, variables, and functions.

Long Description
Students should be able to solve problems with programming and computational thinking. They should be able to use looping behavior, conditional statements, logic, expressions, variables, and functions and recognize when and where they are necessary.

Explanation for Teachers
Students should be able to use computational thinking to solve problems; i.e. logic puzzles.

Purpose
Basic programming is able to teach students valuable problem solving skills that can help them in every course. Alongside this, students will be required to understand programming syntax which can help in understanding English and mathematical symbols. Lastly, students will be able to implement fail-until-succeed techniques.

Objectives
Students will learn:

- Logical/step-by-step problem solving
- Basics of programming including:
  - Writing code
  - Functions, expressions, and variables
  - Loops and Conditionals

Proposed Activities

- Variables with Khan Academy (1)
- Functions with Khan Academy (2)
- Introduction to Computer Programming: Operators & Expressions (3)
- Loops with Khan Academy (4)
- Flocabulary: Coding Conditionals (5)
- RoboZZle (6)
- Network Protocols (7)
- Finite State Automata (8)
- Programming Languages (9)
- BrainPOP Meme Project (13)

Alternative Approaches

- The following resources can be applied without access to computers:
    - 10 Games That Promote Problem-Solving Skills (10)
    - Ways of Teaching Code (11)
- Individualized Instructions
- Small Group Work

Resources

- Simple Programmer: Solving Problems, Breaking it Down (12)
- Compound Conditions Powerpoint (14)

Hyperlinks

1. https://www.khanacademy.org/computing/computer-programming/programming/variables/p/intro-to-variables
2. https://www.khanacademy.org/computing/computer-programming/programming/functions/p/functions
3. http://guyhaas.com/bfoit/itp/Operators.html
4. https://www.khanacademy.org/computing/computer-programming/programming/looping/p/intro-to-while-loops
5. https://www.flocabulary.com/unit/coding-conditionals/
6. http://www.robozzle.com/js/index.aspx
7. http://csunplugged.org/network-protocols/
8. http://csunplugged.org/finite-state-automata/
9. http://csunplugged.org/programming-languages/
10. https://www.stenhouse.com/sites/default/files/public/legacy/pdfs/8247ch10.pdf
11. https://www.edutopia.org/blog/15-ways-teaching-students-coding-vicki-davis
12. https://simpleprogrammer.com/2011/01/08/solving-problems-breaking-it-down/
13. http://www.cs.bsu.edu/cs4ms/docs/BrainPopLesson.pdf
14. http://www.cs.bsu.edu/cs4ms/docs/CompoundConditionalsSlides.pptx

6-8.PA.3 - Flexibility in Problem Solving

Short Description
Demonstrate dispositions amenable to open-ended problem solving and programming.

Long Description
Students should be comfortable with solving open-ended problems (e.g., comfort with complexity, persistence, brainstorming, adaptability, patience, propensity to tinker, creativity, accepting challenge). This should include both in general and with programming.

Explanation for Teachers
Students can solve open-ended problems.

Purpose
Students will often face problems on homework assignments or in life that do not have a clear path to follow. This standard will build problem solving skills to help students resolve issues that may not have obvious solutions. Some of the skills students will learn include flexibility in addressing an issue by looking at the problem from various angles, adaptability to change the working solution as better solutions become known, and persistence in pursuing solutions through potential failure.

Objectives

- Open-ended problem solving skills such as:
  - Flexibility
  - Adaptability
  - Persistence

Proposed Activities

- 7 Tips For Building Flexible Thinking (1)
- Factory Balls (2)
- Invert Selection (3)
- Sugar, Sugar (4)
- Big Wash (5)

Alternative Approaches

- Techniques for Teaching Flexibility (6)

Resources

- "6 Ways Kids Use Flexible Thinking to Learn" (7)
- "Teaching Kids to Debug Code Independently" (8)

Hyperlinks

1. https://www.understood.org/en/school-learning/learning-at-home/homework-study-skills/7-tips-for-building-flexible-thinking
2. https://www.coolmath-games.com/0-factory-balls
3. https://www.coolmath-games.com/0-invert-selection
4. https://www.coolmath-games.com/0-sugar-sugar
5. https://www.coolmath-games.com/0-big-wash
6. http://archive.brookespublishing.com/articles/ed-article-0911.htm
7. https://www.understood.org/en/learning-attention-issues/child-learning-disabilities/executive-functioning-issues/6-ways-kids-use-flexible-thinking-to-learn
8. https://www.edsurge.com/news/2017-01-19-teaching-kids-to-debug-code-independently

6-8.NC.1 - Collaborative Design and Publication

Short Description
Collaboratively design, develop, publish, and present products (e.g., videos, podcasts, websites) using technology resources that demonstrate and communicate curriculum concepts.

Long Description
Students should be able to work collaboratively, with their peers, to design and develop products, demonstrating knowledge from the rest of the curriculum. They should also be able to choose the right publication method for their product and be able to present their product to an appropriate audience.

Explanation for teachers
Students use production and design tools such as Microsoft/Google Suite.

Purpose
Students will gain the ability to use collaborative techniques along with knowledge of technological tools that will aid them in collaboration. This can be applied to any subject as they might not have been exposed to tools they could use to solve problems.

Objectives
Students will learn:

- the importance of collaboration between peers
- the process of designing and development of products
- about multimedia publication and presentation methods

Proposed Activities

- Collaboration and Gender Equality Game (1)
- Learning to Teach Programming (2)
- BrainPOP Meme Project (5)

Alternative Approaches

- Encourage paired programming and collaboration among peers when seeking help, prior to assistance from the instructor, in order to promote collaborative strategies.
- Alternate Teaching Approaches for Computer Science (3)

Resources

- Collaboration (4) - Defining collaboration and types of collaboration

Hyperlinks

1. https://drive.google.com/file/d/0B-vhvbZ_yjScM0d2ZEN2Qm9tQWs/view
2. https://medium.com/@ed_saber/to-learn-to-teach-to-build-for-the-web-d9a3a280c2f7
3. https://ctrl.education.illinois.edu/TACTICal/Collaboration
4. http://www.aiim.org/What-is-Collaboration
5. http://www.cs.bsu.edu/cs4ms/docs/BrainPopLesson.pdf

6-8.NC.2 - Feedback Integration

Short Description

Exhibit dispositions necessary for collaboration: providing useful feedback, integrating feedback, understanding and accepting multiple perspectives, socialization.

Long Description

Students should be able to provide useful feedback towards their peers. They should know the different methods of feedback integration. They should be able to understand the importance of other peers' perspectives, and to be able to respect those perspectives.

Explanation for teachers

Any large scale task involving computers is going to be collaborative. Therefore, it is imperative that students are able to communicate with each other well.

Purpose

Students will benefit from this standard by gaining an understanding of respectful collaboration. This can be applied to English as a peer editing skill and to group work as communication techniques. Students will also learn how to gracefully accept others' opinions to improve their work.

Objectives

- The importance of providing feedback
- Types of productive feedback
- The importance of seeing things from other perspectives
- The importance of socialization and working together

Proposed Activities

- 10 Collaborative Technology Projects Your Students Will Love! (1)
- Focused Feedback (Additional Activities)

Alternative Approaches

- Google Drive Approach (2) - Using google applications, such as google docs, for collaboration and providing feedback as a way to help encourage students with special needs to work with other students, as well as their instructors.

Resources

- “Pair programming in the classroom” (3)
- “Pair Programming Strategies for Middle School Girls” (4)
- “3 best practices for pair programming” (5)
- “Pair Programming: 10 Cool Tips to Make It Work in Your Classrooms” (6)

Hyperlinks

1. https://www.weareteachers.com/10-collaborative-technology-projects-your-students-will-love/
2. http://www.techlearning.com/default.aspx?tabid=100&entryid=471
3. https://www.khanacademy.org/resources/out-of-school-time-programs/teaching-computing/a/pair-programming-in-the-classroom
4. https://pdfs.semanticscholar.org/c143/ef2b9060bcd125f959bbb34223a29d970bef.pdf
5. https://www.iste.org/explore/articleDetail?articleid=221
6. https://www.etr.org/blog/research-pair-programming/

6-8.IC.1 - Legal and Ethical Behaviors

Short Description

Exhibit legal and ethical behaviors when using technology and information and discuss the consequences of misuse.

Long Description

Students should be able to understand the impact they have through interactions online, especially in regards to cyberbullying. Students should also understand copyright policies, the concept of fair use, and the legal and ethical ramifications of piracy and **plagiarism**. Students should also learn methods to identify and avoid potential scams.

Explanation For Teachers

Demonstrate the legal consequences of misusing the internet.

Purpose

Students will benefit from using technology legally and ethically. This could keep students from performing actions that may get them in trouble, but also outline the benefits of using technology in a healthy way. It can also help shy students away from cyberbullying their peers in the future if they know the impact of such behavior.

Objectives

- Define and recognize examples of cyberbullying
- Discuss examples and consequences of piracy
- Discuss examples and consequences of plagiarism
    - Understand fair use and copyright policies
- Explore ways to avoid Internet scams

Proposed Activities

- Cyberbullying: Be Upstanding (1)
- Scams and Schemes (2)
- Cyberbullying: Crossing the Line (3)
- PBS: Copyright & Fair Use (4)
- B4U Young Minds Inspired - Internet Use and Copyright (5)
- Utah Education Network - Computer Ethics (6)
- Legitimacy of Mail (Additional Activities)
- Fair Use of Media Files (Additional Activities)
- Intended Message Tone (Additional Activities)

Resources

- The Ten Commandments of Computer Ethics (7)
- Internet Safety (8)

Hyperlinks

1. https://www.commonsense.org/education/lesson/cyberbullying-be-upstanding-6-8
2. https://www.commonsensemedia.org/educators/lesson/scams-and-schemes-6-8
3. https://www.commonsensemedia.org/educators/lesson/cyberbullying-crossing-line-6-8
4. https://studentreportinglabs.org/resource/lesson-14-structure-broadcast-news/
5. http://ymiclassroom.com/wp-content/uploads/2012/05/B4UCopy_middleschool.pdf
6. http://www.uen.org/Lessonplan/preview?LPid=486
7. http://cpsr.org/issues/ethics/cei/
8. http://www.cs.bsu.edu/cs4ms/docs/InternetSafetyLessonPlan.pdf
   http://www.cs.bsu.edu/cs4ms/docs/InternetSafetySlides.pptx

6-8.IC.2 - Impact of Technology

Short Description

Analyze the positive and negative impacts of technology on one's personal life, society, and our culture.

Long Description

Students should be able to describe how technology has changed the way we view and interact with the world in both positive and negative ways. They should also have an understanding of how technology and the computer have evolved over time and how that has affected society. They should be able to identify positive and negative consequences of the involvement of technology in their daily lives.

Explanation For Teachers

Show the benefits and determinants that technology has brought to our lives and others.

Purpose

Students will gain an appreciation for technology in their lives. Students will learn how different forms of technology assist them in everyday tasks that can transition their other courses by providing them with tools that eliminate difficulties. Students will also benefit by learning problems with technological involvement in our lives, including dependencies, and how to avoid these in order to live healthy, balanced lives.

Objectives

- Map the evolution of technology, the computer, and the internet on a timeline
- Determine what aspects of modern life rely on technology
- Classify impacts of technology as positive or negative
- Outline the consequences for internet activity

Proposed Activities

- Digital Life 101 (1)
- Trillion Dollar Footprint (2)
- Technology Timeline (Additional Activities)
- Dark Classroom (Additional Activities)
- Future Technology (Additional Activities)

Alternative Approaches

- Have a discussion with the class about how technology positively and/or negatively impacts their lives.
- Use technology as a teaching tool and discuss how it impacts the learning environment in the classroom.

Resources

- Timeline of Technology Video (3)
- "Technology and Society" (4)
- "Environmental and Societal Impact of Technology" (5)

Hyperlinks

1. https://www.commonsense.org/education/lesson/digital-life-101-6-8
2. https://www.commonsensemedia.org/educators/lesson/trillion-dollar-footprint-6-8
3. https://www.youtube.com/watch?v=be3KZDFvDeA
4. https://www.useoftechnology.com/technology-society-impact-technology-society/
5. http://www.digitalresponsibility.org/environmental-and-societal-impact-of-technology/
6. https://sites.google.com/site/undergroundrailroadgame/home-1

6-8.IC.3 - Accuracy and Bias

Short Description
Evaluate the accuracy, relevance, appropriateness, comprehensiveness, and biases that occur in electronic information sources.

Long Description
Students should learn how to correctly identify and cite reputable sources for information. They should also be able to evaluate the credibility of an information source. Students should know the difference between bias and unbiased information through electronic sources.

Explanation For Teachers
Demonstrate that information on the internet can be bias and misinformed. Demonstrate to the students how to tell the difference and point out the bias or misinformation.

Purpose
This standard will be used to teach students how to think for themselves by learning information from credible sources. By teaching students how to glean information from credible sources, the standard improves their ability to write better research reports, as well as improving their personal accountability when generating values and ideals.

Objectives

- Identify reputable and non reputable sources
- Create citations from information in an article
- Define fake news and recognize examples
- Comparing articles on the same topic from opposite outlooks

Proposed Activities
From the Standards:

- Strategic Searching (1)
- Identifying High Quality Sites (2)
- The Credibility Challenge (3)
- Reading and Writing Citations (4)
- Teaching Your Students About Fake News (5)
- Fighting Fake News (6)

Alternative Approaches

- Magazine Credibility (Additional Activities)
- News Biases (Additional Activities)

Resources

- EasyBib Evaluating Sources (7)
- Media Bias Lesson Plan (8)

Hyperlinks

1. https://www.commonsensemedia.org/educators/lesson/strategic-searching-6-8
2. https://www.commonsensemedia.org/educators/lesson/identifying-high-quality-sites-6-8
3. http://www.annenbergclassroom.org/page/the-credibility-challenge
4. https://style.mla.org/2016/11/17/reading-and-writing-citations/
5. http://www.pbs.org/newshour/extra/lessons_plans/lesson-plan-how-to-teach-your-students-about-fake-news/
6. http://ww2.kqed.org/lowdown/wp-content/uploads/sites/26/2016/12/Fake-news-lesson-plan.pdf
7. http://www.easybib.com/guides/students/writing-guide/ii-research/c-evaluating-sources-for-credibility/
8. http://www.pbs.org/newshour/extra/lessons-plans/decoding-media-bias-lesson-plan/

6-8.IC.4 - Access to Technology

Short Description
Describe ethical issues that relate to computers and networks (e.g., security, privacy, ownership, and information sharing), and discuss how unequal distribution of technological resources in a global economy raises issues of equity, access, and power.

Long Description
Students should learn how ownership differs between physical and digital media, understand copyright laws and how it affects digital online media, evaluate how different areas have different access to technology and technological access, and how online censorship can impact social movements. Student should be able to know how access to more technology helps raise the equity and power of the area.

Explanation For Teachers
Demonstrate how technology has affected areas' power and equity and the ethical issues within.

Purpose
Students should learn what their rights are as a digital citizen, as well as the rights of others, to better understand how their activity with computers can affect them and those around them. This will help students understand the risks involved in everyday computer use, understand how internet control affects day to day life, and how their favorite sites are selling their personal information to hundreds of companies, and how they have agreed to it.

Objectives
Students will learn:

- Understand net neutrality and internet hosting
- Define categories of ownership in regards to physical and digital media
- Recognize the geographical impact on technological resources and access
- Understand the impact of censorship online and how it impacts to social movements

Proposed Activities

- A Creator's Rights (1)
- Rework, Reuse, Remix (2)
- Information Hiding (3)
- Public Key Encryption (4)
- Scout Patrol (5)
- The Net Neutrality Debate (6)

Alternative Approaches

- Discuss and brainstorm methods to bring technological access to all people, regardless of what challenges they may face. Include physical limitations and economics in the discussion to recognize the changes in access among members of the same population.

Resources

- Map of the Internet (7)
- Pew Research Group Internet Trends (8)

Hyperlinks

1. https://www.commonsense.org/education/lesson/a-creators-rights-6-8
2. https://www.commonsensemedia.org/educators/lesson/rework-reuse-remix-6-8
3. http://csunplugged.org/information-hiding/
4. http://csunplugged.org/public-key-encryption/
5. http://csunplugged.org/scout-patrol-encryption/
6. http://www.pbs.org/moyers/moyersonamerica/netatrisk_lesson.html
7. https://internet-map.net/
8. http://www.pewglobal.org/2016/02/22/internet-access-growing-worldwide-but-remains-higher-in-advanced-economies/

Additional Activities

Abstraction of People

6-8.DI.4

Explore the aspects that all people have. For example, think about how people can be viewed as their mind, their body, or their atomic (biological) makeup. Let students offer their own examples of how humans are abstractions to see if the students understand the concept.

Algorithmic Task Completion

6-8.DI.5

Have students, in groups of 4, decide on a Rube Goldberg machine each to complete an arbitrary task. Instead of having a student design a full machine, students will draw a piece of the machine. Then, the students will rotate their paper right in their group. The next student will draw another piece to the machine. When the student gets their paper back, it should be a completed Rube Goldberg machine that completes their task. You can have a kind of bank of suggested ideas of pieces of the machine, such as a marble run or a lego robot kicking a ball.

Animal Classification

6-8.DI.2

6-8.DI.3

Use a flowchart to classify an animal into a class and species. Assign each student an animal and have them find the class and species based on the sorting network. This activity can be incorporated into science classes as well.

Binary Bracelets

6-8.DI.3

Have students create bracelets using the first letters of their first names. The following worksheet from code.org explains what beads correspond to each letter:
https://code.org/curriculum/course2/14/Activity14-BinaryBracelets.pdf
This project can also be accomplished using string and black and white beads.

Board Game Instructions

6-8.DI.1

Divide the class into small groups or pairs. Assign each group a card or board game, and have the students write down step-by-step instructions for the game using their memories. As a class, use the board game pieces or cards to play the game following the students' specific instructions. Determine whether the instructions for the game were written down clearly and what steps were included in the rules for playing each game.

Circuit Board Math

6-8.DI.5

Have students build an addition circuit to demonstrate to students how a computer does math at it's core. See the following online resource: *http://www.snapcircuits.net/*

Computational Differences

6-8.DI.4

Demonstrate how a computer program may convert a decimal number to a binary number. Then demonstrate the steps of the conversion on the chalkboard or dry erase board. Compare the easiness and the directions that are involved with both procedures.

Computer Component Sorting

6-8.CD.1

Take small paper squares and write computer components on each square. Distribute the squares to the students and have them sort the squares as to whether the component belongs to the software or hardware category.

Dark Classroom

6.8.IC.2

Experience a classroom without technology for a brief period of time. Have all students disregard electronic devices, turn off lights and computers, and hold a lecture without technology. See what changes and whether or not there are advantages or disadvantages to learning in an environment with technology. Do students seem more engaged? Are questions harder to answer? Is it more challenging to take notes or stay focused?

Fair Use of Media Files

6-8.IC.1

Identify if the use of a video or audio file is classified as fair use. Find a video on YouTube that may contain copyrighted material. Have the students identify the author of the video and determine whether the material is fair use or not.

Focused Feedback

6-8.NC.2

At the end of an activity, have students provide feedback and instructions to their peers. Encourage constructive criticism on topics in the class, such as algorithm design. One approach is to have students write feedback for students' projects on sticky notes and to have the responses hung on the board beside the project. Have students discuss what their peers did well and where they may need to seek improvement. Brainstorm ways as a class that the students can improve their work.

Future Technology

6.8.IC.2

Brainstorm and predict the future of technology. Have students draw pictures of what they think is going to happen next in technology, and devise a path for getting to this step. What is this new technology going to accomplish? Who will use it? Are there known consequences of this tool if it becomes integrated in the future society?

Hands On Troubleshooting

6-8.CD.2

Attempt to recreate everyday computer issues and have the students resolve the problems. Disconnect computer cables, if applicable, and have the students put the computer back together. Have students check for software updates and determine if any applications are outdated.

Human Bar Graph

6.8.PA.1

Have students line up along a whiteboard or blackboard in order from shortest to tallest. Mark each of their heights on the board. Once all of the students' heights has been marked, have the students move back to see the bar graph they created as a class.

Human Network

6-8.CD.3

Build a human network by having students stand in a line. Transport a verbal message from the first student through the last by having them whisper the phrase to each other. At the end of the line, determine whether the message changed. Repeat this process using a written message on a piece of paper. Have students pass the paper down the line. Which process was the most efficient? What must be done to ensure the information retains its original meaning? Are there any ways to guarantee that the information hasn't changed by the other end of the human network?

Intended Message Tone

6-8.IC.1

Determine the intended message tone from a written format and predict how the message will be received. Read a message in different emotions (angry, happy, hopeful, sad), and notice the changes that the reader may sense.

Legitimacy of Mail

6-8.IC.1

Collect examples of junk or spam mail in various mediums (email and physical mail). Have the students identify the sender, subject, and intentions of the mail. Have the students determine the trustworthiness of the mail and whether actions need to be taken (replying or blocking email).

Lego Structures

6-8.DI.1

Divide the class into pairs and give each student the same collection of Lego blocks. Have the students sit back-to-back so that they are not able to see what the other is doing. Have one student build a Lego structure. Then have the student give directions to the second student to build an identical Lego structure. After a few minutes, have the students compare their structures. Compare the results based on the order and clarity of directions. An additional approach to this assignment would be to use drawings or folded paper instead of Lego blocks.

Magazine Credibility

6-8.IC.3

Distribute magazines and tabloids around tables in the classroom. Determine what stories contain trustworthy information and which ones are not credible. To relate this activity to English courses, have the students create citations from the magazine articles they are using and ask how the information they are reading is useful. Should the information from the magazine be used in a research essay? How believable are the articles? Who is the target audience for this literature?

News Biases

6-8.IC.3

View news recordings from different news networks in order to identify political biases. Ask students what personal beliefs may influence the author's interpretation of the current event.

Objects and Emotions
6-8.CD.4
Have an inanimate object in the center of the room. Think about how the object may be personified and what feelings the object may appear to have. For example, consider a stuffed animal toy or a small robotic toy. Are the emotions genuine? How do human emotions differ from machine emotions?

Program Data Path
6-8.CD.1
Draw out the data path for what happens when a person clicks on a program to run it. Show that when a user clicks on a program, the computer will register the click, copy the file from the hard drive into RAM, and run the program. This activity can be drawn on whiteboard, blackboard, or any computer. Students can try to guess the path to start and the teacher can show the correct path. See Wiki Page for more information about how a program executes.

Replicated Troubleshooting
6-8.CD.2
Create a paper diagram labeled with parts of a machine or process. This could even apply to everyday processes, such as washing your hands or making a peanut butter and jelly sandwich. Label each piece of the diagram with its specific function. Remove one step in the diagram, and show how it breaks the system. Hide the removed piece of the process or machine and have students try to figure out what is missing.

Representing Data
6.8.PA.1
Have the students measure some arbitrary data (i.e. arm length, foot width, how far tables are away from the wall, how many siblings each student has, etc.). Compile all of the student's data into a single list. Teach students how to use a program like Excel or Google Sheets. Have them turn the data into different graphs and charts to represent the data. Show why differences in graph types and differences in the data will create differences in the results.

Technology Timeline

6.8.IC.2

Create a paper timeline of technological advancement. Get a large strip of paper and label it with dates. For example, the timeline may start in the 1800s and progress to modern day. For each technological invention, have students approximate where on the timeline the technology originated. Chart the dates for each evolving technology leading to modern technology. Possible technological inventions to include are the microwave oven, the iPod, the first cell phone, and websites such as Wikipedia or Google.

Troubleshooting Solutions

6-8.CD.2

Propose various computer issues and ask students how they would resolve the problems. If students have multiple solutions to the same problem, have them evaluate the solutions to determine which will be the most effective.

Further Resources

Name	Homepage	Focus	Paid Material	Coding Specific	Accounts
Alice Programming	http://www.alice.org/	Grades 3-12		Yes	No
Code Academy	https://www.codecademy.com/		Yes	Yes	Yes
Code Avengers	https://www.codeavengers.com/		Yes	Yes	Yes
Code School	https://www.codeschool.com/		Yes	Yes	Yes
Code.org	https://code.org/				
CodeCombat	https://codecombat.com/	Grades 4+	Yes	Yes	Yes
CodeHS	https://codehs.com/	High schoolers	Unsure	Yes	Yes
CS Teachers Association	https://www.csteachers.org/	Teacher Help			Yes
CS Unplugged	http://csunplugged.org/	No technology			
Free Code Camp	https://www.freecodecamp.org/			Yes	Yes
GitHub	https://github.com/			Yes	Yes
HackerRank	https://www.hackerrank.com/	Experienced coders		Yes	
Khan Academy	https://www.khanacademy.org/computing/computer-science	General			
Lightbot	http://lightbot.com/	Simulations	Yes		
Nextech	http://nextech.org/programs/csedweek/	Indiana			
Rosetta Code	https://rosettacode.org			Yes	
Scratch	https://scratch.mit.edu	Kids		Yes	
Spritebox	http://spritebox.com	Kids	Yes		
SQLZoo	http://sqlzoo.net/			Yes	
Teaching Kids Programming	http://teachingkidsprogramming.org/	Kids			
The Odin Project	https://www.theodinproject.com/	General		Yes	
Treehouse	https://teamtreehouse.com/		Yes	Yes	Yes
Tynker	https://www.tynker.com/	Grades 3-8	Yes	Yes	Yes
Upskill	http://upskillcourses.com/	General		Yes	