Arizona Computer Science Standards

STATE BOARD OF EDUCATION MEETING

October 22, 2018

Purpose

To present to the Arizona State Board of Education the final draft of the **Arizona Computer Science Standards** for consideration and adoption.

Governor’s Office of Education

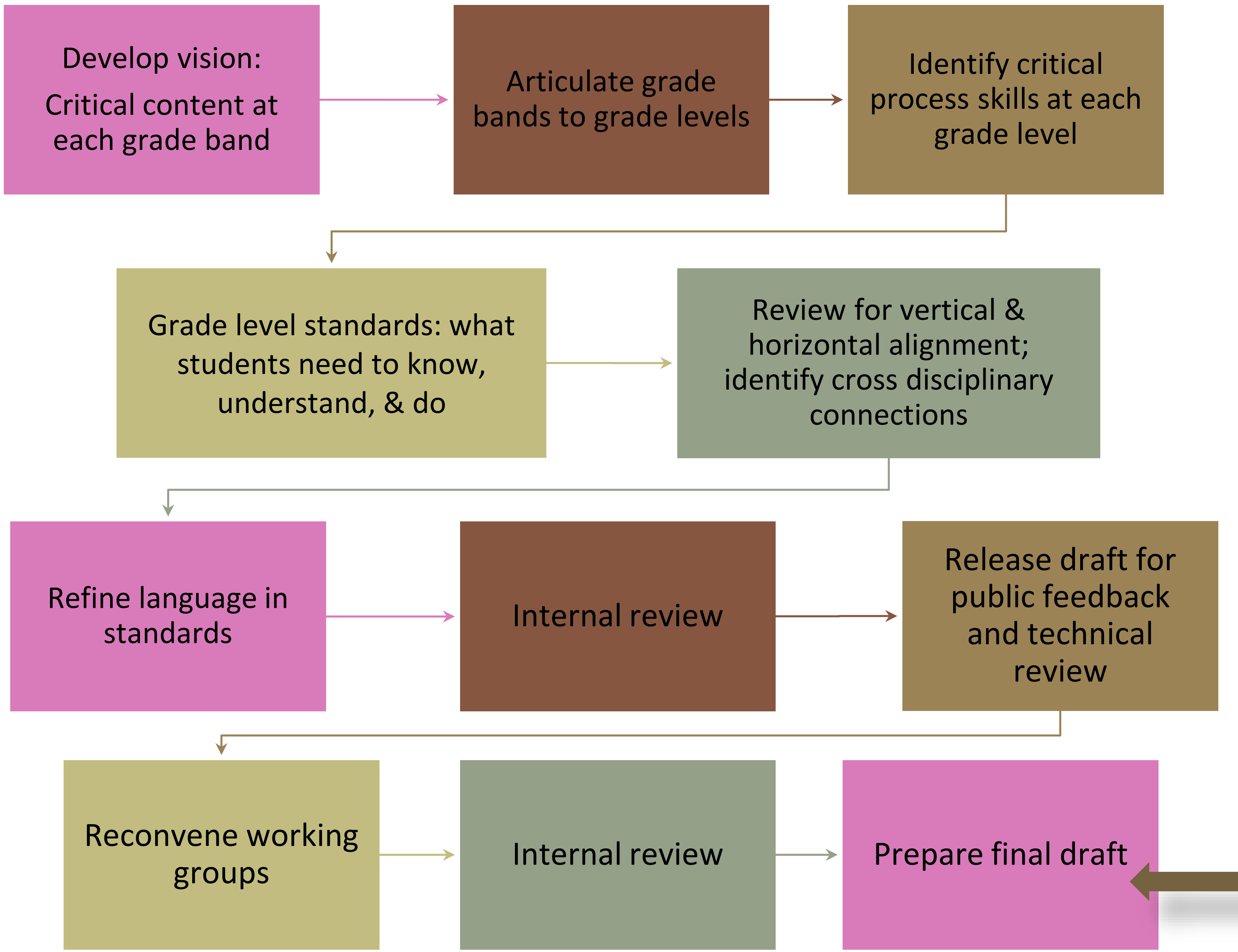
- Governor’s Office of Education was appropriated $200,000 during the 2017 legislative session to support the development of computer science standards for K-12.

- K-12 Academic Standards, in collaboration with the Governor’s office, convened educators, content experts, and other stakeholders to develop standards.

Standards Development Process

An on-going cycle of improvement

Informed by research & public comment



Standards, Curriculum, & Instruction

Standards – What a student needs to know, understand, and be able to do by the end of each grade. Standards build across grade levels in a progression of increasing understanding and through a range of cognitive demand levels. Standards are adopted at the state level by the State Board of Education.

This is the "WHAT"

Standards, Curriculum, & Instruction

Curriculum – The resources used for teaching and learning the standards. Curricula are adopted at a local level by districts and schools.

Instruction – The methods used by teachers to teach their students. Instructional techniques are employed by individual teachers in response to the needs of the students in their classes to help them progress through the curriculum in order to master the standards.

This is the "HOW"

Computer Science Standards

<u>Working Groups</u>

- **Over 60 Educators**
  - Representing
    - **41 schools districts, charters, and institutes of higher education**
    - **8 counties**: *Apache, Cochise, Coconino, Maricopa, Navajo, Pima, Yavapai, Yuma*
  - ADE Interagency Colleagues from Career and Technical Education

All working group agendas and presentations are located on the Computer Science Standards Review Update Page.

http://www.azed.gov/standards-practices/arizona-computer-science-standards-dev/

Thank You Arizona Educators!

Research Base

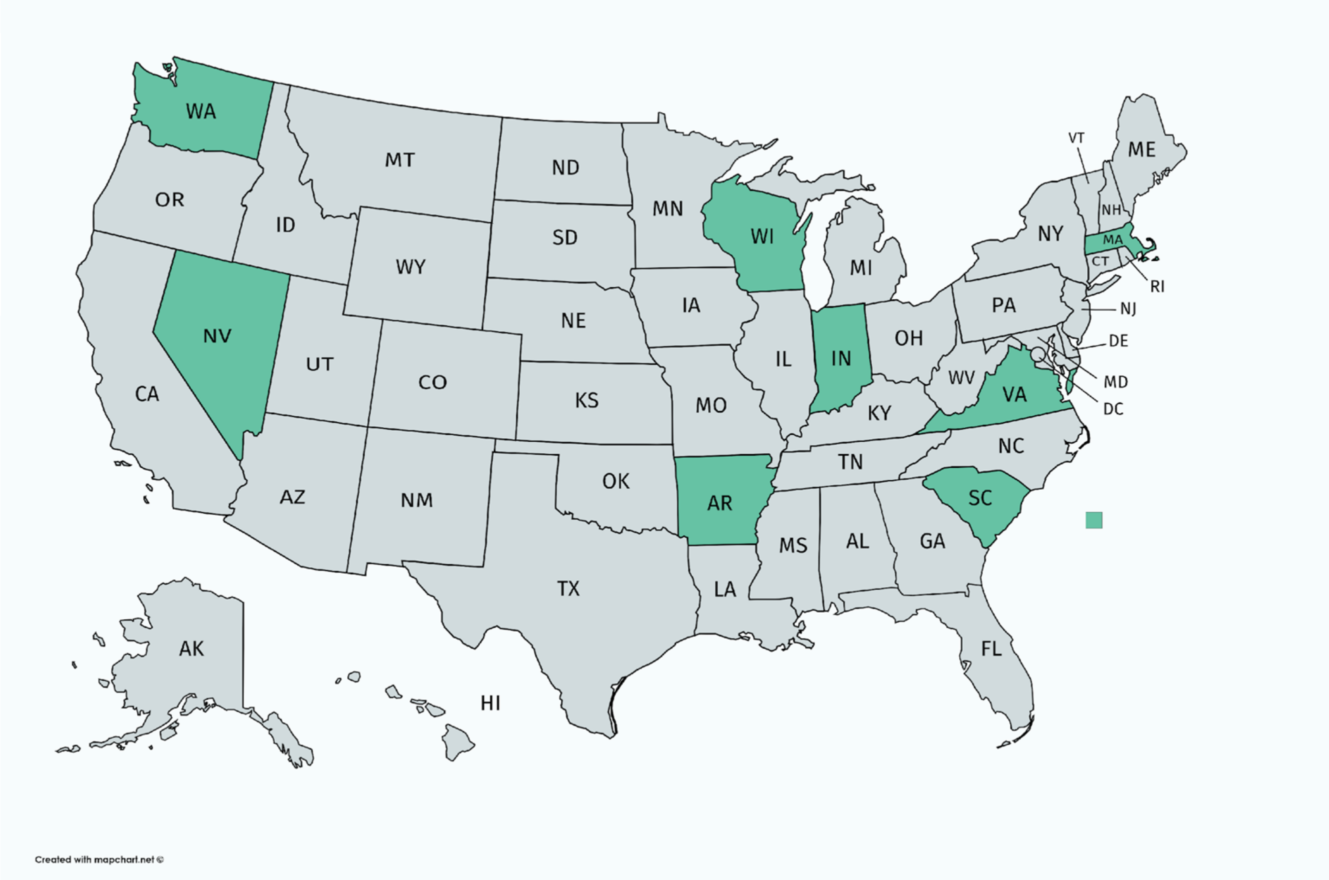
CSTEACHERS.ORG

COMPUTER SCIENCE TEACHERS ASSOCIATION

K-12 Computer Science Standards, Revised 2017

K12 COMPUTER SCIENCE FRAMEWORK

2016



Computer Science Standards from other states (2016-2017)

Key Themes From Public Comment

- ✓ Limited time to teach computer science
- ✓ The ability to prepare, train, and recruit qualified teachers for computer science
- ✓ Examples of how to implement the standards
- ✓ Readability: too narrow, too broad, too lengthy

Technical Reviewers

Dr. Ashish Amresh, Embry-Riddle University

Assistant Professor and Program Chair, Simulation Science, Games, and Animation

Dr. LeeAnn Lindsey, Edvolve

Education Technology and Digital Literacy Consultant

Dr. Brian Nelson, Arizona State University

Professor of Education Technology

Dr. Chris Stephenson, Google, Inc

Head of Computer Science Education Programs

Dr. Alicia Nicki Washington, Winthrop University

Associate Professor of Computer Science, Member of K-12 Computer Science Framework Writing Group

Overarching Theme from Technical Review

These standards will enhance opportunities for students to <u>**solve problems**</u>, <u>**think critically and computationally**</u>, and <u>**prepare students for the future**</u>.

Key Themes From Technical Review

- ✓ Consistency in grade level storylines
- ✓ Clarity in the vision
- ✓ Clarity in the introduction
- ✓ Emphasis and articulation of equity
- ✓ Requirement for implementing the standards

Research Base

Essential Concepts and Subconcepts in Computer Science

Computing Systems

- *Devices*
- *Hardware and Software*
- *Troubleshooting*

Networking and the Internet

- *Network Communication and Organization*
- *Cybersecurity*

Data and Analysis

- *Collection, Visualization and Transformation*
- *Storage*
- *Inference Models*

Algorithms and Programming

- *Algorithms*
- *Variables*
- *Control*
- *Modularity*
- *Program Development*

Impacts of Computing

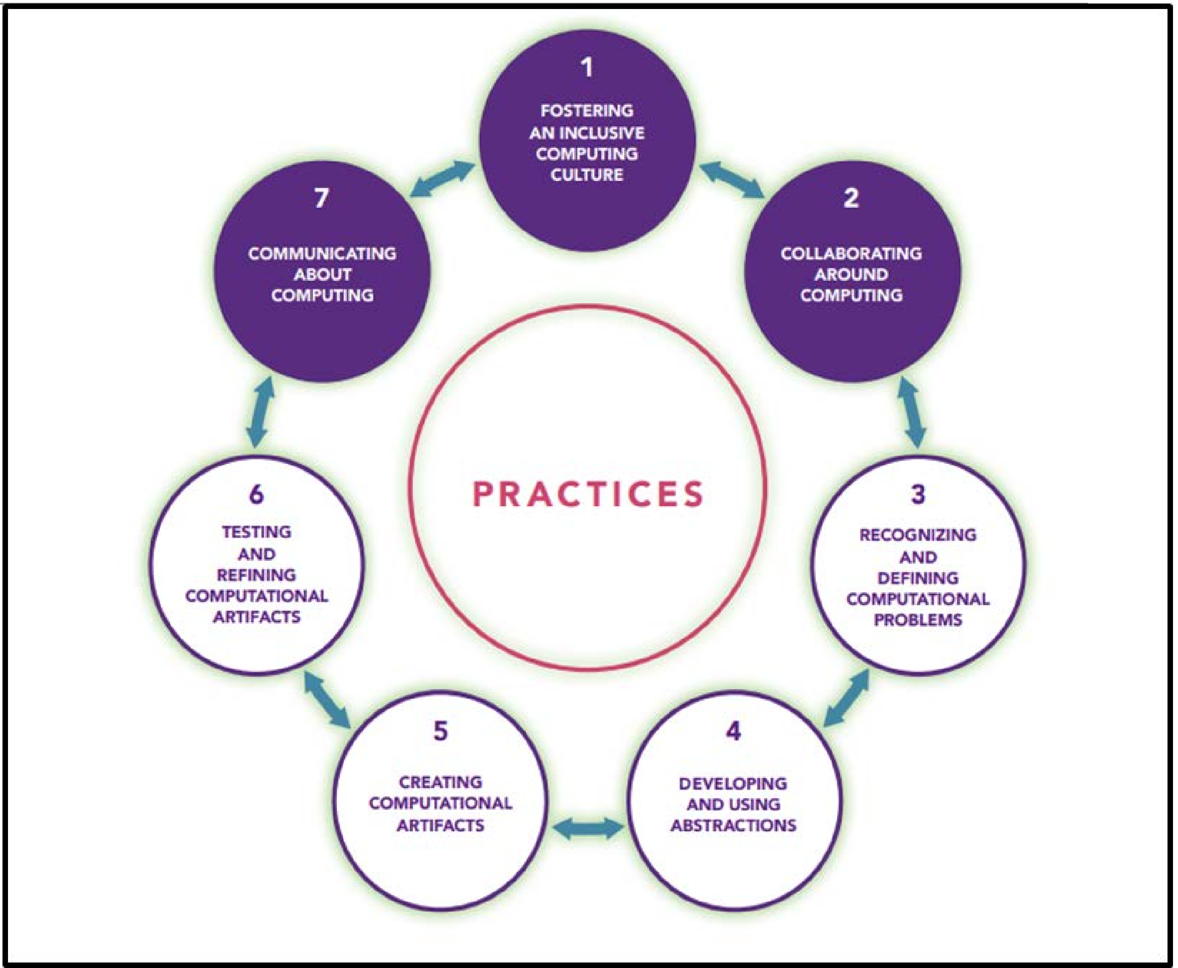
- *Culture*
- *Social Interactions*
- *Safety, Law, and Ethics*

Research Base

Computer Science Practices Including Computational Thinking

Practices 1, 2, and 7
General practices of CS that support computational thinking

Practices 3-6:
Computational thinking practices



Computational Thinking is……

- The **thought processes involved in expressing solutions** as computational steps or algorithms that can be carried out by a computer

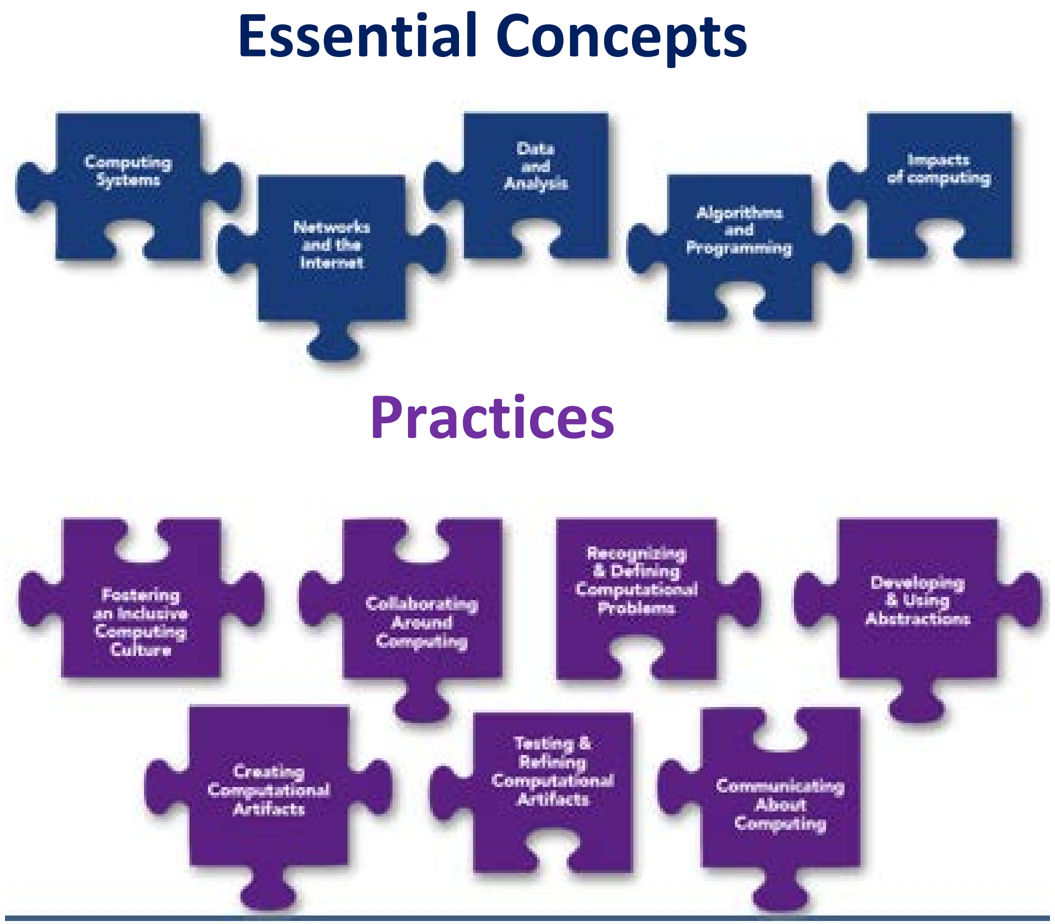
- Essentially a **problem-solving process that involves designing solutions** that capitalize on the power of computers

- **Beyond the borders of computer science to a variety of disciplines**, such as science, technology, engineering, and mathematics (STEM), but also the arts and humanities

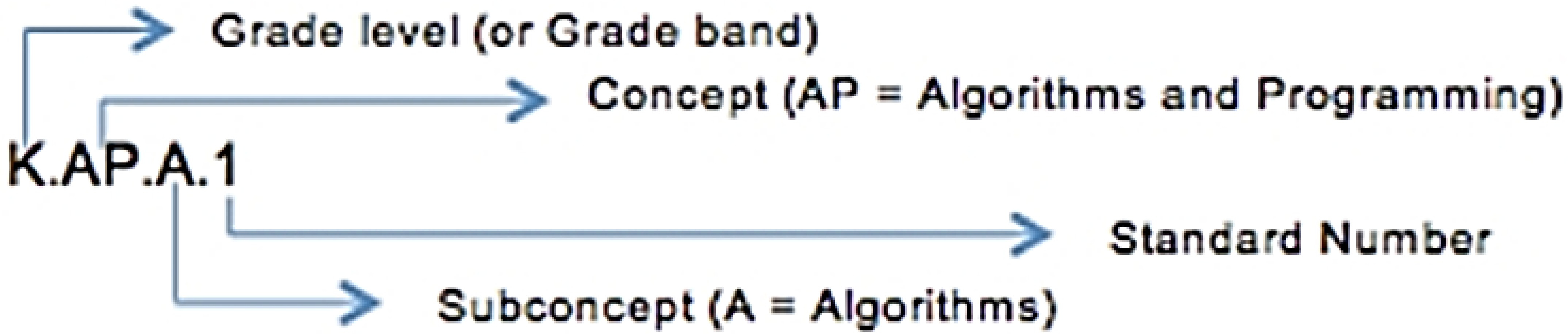
Research Base

Computer Science Practices Including Computational Thinking

Multiple concepts and practices fit together to create meaningful experiences in computer science.



Coding of the Computer Science Standards



Components of the Computer Science Standards

	8th Grade	High School
	8.CS.HS.1	**HS.CS.HS.1**
Standard	Design and evaluate projects that combine hardware and software components to collect and exchange data.	Describe levels of abstraction and interactions between application software, system software, and hardware layers.
Description	Collecting and exchanging data involves input, output, storage, and processing. When possible, students should select the hardware and software components for their project designs by considering factors such as functionality, cost, size, speed, accessibility, and aesthetics. For example, components for a mobile app could include: accelerometer, GPS, and speech recognition. The choice of a device that connects wirelessly through a Bluetooth connection versus a physical USB connection involves a tradeoff between mobility and the need for an additional power source for the wireless device.	At its most basic level, a computer is composed of physical hardware and electrical impulses. Multiple layers of software are built upon the hardware and interact with the layers above and below them to reduce complexity. System software manages a computing device's resources so that software can interact with hardware. For example, text editing software interacts with the operating system to receive input from the keyboard, convert the input to bits for storage, and interpret the bits as readable text to display on the monitor. System software is used on many different types of devices, such as smart TVs, assistive devices, virtual components, cloud components, and drones. For example, students may explore the progression from voltage to binary signal to logic gates to adders and so on. Knowledge of specific, advanced terms for computer architecture, such as BIOS, kernel, or bus, is not expected at this level.
Practices	Practice(s): Creating Computational Artifacts: 5.1	Practice(s): Developing and Using Abstractions: 4.1

K12 COMPUTER SCIENCE FRAMEWORK

Research Base
Computing Systems

Computing Systems

- *Devices*
- *Hardware and Software*
- *Troubleshooting*

People interact with a wide variety of computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively.
The physical components that make up a computing system, computer, or computing device (Hardware); and the programs that run on a computer system, computer, or other computing device
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.

Computing Systems: Grades K-2

Hardware and Software

4th Grade	6th Grade	8th Grade
4.CS.HS.1	**6.CS.HS.1**	**8.CS.HS.1**
Recognize that bits serve as the basic unit of data in computing systems and can represent a variety of information.	Explain how hardware and software can be used to collect and exchange data.	Design and evaluate projects that combine hardware and software components to collect and exchange data.
Hardware and software communicate in binary digits commonly represented in 0s and 1s. Students discuss how bits are a unit of data.	Collecting and exchanging data involves input, output, storage, and processing. For example, students can describe how components of a device are used to collect data. Such components might include: accelerometer, Global Position System (GPS), microphone, fingerprint sensor, etc.	Collecting and exchanging data involves input, output, storage, and processing. When possible, students should select the hardware and software components for their project designs by considering factors such as functionality, cost, size, speed, accessibility, and aesthetics. For example, components for a mobile app could include: accelerometer, GPS, and speech recognition. The choice of a device that connects wirelessly through a Bluetooth connection versus a physical USB connection involves a tradeoff between mobility and the need for an additional power source for the wireless device.
Practice(s): Communicating About Computing: 7.2	Practice(s): Creating Computational Artifacts: 5.1	Practice(s): Creating Computational Artifacts: 5.1

Research Base
Networks and the Internet

Networking and the Internet	
• *Network Communication and Organization* • *Cybersecurity*	Computing devices do not operate in isolation.
	Networks connect computing devices to share information and resources and are an increasingly integral part of computing.
	Networks and communication systems provide greater connectivity in the computing world by providing fast, secure communication and facilitating innovation.

K12 COMPUTER SCIENCE FRAMEWORK

Research Base
Networks and the Internet

Networks and communication systems provide greater connectivity in the computing world by providing fast, secure communication and facilitating innovation.

Networks and the Internet: Grades 3, 7, and High School

Cybersecurity

3rd Grade	7th Grade	High School
3.NI.C.1	**7.NI.C.1**	**HS.NI.C.1**
Identify real-world cybersecurity problems and how personal information can be protected.	Evaluate multiple methods of encryption for the secure transmission of information.	Describe how sensitive data can be affected by malware and other attacks.
Just as we protect our personal property online, we need to protect our devices and the information stored on them. Information can be protected using various security measures. These measures can be physical and/or digital. For example, discussion topics could be based on current events related to cybersecurity or topics that are applicable to students and the programs/devices they use such as adding passwords to lock devices.	Encryption can be as simple as letter substitution or as complicated as modern methods used to secure networks and the Internet. The students will examine the different levels of complexity used to hide or secure information. For example, students explore different methods of securing messages using methods such as Caesar ciphers or steganography (i.e., hiding messages inside a picture or other data).	Network security depends on a combination of hardware, software, and practices that control access to data and systems. Potential security problems, such as denial-of-service attacks, ransomware, viruses, worms, spyware, and phishing, present threats to sensitive data. Students might reflect on case studies or current events in which governments or organizations experienced data leaks or data loss as a result of these types of attacks.
Practice(s): Communicating about Computing, Recognizing and Defining Computational Problems: 7.1, 3.1	Practice(s): Developing and Using Abstractions: 4.4	Practice(s): Communicating About Computing: 7.2

Networks and the Internet: Grades 7 and High School (Cnt'd)

Cybersecurity

7th Grade	High School
7.NI.C.2	**HS.NI.C.2**
Explain how physical and digital security measures protect electronic information.	Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.
Information that is stored online is vulnerable to unwanted access. Examples of physical security measures to protect data include keeping passwords hidden, locking doors, making backup copies on external storage devices, and erasing a storage device before it is reused. For example, digital security measures include secure router admin passwords, firewalls that limit access to private networks, and the use of a protocol such as HTTPS to ensure secure data transmission.	Security measures may include physical security tokens, two-factor authentication, and biometric verification. The timely and reliable access to data and information services by authorized users, referred to as availability, and is ensured through adequate bandwidth, backups, and other measures. Students should systematically evaluate different security measures based on the requirements or constraints of a situation, such as through a cost-benefit analysis. Eventually, students should include more factors in their evaluations, such as how efficiency affects feasibility or whether a proposed approach raises ethical concerns, and make recommendations based on their analysis.
Practice(s): Communicating About Computing: 7.2	Practice(s): Recognizing and Defining Computational Problems: 3.3

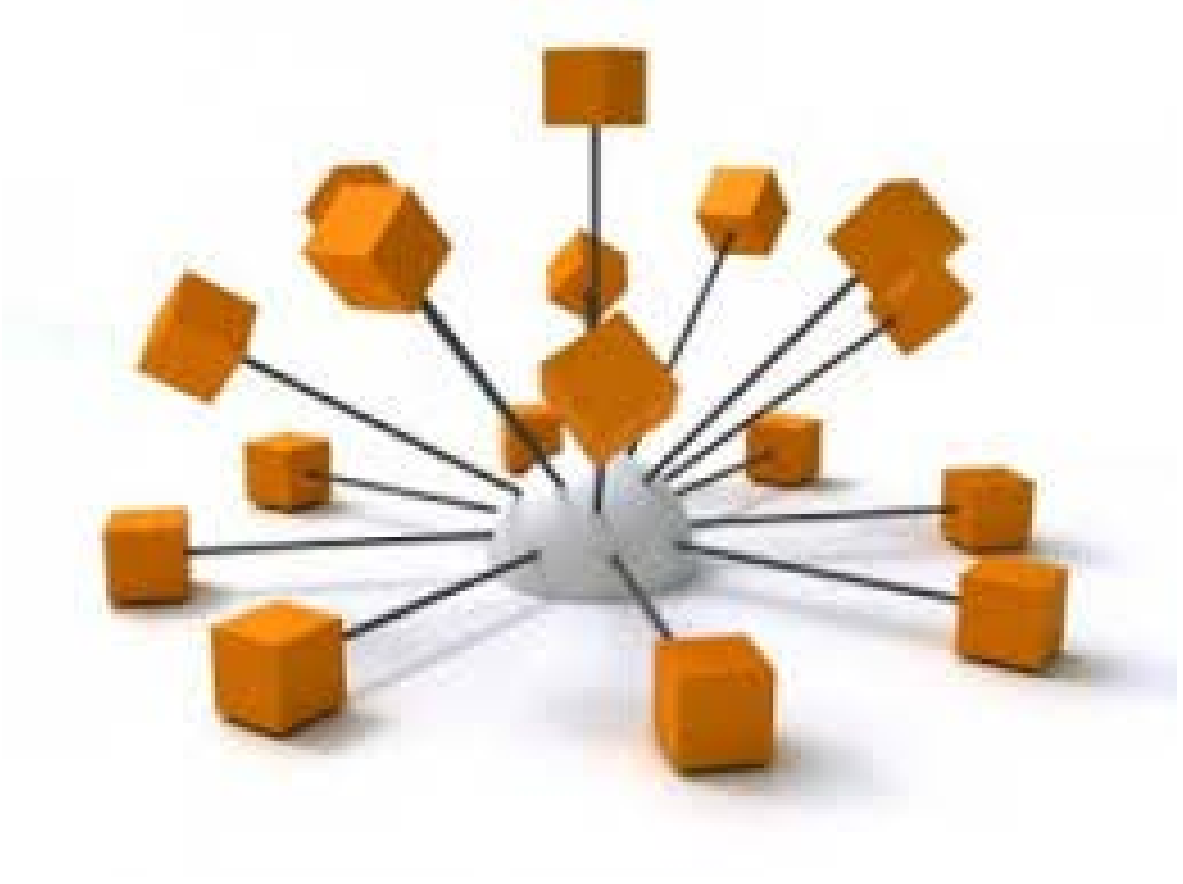
Research Base
Data and Analysis

Data and Analysis

- *Collection, Visualization and Transformation*
- *Storage*
- *Inference Models*

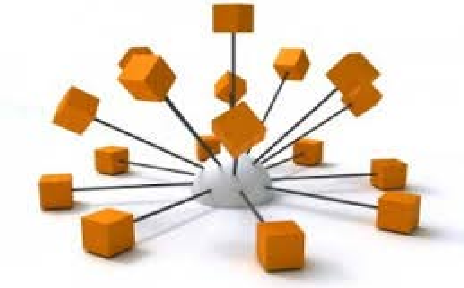
Computing Systems exist to process data.
The amount of digital data generated in the world is rapidly expanding, so the need to process data effectively is increasingly important.
Data is collected and stored so that it can be analyzed to better understand the world and make more accurate predictions.

Research Base
Data and Analysis



Data and Analysis

Collection, Visualization, and Transformation



Kindergarten	3rd Grade	6th Grade
K.DA.CVT.1	**3.DA.CVT.1**	**6.DA.CVT.1**
With teacher guidance, collect and transform data using digital devices; Display data for communication in various visual formats.	Select tools from a specified list to collect, organize, and present data visually to highlight relationships and support a claim.	Compare different computational tools used to collect, analyze and present data that is meaningful and useful.
The collection and use of data about the world around them is a routine part of life and influences how people live. Many everyday objects, such as cell phones, digital toys, and cars, can contain tools (such as sensors) and computers to collect and display data from their surroundings. Students could collect data on the weather, such as sunny days versus rainy days, the temperature at the beginning of the school day and end of the school day, or the inches of rain over the course of a storm. Students could count the number of pieces of each color of candy in a bag of candy, such as Skittles or M&Ms. Students could create surveys of things that interest them, such as favorite foods, pets, or TV shows, and collect answers to their surveys from their peers and others. The data collected could then be organized into two or more visualizations, such as a bar graph, pie chart, or pictograph.	Tools are chosen based upon the type of measurement they use as well as the type of data people wish to observe. Organizing data can make interpreting and communicating it to others easier. Data points can be clustered by a number of commonalities.	As students continue to explore ways to gather, organize and present data visually to support a claim, they will need to understand when and how to transform data for this purpose. Examples of these computational tools could include Microsoft Excel and Google Sheets.
Practice(s): Communicating About Computing, Developing and Using Abstractions: 7.3, 4.4	Practice(s): Developing and Using Abstractions, Creating Computational Artifacts: 4.1, 5.1	Practice(s): Testing and Refining Computational Artifacts: 6.3

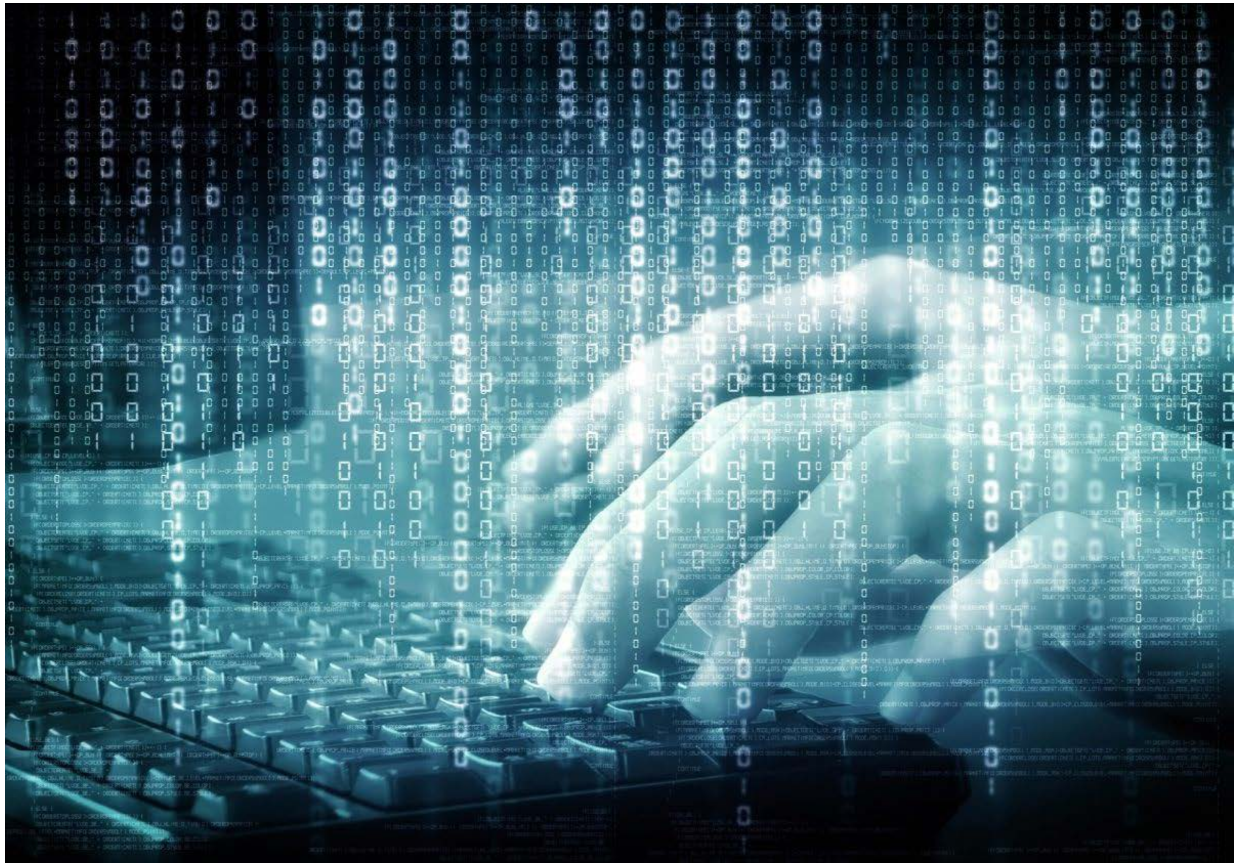
Research Base
Algorithms and Programming

Algorithms and Programming

- *Algorithms*
- *Variables*
- *Control*
- *Modularity*
- *Program Development*

An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices.
Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems.
The development process to create meaningful and efficient programs involves choosing which information to use and how to process and store it, breaking apart large problems into smaller ones, recombining existing solutions, and analyzing different solutions.

Research Base
Algorithms and Programming



Algorithms and Programming
Program Development

2nd Grade	5th Grade	8th Grade
2.AP.PD.1	**5.AP.PD.1**	**8.AP.PD.1**
Develop plans that describe a program's sequence of events, goals, and expected outcomes.	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
Programming is used as a tool to create products that reflect a wide range of interests, such as video games, interactive art projects, and digital stories. Students could create a planning document, such as a story map, a storyboard, or a sequential graphic organizer, to illustrate what an end product will do. Students at this stage may complete the planning process with help from their teachers. For example, students create a graphic organizer modeling the life cycle of a plant.	Planning is an important part of the iterative process of program development. Students outline key features, time and resource constraints, and user expectations. Students should document the plan as, for example, a storyboard, flowchart, pseudocode, or story map.	Development teams that employ user-centered design create solutions (e.g., programs and devices) that can have a large societal impact, such as an app that allows people with speech difficulties to translate hard-to-understand pronunciation into understandable language. Students should begin to seek diverse perspectives throughout the design process to improve their computational artifacts. Considerations of the end-user may include usability, accessibility, age-appropriate content, respectful language, user perspective, pronoun use, color contrast, and ease of use.
Practice(s): Creating Computational Artifacts, Communicating About Computing: 5.2, 7.2	Practice(s): Fostering an Inclusive Computing Culture, Creating Computational Artifacts: 1.1, 5.1	Practice(s): Collaborating Around Computing, Fostering an Inclusive Computing Culture: 2.3, 1.1

Research Base
Impacts of Computing

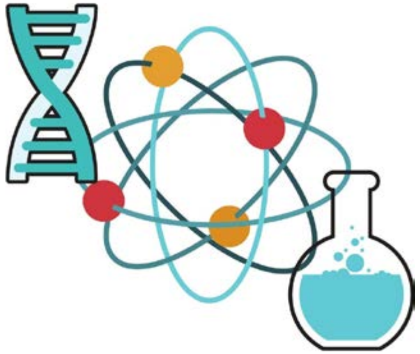
Impacts of Computing	
• *Culture* • *Social Interactions* • *Safety, Law, and Ethics*	Computing affects many aspects of the world in both positive and negative ways at local, national, and global levels.
	Individuals and communities influence computing through their behaviors and cultural and social interactions, and in turn, computing influences new cultural practices.
	An informed and responsible person should understand the social implications of the digital world, including equity and access to computing.

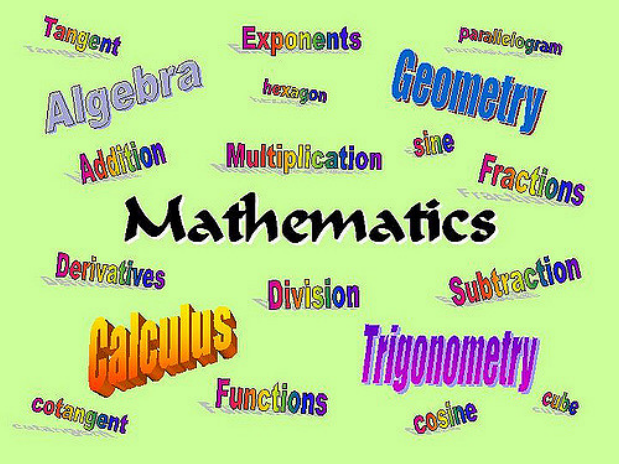
Research Base
Impacts of Computing

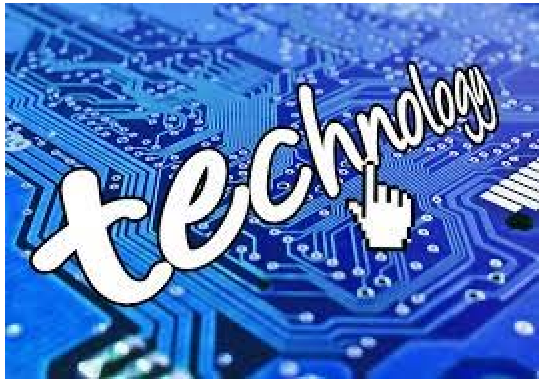
Impacts of Computing
Culture

5th Grade	7th Grade	High School
5.IC.C.1	**7.IC.C.1**	**HS.IC.C.1**
Discuss computing technologies that have changed the world.	Explain how some of the tradeoffs associated with computing technologies can affect people's everyday activities and career options.	Evaluate the ways access to computing impacts personal, ethical, social, economic, and cultural practices.
New computing technology is created and existing technologies are modified for many reasons, including in order to increase their benefits, decrease their risks, and meet societal needs. Students discuss topics that relate to the history of technology and the changes in the world due to technology. Students discuss how culture influences changes in technology. Topics could be based on current news content in areas, such as robotics, wireless Internet, mobile computing devices, GPS systems, wearable computing, or how social media has influenced social, cultural and political changes.	Advancements in computer technology are neither wholly positive nor negative. However, the ways that people use computing technologies have tradeoffs. Students should consider current events related to broad ideas, including privacy, communication, and automation. For example, driverless cars can increase convenience and reduce accidents, but they are also susceptible to hacking. The emerging industry will reduce the number of taxi and shared-ride drivers, but will create more software engineering and cybersecurity jobs.	Computing may improve, harm, or maintain practices. Equity deficits, such as minimal exposure to computing, access to education, and training opportunities, are related to larger, systemic problems in society. Students should be able to evaluate the accessibility of a product to a broad group of end users, such as people who lack access to broadband or who have various disabilities.
Practice(s): Recognizing and Defining Computational Problems: 3.1	Practice(s): Communicating About Computing: 7.2	Practice(s): Fostering an Inclusive Computing Culture: 1.2

Connection to Other Disciplines

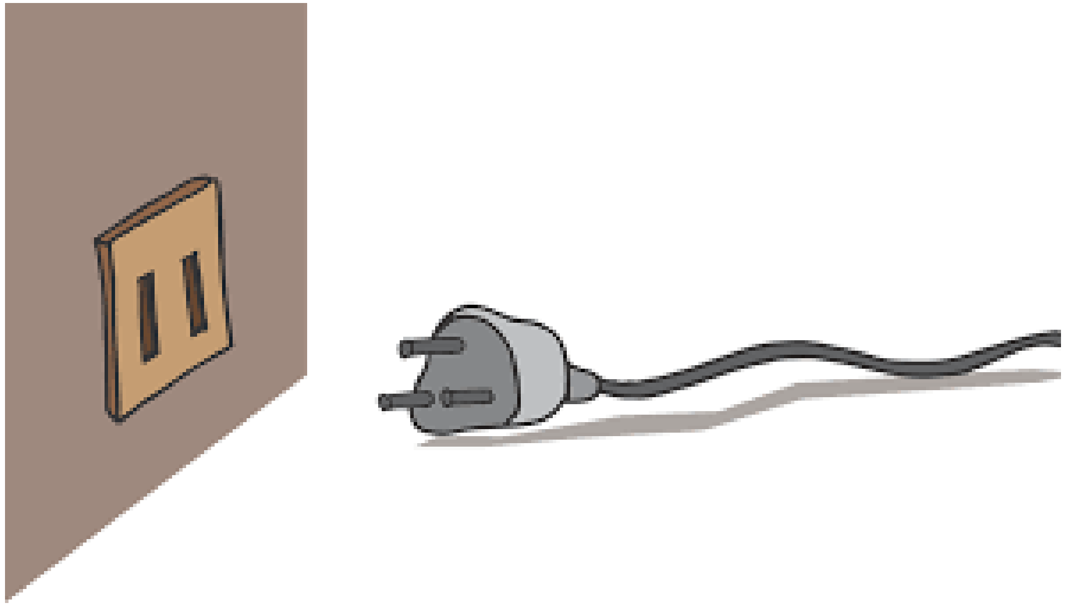






How Do We Teach Computer Science?

Plugged



Unplugged

Next Steps

- ➢Implementation of the standards will begin during 2020-2021 school year
- ➢ADE Implementation Support
  - Professional Development
    - CsforAZ PD Week
    - CSforALL SCRIPT Training
    - ADE Webinars and Guidance Documents
  - Computer Science Professional Development Program Fund


Questions?