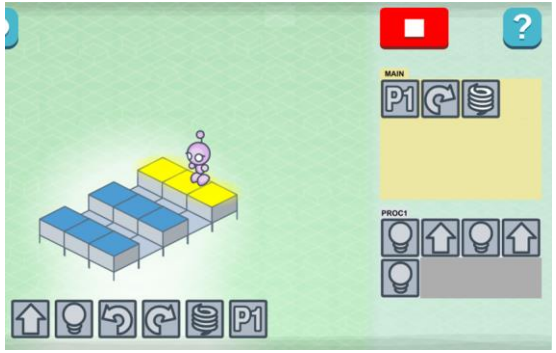
Computational Thinking Across the Curriculum

Computational Thinking (CT) Across the Curriculum

Let’s get started with an activity right away!

CT in ELA

- Thinking and communicating like a robot (procedural language)
- Graph paper programming-
  - Describe in natural language
  - Use symbols (abstraction) to communicate
- Lightbot
  - Use abstraction, patterns and algorithms to move the bot



ABCYA (ipad)

Goals for this session

- Define Computer Science (CS) and Computational Thinking (CT)
- Understand why it is important to include CT and CS in your curriculum
- Connect CT and CS to your curriculum
- Get you interested in learning more about CS and CT by participating in a future PD workshop!

Introductions

Paige Prescott

- CSTA-NM President
- Facilitator for Code.org and Project GUTS
- Current PhD student in Learning Sciences, focus on Computer Science education

Who is in the audience?

K-2? 3-5? Middle school math? Middle school science? other?

What is Computer Science (CS)?

Computer science is a discipline that involves the *understanding and design of computers, computational processes, and digital artifacts*

Learning to code/program a computer is *part* of CS but not the only thing studied in CS.

What is NOT Computer Science

CS is NOT just about *using* computers

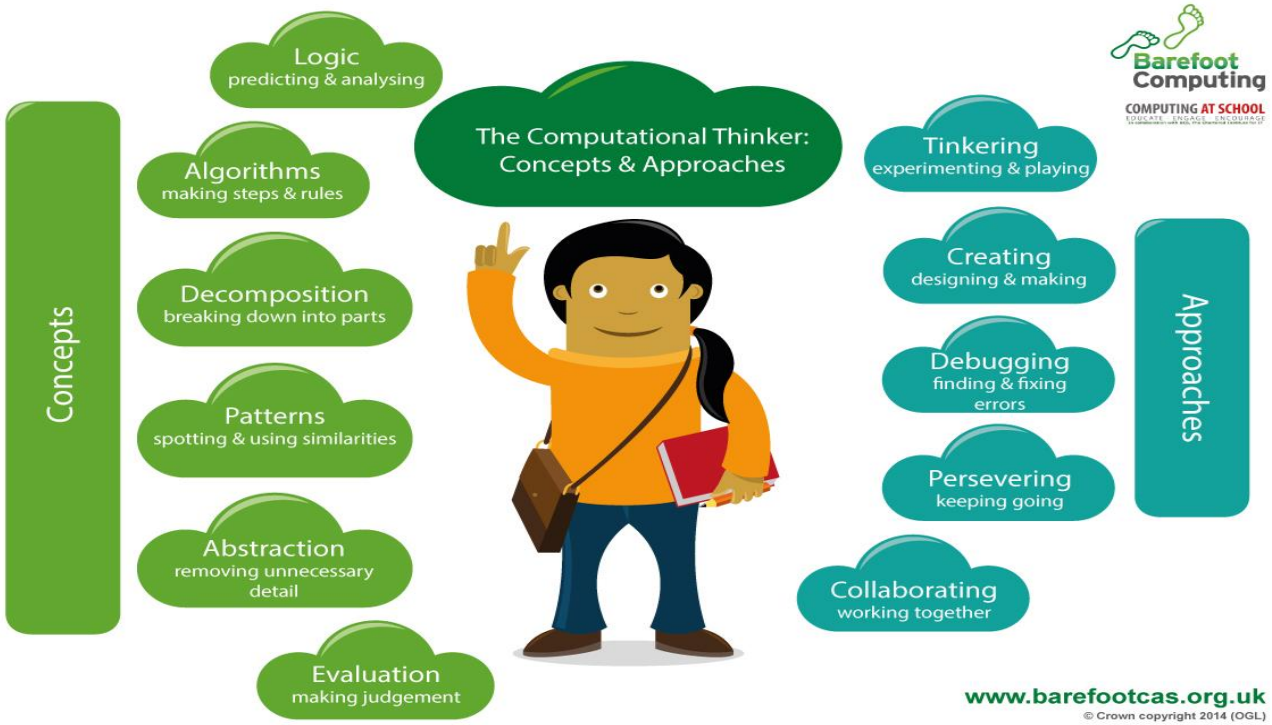
Using Computers	Doing Computer Science
Using a computer to search the internet for information for a presentation on a topic	- Learn how the internet works and how information is sent from computer to computer. - Learning how to do search the internet using advanced search options like Booleans.
Using a computer to fill out a form/quiz	Create a survey that classmates fill out, analyze the data in the spreadsheet, help make decisions based on that data.
Creating a presentation using Google Presentation or Powerpoint	Create/code an animation in StarLogo Nova or Scratch to have a character explain a concept or story.

What is Computational Thinking (CT)?

CT is a ***problem-solving process*** that includes (but is not limited to) the following characteristics:

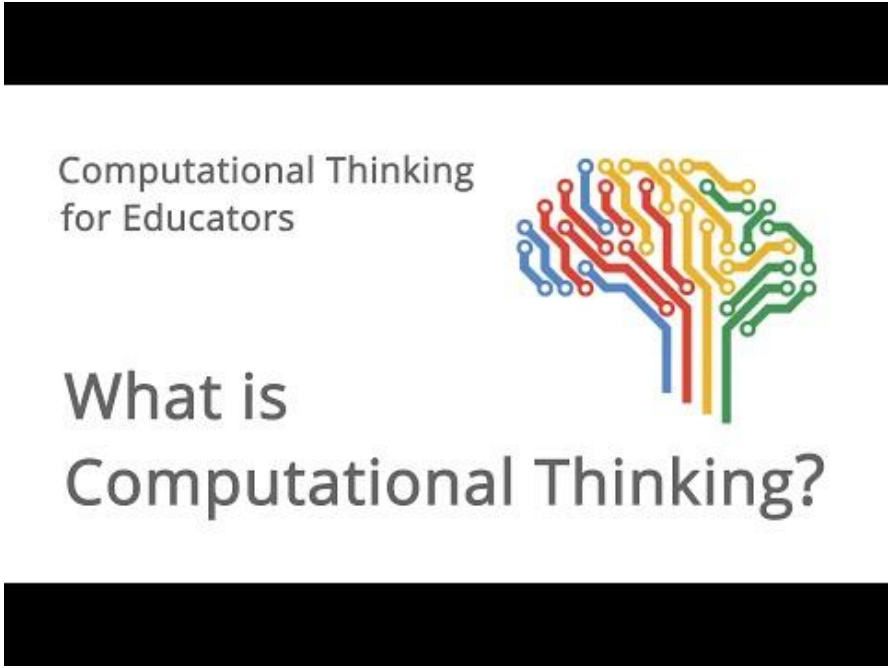
- **Formulating problems** in a way that enables us to use a computer and other tools to help solve them.
- **Logically organizing and analyzing data**
- **Representing data through abstraction** such as models and simulations
- **Automating solutions through algorithmic thinking** (a series of ordered steps)
- **Identifying, analyzing, and implementing possible solutions** with the goal of achieving the most efficient and effective combination of steps and resources
- **Generalizing and transferring this problem solving process** to a wide variety of problems

What is Computational Thinking (CT)?



Barefoot would like to acknowledge the work of Julia Briggs and the eLIM team at Somerset County Council for their contribution to this poster.

What is Computational Thinking (CT)?



Example of a CT activity

Sorting Algorithms

CS & CT connections

Computational thinking and the concepts behind it, form the basis for much of computer science.

Computer programming is all about describing a situation precisely, and giving good directions for what to do when conditions change.

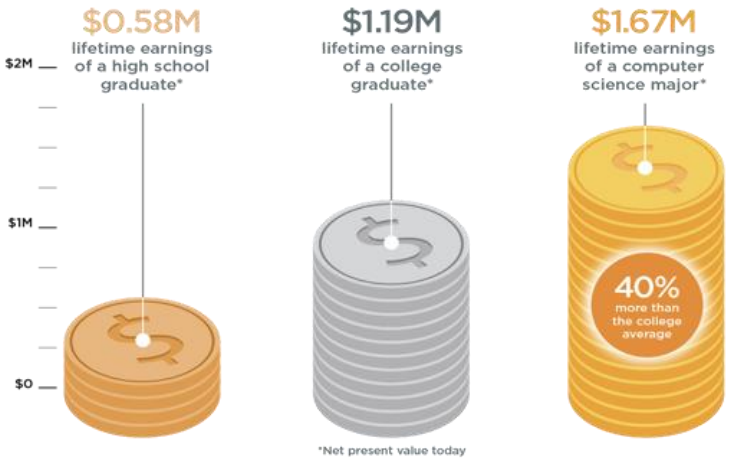
Critical Thinking + Computing Power = Making Decisions and Innovate Solutions

Consider these examples:

- If the temperature goes below 65°, turn on the heat
- When the drum solo starts, mute the guitar track and put a spotlight on the drummer
- It's just a jump to the left, and then a step to the right. With your hands on your hips, you bring your knees in tight.

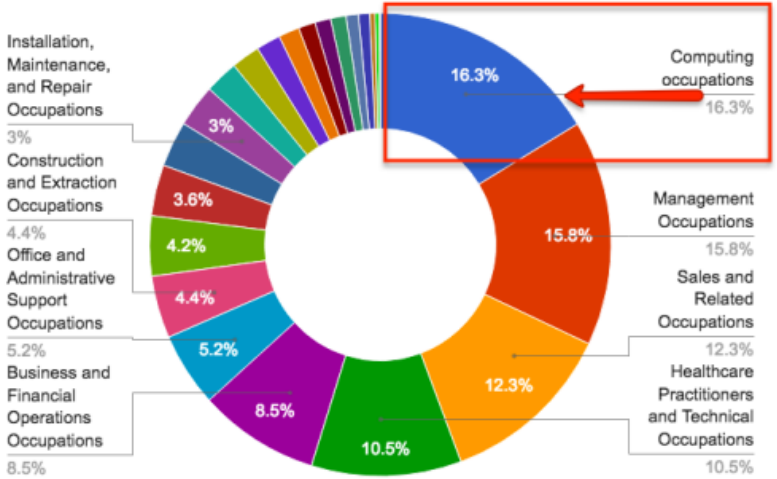
All of these statements embody computational thinking.
They could all be coded to run as computer programs, too!

Why CS & CT are important for your students



It's an economic issue for their future

It's where the jobs are in the future



Computing occupations are the largest category of new wages in the United States – ahead of management, healthcare, finance, engineering, sales, **or any other category**.

Source:

Source: BLS

Why CS & CT are important for your students

Misconception: Some think that the Tech Industry is trying to hire programmers in *California.*

Fact: Every industry is trying to hire computer scientists EVERYWHERE in the US

In New Mexico- 1,653 open computing jobs

- **3.5x the average demand rate in our state**
- **Average salary for these jobs $79,230, which is significantly higher than the average salary in the state ($44,160).**

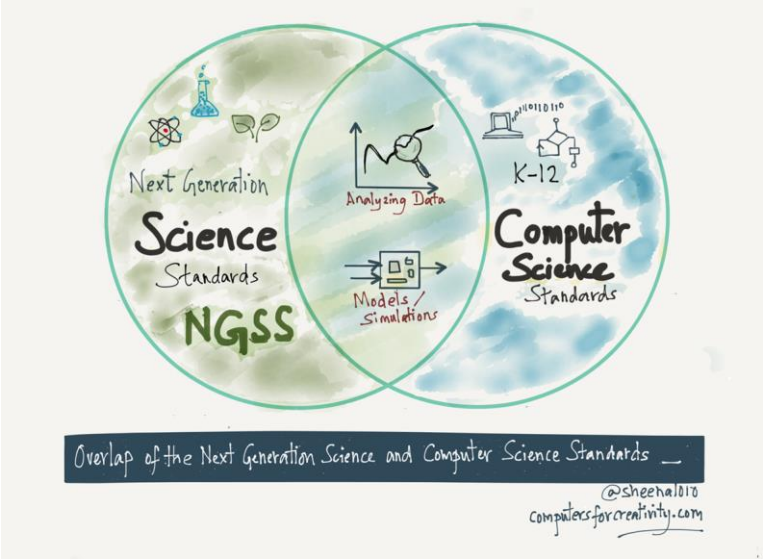
K12 Solutions | onramp to computer science

	EXPOSING/ INSPIRING	INTERDISC / PBL	ONLINE CODING CURRICULUM			CODING PLATFORMS / TOOLS
Elem	Elem/Mid PIPER LEGO education MINECRAFT	Elem/Mid/High PLTW	Elem/Mid Tynker CODESTERS codemonkey	Kodable	Elem/Mid/High CODE iTCH TechSmart Kids vidcode	SCRATCH
Mid		Mid/High Globaloria PROJECT GUTS BOOTSTRAP		Mid/High CodeHS pythonroom		repl.it codio GitHub Classroom trinket codenvy codeanywhere CODEPEN CLOUD9
High			bjc PLTW			

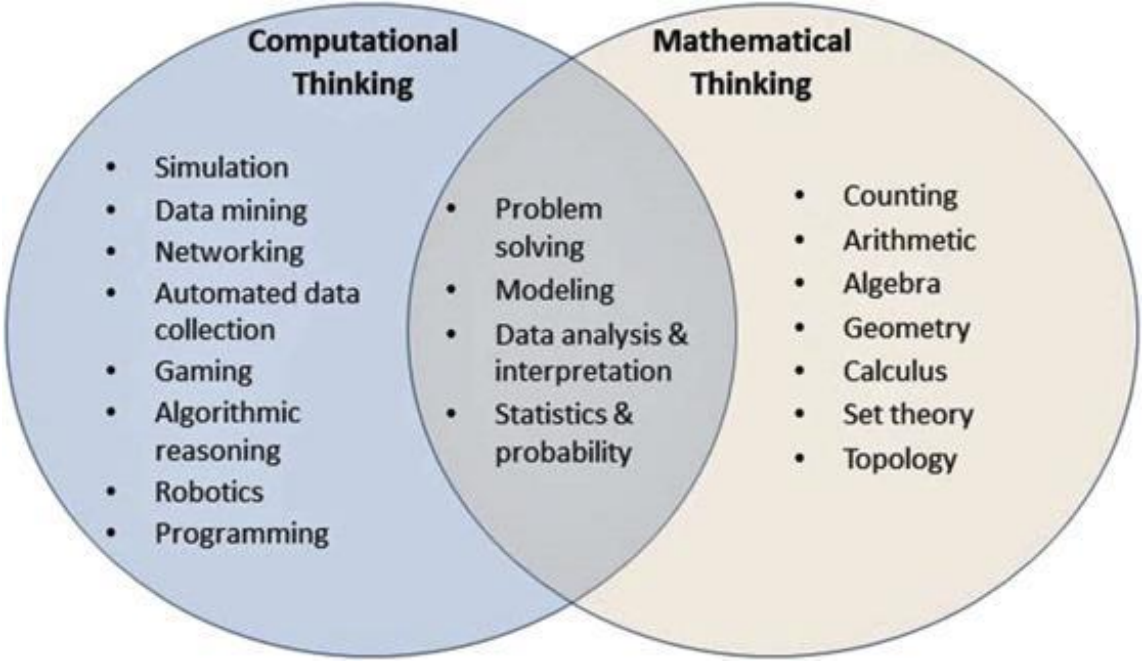
Science and CS/CT

NGSS Framework- 8 Scientific Practices

1. Asking questions and defining problems
2. Developing and using models
3. Planning and carrying out investigations
4. Analyzing and interpreting data
5. Using mathematics and computational thinking
6. Constructing explanations and designing solutions
7. Engaging in argument from evidence
8. Obtaining, evaluating, and communicating information



Math and CT


Computational Thinking
Mathematical Thinking
Simulation
Data mining
Networking
Automated data collection
Gaming
Algorithmic reasoning
Robotics
Programming
Problem solving
Modeling
Data analysis & interpretation
Statistics & probability
Counting
Arithmetic
Algebra
Geometry
Calculus
Set theory
Topology

CS/CT integration examples

Computational Thinking Concept	Subject Area Application
Break a problem into parts or steps	**Literature:** Break down the analysis of a poem into analysis of meter, rhyme, imagery, structure, tone, diction, and meaning.
Recognize and find patterns or trends	**Economics:** Find cycle patterns in the rise and drop of the country's economy.
Develop instructions to solve a problem or steps for a task	**Culinary Arts:** Write a recipe for others to use.
Generalize patterns and trends into rules, principles, or insights	**Mathematics:** Figure out the rules for factoring 2nd-order polynomials
	Chemistry: Determine the rules for chemical bonding and interactions.

CS/CT integration examples

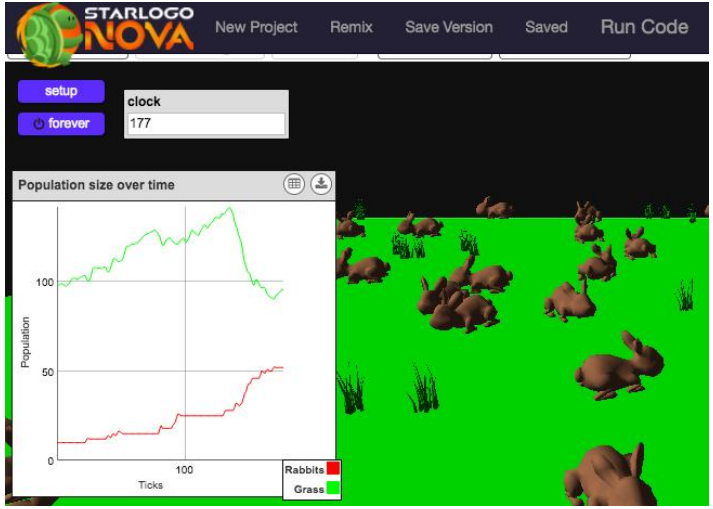
TABLE 1: CORE COMPUTATIONAL THINKING CONCEPTS AND CAPABILITIES

CT Concept, Capability	CS	Math	Science
Data collection	Find a data source for a problem area	Find a data source for a problem area, for example, flipping coins or throwing dice	Collect data from an experiment
Data analysis	Write a program to do basic statistical calculations on a set of data	Count occurrences of flips, dice throws and analyzing results	Analyze data from an experiment
Data representation	Use data structures such as array, linked list, stack, queue, graph, hash table, etc.	Use histogram, pie chart, bar chart to represent data; use sets, lists, graphs, etc. To contain data	Summarize data from an experiment
Problem Decomposition	Define objects and methods; define main and functions	Apply order of operations in an expression	Do a species classification
Abstraction	Use procedures to encapsulate a set of often repeated commands that perform a function; use conditionals, loops, recursion, etc.	Use variables in algebra; identify essential facts in a word problem; study functions in algebra compared to functions in programming; Use iteration to solve word problems	Build a model of a physical entity
Algorithms & procedures	Study classic algorithms; implement an algorithm for a problem area	Do long division, factoring; do carries in addition or subtraction	Do an experimental procedure
Automation		Use tools such as: geometer sketch pad; star logo; python code snippets	Use probeware
Parallelization	Threading, pipelining, dividing up data or task in such a way to be processed in parallel	Solve linear systems; do matrix multiplication	Simultaneously run experiments with different parameters
Simulation	Algorithm animation, parameter sweeping	Graph a function in a Cartesian plane and modify values of the variables	Simulate movement of the solar system

Curriculum that integrates CT and CS

Project GUTS

an award-winning CS in Science curriculum that has students explore building computer models and using simulations to learn about our world. The curriculum uses a block-based language that allows students to learn both science and computer science skills as they build models of epidemics, ecosystems, chemical reactions, climate change or hydrology.



Curriculum that integrates CT and CS

Bootstrap

Curriculum aligned to National and State Standards for Mathematics, covering most Functional and Algebraic standards from Grade 7 through Algebra 2. This alignment makes it possible to integrate Bootstrap into the classroom smoothly, using time you've already planned into your pacing guidelines or scope and sequence plans.

For states using the Common Core, Bootstrap is also a model implementation of Common Core Standards for Mathematical Practice, offering explicit pedgagogical recommendation across all eight practice standards.

BOOTSTRAP

Equity • Scale • Rigor

CT and Social Studies

Plan a presentation about at least 5 important sites in NM.

- Which sites will you choose and why?
- How will you communicate your information to an audience?

CT and Social Studies

Now, plan to communicate this same information by making a program in Scratch.

Why integrate CT and CS?

When students learn computational thinking and computer science skills across the curriculum it become internalized and can be easily transferred from one setting to another. Students develop skills that can be applied in a variety of situations—in other classes, in their hobbies, in their community—from a variety of perspectives and in an authentic setting. As more and more teachers emphasize these skills, students will begin to apply them naturally in new and exciting ways.

CT Resources

ISTE & CSTA- https://www.iste.org/explore/articleDetail?articleid=152&category=Solutions&article=Computational-thinking-for-all

CS Framework- https://k12cs.org/computational-thinking/

Google- https://edu.google.com/resources/programs/exploring-computational-thinking/#!home

http://www.gettingsmart.com/2018/05/10-classroom-ready-computational-thinking-resources-for-k-12/

Discussion

Discuss these questions with colleagues near you:

How are you already doing CT in your curriculum? How can you do more CT?

How are you already doing CS in your curriculum? How can you do more CS?

Be ready to share your ideas and summary of your discussion.