Computer Science Essential Concepts And Subconcepts

Concept: Computing Systems

Kindergarten - Highschool

Computer Science Essential Concepts and Subconcepts

The Arizona Computer Science Standards for grades kindergarten through twelve are organized into five Essential Concepts:

- **Computing Systems:** This involves the interaction that people have 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 (hardware) and instructions (software) that make up a computing system communicate and process information in digital form. An understanding of hardware and software is useful when troubleshooting a computing system that does not work as intended. Computing Systems has three subconcepts, they are: Devices, Hardware and Software, and Troubleshooting.

- **Networks and the Internet (with Cybersecurity):** This involves the networks that connect computing systems. 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. Networking and the Internet must also consider Cybersecurity. Cybersecurity, also known as information technology security, involves the protection of computers, networks, programs, and data from unauthorized or unintentional access, manipulation, or destruction. Many organizations, such as government, military, corporations, financial institutions, hospitals, and others collect, process, and store significant amounts of data on computing devices. That data is transmitted across multiple networks to other computing devices. The confidential nature of government, financial, and other types of data requires continual monitoring and protection for the sake of continued operation of vital systems and national security. This concept has two subconcepts within it, they are: Cybersecurity, and Network Communication and Organization.

- **Data and Analysis:** This involves the data that exist and the computing systems that exist to process that 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. This concept has three subconcepts, they are: Collection, Visualization and Transformation, Storage, and Inference and Models

- **Algorithms and Programming:** Involves the use of algorithms. 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. This concept has 5 subconcepts, they are: Algorithms, Variables, Control, Modularity, and Program Development

- **Impacts of Computing:** This involves the effect that computing has on daily life. 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. This concept has 3 subconcepts, they are: Culture, Social Interactions, and Safety, Law, and Ethics

Concepts are categories that represent major content areas in the field of computer science. They represent specific areas of disciplinary importance rather than abstract, general ideas. Each essential concept is supported by various subconcepts that represent specific ideas within each concept. Figure 1 provides a visual representation of the Essential Concepts and the supporting subconcepts.

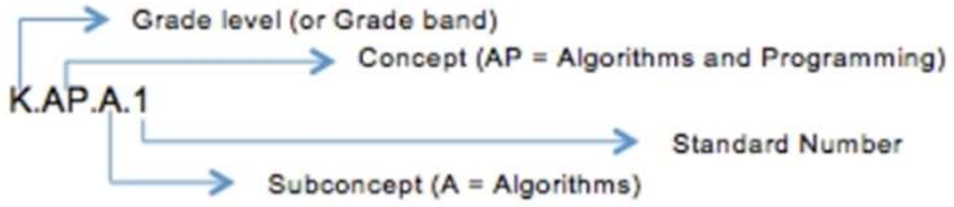
Figure 1: Computer science essential concepts and subconcepts

Computing Systems	Networking and the Internet	Data and Analysis	Algorithms and Programming	Impacts of Computing
Devices	*Network Communication and Organization*	*Collection*	*Algorithms*	*Culture*
Hardware and Software	*Cybersecurity*	*Storage*	*Variables*	*Social Interactions*
Troubleshooting		*Visualization and Transformation*	*Control*	*Safety, Law, and Ethics*
		Inference and Models	*Modularity*	
			Program Development	

The pages following break the concepts and subconcepts down by Concept, from Kindergarten through High School. Each Concept is labeled and separated from the next. This will allow teachers to more easily track progression within the standards.

Each standard will list the grade level, the concept, the subconcept, and the standard number. Figure 2 provides an example of the coding for, and how to read, a standard:

Figure 2: Standard Coding Scheme for Standards


Grade level (or Grade band)
Concept (AP = Algorithms and Programming)
K.AP.A.1
Standard Number
Subconcept (A = Algorithms)

Concept: Computing Systems (CS)

Subconcept: Devices (D)	
K.CS.D.1	**With teacher guidance, select and operate an appropriate device to perform a task.** *People use computing devices to perform a variety of tasks accurately and quickly. With teacher guidance, students should be able to select the appropriate device to use for tasks they are required to complete. For example, if students are asked to draw a picture, they should be able to open and use a drawing app/program to complete this task, or if they are asked to create a presentation, they should be able to open and use presentation software.* Practice(s): Fostering an inclusive Computing Culture: 1.2
Subconcept: Hardware and Software (HS)	
K.CS.HS.1	**Use appropriate terminology in identifying and describing the function of common physical components of computing systems.** *A computing system is composed of hardware and software. Hardware consists of physical components. Students should be able to identify and describe the function of external hardware, such as desktop computers, laptop computers, tablet devices, monitors, keyboards, mice, and printers.* Practice(s): Communicating about Computing: 7.2
Subconcept: Troubleshooting (T)	
K.CS.T.1	**Discuss basic hardware and software problems.** *Problems with computing systems have different causes. Students should be able to communicate a hardware or software problem (e.g., when an app or program is not working as expected, a device will not turn on, the sound does not work, etc.).* Practice(s): Testing and Refining Computational Artifacts, Communicating About Computing: 6.2, 7.3
Subconcept: Devices (D)	
1.CS.D.1	**With teacher guidance, select and operate appropriate devices and software to perform a task.** *People use computing devices to perform a variety of tasks accurately and quickly. With teacher guidance students should be able to select the appropriate app/program to use for tasks they are required to complete. For example, if students are asked to draw a picture, they should be able to open and use a drawing app/program to complete this task.* Practice(s): Fostering an inclusive Computing Culture, Communicating About Computing: 1.2, 7.3
Subconcept: Hardware and Software (HS)	

1.CS.HS.1	**Use appropriate terminology in identifying and describing the function of common physical components of computing systems.** *A computing system is composed of hardware and software. Hardware consists of physical components. Students should be able to identify and describe the function of external hardware, such as desktop computers, laptop computers, tablet devices, monitors, keyboards, mice, and printers. Students should be able to identify software such as: web browsers, games, etc.* Practice(s): Communicating about Computing: 7.2
Subconcept: Troubleshooting (T)	
1.CS.T.1	**Identify basic hardware and software problems using accurate terminology.** *Problems with computing systems have different causes. Students should be able to communicate a hardware or software problem with accurate terminology (e.g., when an app or program is not working as expected, a device will not turn on, the sound does not work, etc.).* Practice(s): Testing and Refining Computational Artifacts, Communicating About Computing: 6.3, 7.2
1.CS.T.2	**With teacher guidance, begin to use basic troubleshooting strategies.** *Students would be able to use simple troubleshooting strategies. For example, turning a device off and on to reboot it, closing and reopening an app, turning on speakers, or plugging in headphones, and then adjusting volume.* Practice(s): Testing and Refining Computational Artifacts: 6.2
Subconcept: Devices (D)	
2.CS.D.1	**Recognize that users have different needs and preferences for technology they used by selecting and operating appropriate devices.** *People use computing devices to perform a variety of tasks accurately and quickly. Students should be able to select the appropriate app/program to use for tasks they are required to complete. For example, if students are asked to draw a picture, they should be able to open and use a drawing app/program to complete this task, or if they are asked to create a presentation, they should be able to open and use presentation software. In addition, with teacher guidance, students should identify and discuss preferences for software with the same primary functionality.* *Practice(s):* Fostering an inclusive Computing Culture, Communicating About Computing: 1.1, 7.3
Subconcept: Hardware and Software (HS)	
2.CS.HS.1	**Understand how computing systems use both hardware (device) and software (program/app) to process information.** *A computing system is composed of hardware and software. Hardware consists of physical components. Students should be able to identify and describe the function of external hardware, such as desktop computers, laptop computers, tablet devices, monitors, keyboards, mice, and printers.*

	Practice(s): Communicating about Computing : 7.2

Subconcept: Troubleshooting (T)	
2.CS.T.1	**Explain basic hardware (device) and software (program/app) problems using accurate terminology.** *Problems with computing systems have different causes. Students at this level do not need to understand those causes, but they should be able to communicate a problem with accurate terminology (e.g., when an app or program is not working as expected, the device is frozen, the link doesn't work, the internet is not connecting,* *Practice(s):* Communicating About Computing: 7.2
2.CS.T.2	**With teacher guidance, use basic troubleshooting strategies.** *Students should be able to use simple troubleshooting strategies, including turning a device off and on to reboot it, closing and reopening an app, turning on speakers, or plugging in headphones and adjusting volume.* *Practice(s):* Testing and Refining Computational Artifacts: 6.2
Subconcept: Devices (D)	
3.CS.D.1	**Identify how internal and external parts of computing devices function to form a system within a single device and hardware that connects to the device to extend capability.** *Keyboard input or a mouse click could cause an action to happen or information to be displayed on a screen; this could only happen because the computer has a processor to evaluate what is happening externally and produce corresponding responses. Students describe how devices and components interact using correct terminology.* *Practice(s): Communicating About Computing, Recognizing and Defining Computational Problems: 7.2, 3.2*
Subconcept: Hardware and Software (HS)	
3.CS.HS.1	**Recognize that hardware (devices) and software (programs/apps) communicate in a special language that the computing system can understand.** *Computing systems convert instructions, such as "print," "save," or "crop," into a special language that the computer can understand. Students discuss the process that happens when hardware communicates with software.* *Practice(s): Communicating About Computing: 7.2*
3.CS.HS.2	**Recognize that hardware (devices) can only accomplish the specific tasks the software (programs/apps) is designed to accomplish.** *Cameras can take pictures because the camera software allows them to do so. Students discuss examples of different hardware and the tasks they can accomplish.* *Practice(s): Communicating About Computing: 7.2*
Subconcept: Troubleshooting (T)	
3.CS.T.1	**Identify and use common troubleshooting strategies to solve simple hardware and software problems.** *Although computing systems may vary, common troubleshooting strategies, such as checking connections and power or swapping a working part in place of a potentially defective part can be used to restore functionality. Restarting a computer (rebooting) is*

	commonly effective because it resets the machine. Computing devices are composed of an interconnected system of hardware and software, troubleshooting strategies may need to address both. *Practice(s): Communicating About Computing: 7.2*
Subconcept: Devices (D)	
4.CS.D.1	**With teacher guidance, model how internal and external parts of computing** *connect multiple devices in a computing system*. *Computing devices may be connected to other devices or components to extend their capabilities, such as sensing and sending information. Connections can take many forms, such as physical or wireless. Together, devices and components form a system of interdependent parts that interact for a common purpose. Students model the process that happens when multiple devices form a system* *Practice(s): Communicating About Computing, Recognizing and Defining Computational Problems, Creating Computational Artifacts: 7.3, 3.1, 5.2*
Subconcept: Hardware and Software (HS)	
4.CS.HS.1	**Recognize that bits serve as the basic unit of data in computing systems and can represent a variety of information.** *Hardware and software communicate in binary digits commonly represented in 0s and 1s. Students discuss how bits are a unit of data.* *Practice(s): Communicating About Computing: 7.2*
4.CS.HS.2	**Recognize that a single piece of hardware can accomplish different tasks depending on** *its* **software.** *A photo filter application (software) works with a camera (hardware) to produce a variety of effects that change the appearance of an image. This image is transmitted and stored as bits, or binary digits, which are commonly represented as 0s and 1s. All information, including instructions, is encoded as bits. Students discuss a variety of software and hardware that work together.* *Practice(s): Practice(s): Communicating About Computing: 7.2*
Subconcept: Troubleshooting (T)	
4.CS.T.1	**Develop** *and apply simple troubleshooting strategies* **to solve simple hardware and software problems.** *Although computing systems may vary, common troubleshooting strategies such as checking connections and power, or swapping a working part in place of a potentially defective part, can be used to restore functionality. Restarting a device (rebooting) is commonly effective because it resets the machine. Because computing devices are composed of an interconnected system of hardware and software, troubleshooting strategies may need to address both.* *Practice(s): Recognizing and Defining Computational Problems, Collaborating Around Computing: 3.1, 2.4*
Subconcept: Devices (D)	
5.CS.D.1	**Analyze and model how internal and external parts of computing devices communicate as a system.***Computing devices often depend on other devices or components. A robot depends on a physically attached light sensor to detect changes in brightness, whereas the light sensor depends on the robot for power. A smartphone can use wirelessly connected headphones to send audio information, and the headphones require a music source.*

	Practice(s): Communicating About Computing, Recognizing and Defining Computational Problems, Creating Computational Artifacts, Testing and Refining Computational Artifacts: 7.2, 3.2, 5.2, 6.3
5.CS.D.2	**Explain how computing devices affect humans in positive and negative ways.** *The use of computing devices has potential consequences, especially with regard to privacy and security.* *Practice(s): Fostering an Inclusive Computing Culture, Communicating About Computing: 1.1, 7.2*
Subconcept: Hardware and Software (HS)	
5.CS.HS.1	**Model how information is transformed into binary digits to be stored or processed.** *Hardware and software communicate in binary digits commonly represented in 0s and 1s. Information is transformed into binary digits, for example a song is stored as more binary digits than a photo.* *Practice(s): Communicating About Computing, Creating Computational Artifacts: 7.2, 5.2*
5.CS.HS.2	**Demonstrate and explain how hardware can accomplish different tasks depending on the software.** *In order for a person to accomplish tasks with a computer, both hardware and software are needed. At this stage, a model should only include the basic elements of a computer system, such as input, output, processor, sensors, and storage. Students could draw a model on paper or in a drawing program, program a animation to demonstrate it, or demonstrate it by acting it out in some way.* *Practice(s): Communicating About Computing, Creating Computational Artifacts: 7.2, 5.3*
Subconcept: Troubleshooting (T)	
5.CS.T.1	**Apply potential solutions and solve simple hardware and software problems using common troubleshooting strategies.** *Although computing systems may vary, common troubleshooting strategies such as checking connections in power or swapping a working part in place of a potentially defective part can be used to restore functionality. Restarting a device (rebooting) is commonly effective because it resets the computer machine. Computing devices are composed of an interconnected system of hardware and software, troubleshooting strategies may need to address both. In fifth grade students begin troubleshooting complex problems through networks, routers, and switches.* *Practice(s): Recognizing and Defining Computational Problems, Developing and Using Abstractions: 3.2, 4.1*
Subconcept: Devices (D)	
6.CS.D.1	**Compare computing device designs based on how humans interact with them.** *The study of human–computer interaction (HCI) can improve the design of devices, including both hardware and software. Teachers can guide students to consider usability through several lenses. For example, teachers can have students compare computing devices that have different methods of human interaction (keyboard/mouse/trackpad, touchscreen, voice commands, facial recognition/fingerprint sensing, etc)* *Practice(s): Recognizing and Defining Computational Problems: 3.3*

Subconcept: Hardware and Software (HS)	
6.CS.HS.1	**Explain how hardware and software can be used to collect and exchange 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.* Practice(s): Creating Computational Artifacts: 5.1
Subconcept: Troubleshooting (T)	
6.CS.T.1	**Identify problems that can occur in computing devices and their components within a system.** *Since a computing device may interact with interconnected devices within a system, problems may not be due to the computing device itself but to devices or components connected to it. For example, students can discuss why the internet might not be working on their device. It could be airplane mode, no signal (wifi or mobile data), component malfunction, interference, etc.* Practice(s): Testing and Refining Computational Artifacts: 6.2
Subconcept: Devices (D)	
7.CS.D.1	**Identify some advantages, disadvantages, and consequences with the design of computer devices based on an analysis of how users interact with devices.** *The study of human–computer interaction (HCI) can improve the design of devices, including both hardware and software. Teachers can guide students to consider usability through several lenses, including accessibility, ergonomics, and learnability. For example, assistive devices provide capabilities such as scanning written information and converting it to speech.* *Practice(s): Recognizing and Defining Computational Problems: 3.3*
Subconcept: Hardware and Software (HS)	
7.CS.HS.1	**Design projects that combine hardware and software to collect and exchange data.** *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, Global Position System (GPS), microphone, fingerprint sensor, etc.* Practice(s): Creating Computational Artifacts: 5.1

Subconcept: Troubleshooting (T)	
7.CS.T.1	**Evaluate strategies to fix problems with computing devices and their components within a system.** *Since a computing device may interact with interconnected devices within a system, problems may not be due to the computing device itself but to devices or components connected to it. For example, troubleshooting strategies include following a troubleshooting flow diagram, making changes to software to see if hardware will work, checking connections and settings, and swapping in working components.* Practice(s): Testing and Refining Computational Artifacts: 6.2
Subconcept: Devices (D)	
8.CS.D.1	**Improve the design of computing devices based on an analysis of how users interact them, and consider unintended consequences.** *The study of human–computer interaction (HCI) can improve the design of devices, including both hardware and software. Students should make recommendations for existing devices (e.g., a laptop, phone, or tablet) or design their own components or interface (e.g., create their own controllers). Teachers can guide students to consider usability through several lenses, including accessibility, ergonomics, and learnability. For example, assistive devices provide capabilities such as scanning written information and converting it to speech.* *Practice(s): Recognizing and Defining Computational Problems: 3.3*
Subconcept: Hardware and Software (HS)	
8.CS.HS.1	**Design and evaluate projects that combine hardware and software components to collect and exchange data.** *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): Creating Computational Artifacts: 5.1
Subconcept: Troubleshooting (T)	
8.CS.T.1	**Systematically identify and develop strategies to fix problems with computing devices and their components.** *Since a computing device may interact with interconnected devices within a system, problems may not be due to the computing device itself but to devices or components connected to it. Just as pilots use checklists to troubleshoot problems with aircraft systems, students should use a similar, structured process to troubleshoot problems with computing systems and ensure that potential solutions are not overlooked. Examples of troubleshooting strategies include following a troubleshooting flow diagram, making changes to software to see if hardware will work, checking connections and settings, and swapping in working components.* Practice(s): Testing and Refining Computational Artifacts: 6.2

Subconcept: Devices (D)	
HS.CS.D.1	**Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.** *Computing devices are often integrated with other systems, including biological, mechanical, and social systems. Students could explore how a medical device can be embedded inside a person to monitor and regulate his or her health, a hearing aid (a type of assistive device) can filter out certain frequencies and magnify others, a monitoring device installed in a motor vehicle can track a person's driving patterns and habits, or a facial recognition device can be integrated into a security system to identify a person. The usability, dependability, security, and accessibility of these devices, and the systems with which they are integrated are important considerations in their evolving design. Students are not expected to create integrated or embedded systems at this level.* *Practice(s):* Developing and Using Abstractions: 4.1
Subconcept: Hardware and Software (HS)	
HS.CS. HS.1	**Describe levels of abstraction and interactions between application software, system software, and hardware layers.** *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.* *Practice(s):* Developing and Using Abstractions: 4.1
Subconcept: Troubleshooting (T)	
HS.CS.T.1	**Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.** *Troubleshooting complex problems involves the use of multiple sources when researching, evaluating, and implementing potential solutions. Troubleshooting also relies on experience, such as when people recognize that a problem is similar to one they have seen before or adapt solutions that have worked in the past. Examples of complex troubleshooting strategies include resolving connectivity problems, adjusting system configurations and settings, ensuring hardware and software compatibility, and transferring data from one device to another. Students could create a flow chart, a job aid for a help desk employee, or an expert system.* *Practice(s):* Testing and Refining Computational Artifacts: 6.2