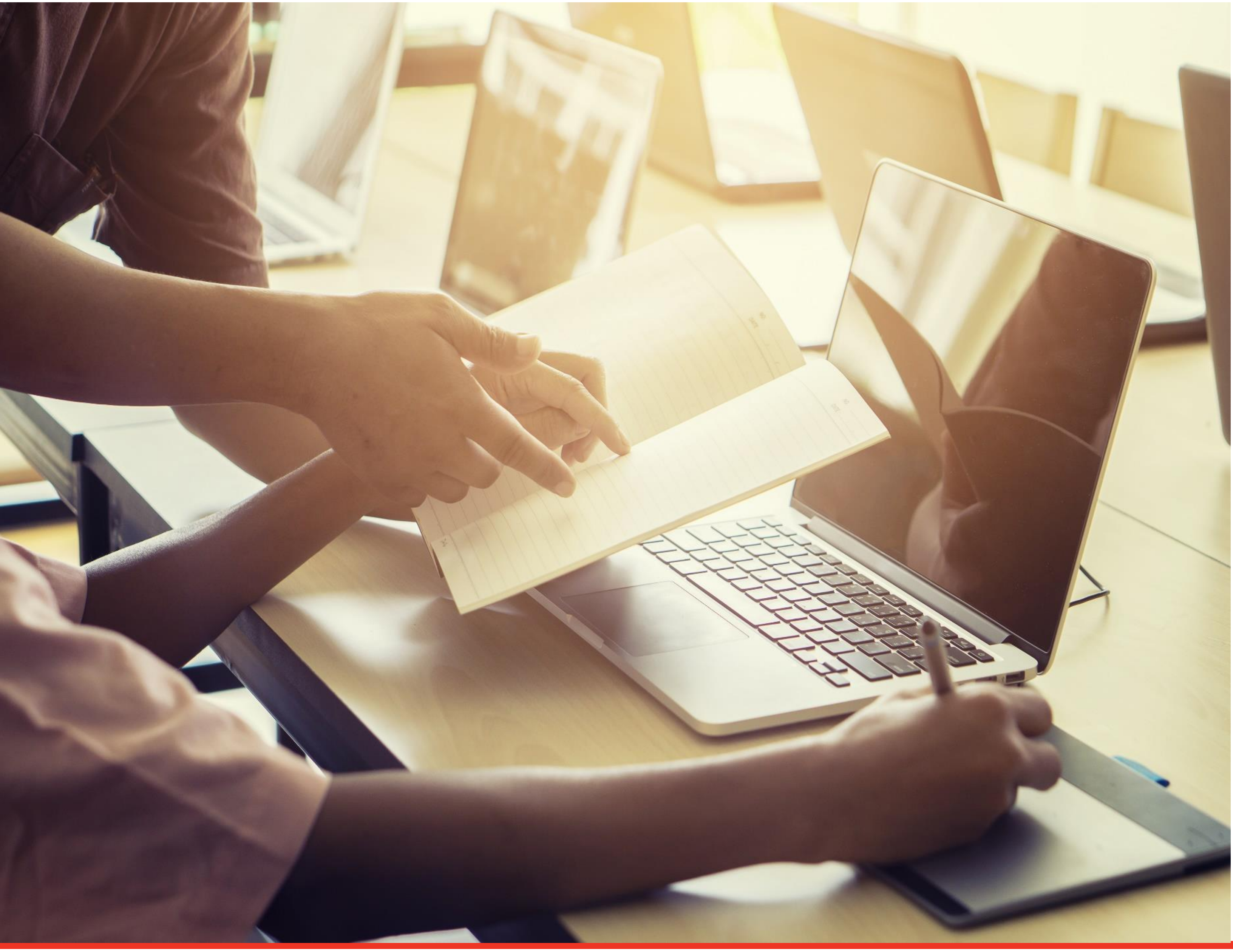

Computer Science Standards

Implementation Guidebook

Tennessee Department of Education | June 2024

Contents

Introduction

The Computer Science Standards Implementation Guidebook serves as a curriculum support document for the Tennessee K-12 Computer Science State Standards. The purpose of this Guidebook is to provide resources that will support computer science standards deconstruction, ensuring standards-based alignment of computer science lessons and assessments. Resources in this guidebook include standard explanations, connections to Core Concepts and Sub-concepts, aligned vocabulary, K-12 Core Concept Progressions, and a systematic process for coaching the standards deconstruction process.

The online version of the Computer Science Standards Implementation Guidebook, as well as resources and professional learning that accompany this guidebook, can be found on our website: **https://www.computersciencetn.org/resources/**.

Part A: Introduction to Tennessee Computer Science Law

Public Chapter #979 of the Public Acts of 2022

In 2022, **Governor Bill Lee** joined with the National Governors Association that includes fifty-five states and territories to sign a compact agreeing to act and expand computer science opportunities for students in their states. This initiative and compact demonstrated growing momentum across the country and Tennessee for computer science education.[1]

The **Tennessee General Assembly** in **May 2022** passed by unanimous vote in both chambers **Chapter 979 of the Public Acts of 2022** (PC979), now codified in Tenn. Code Annotated Title 49, requiring specific actions be taken to ensure all students are fully prepared for the technological jobs of today and of the future. Concurrent with the passing of **Chapter 979**, the state of Tennessee joined **18** other states across the country allocating funding for computer science education.

Chapter 979 of the Public Acts of 2022 requires that high school students receive one full school year of computer science education to satisfy graduation requirements by the 2024-2025 school year. Middle school students are to receive one course in computer science and elementary school students will receive grade-appropriate computer science education by integrating computer science standards into content.

These requirements will increase the availability of computer science courses and curriculum enabling more Tennessee students to benefit from enrolling in computer science education. The **2022 State of Computer Science Education Report** indicates that **75.5% of Tennessee high school students** attend a school that offers foundational computer science, but only **4.3% of students** are enrolled in a computer science course. Of those students enrolled in a computer science course, **28.7% are female**. With the passing of this law, Tennessee is now one of five states requiring students take a computer science course to graduate high school.[1]

The **Code.org Advocacy Coalition** developed nine policy recommendations to make computer science a fundamental part of the state education system. These nine policies contribute to building and sustaining a comprehensive state policy framework that expands the teaching and learning of computer science.[1]

They include:

1. Creating a state plan for K-12 computer science.
2. Defining computer science and establishing rigorous K-12 computer science standards.
3. Allocating funding for computer science teacher professional learning.
4. Implementing clear certification pathways for computer science teachers.
5. Creating preservice programs in computer science at higher education institutions.
6. Establishing computer science supervisor positions in education agencies.
7. Requiring that all high schools offer computer science.
8. Allowing a computer science credit to satisfy a core graduation requirement.
9. Allowing computer science to satisfy a higher education admission requirement.

Following the passing of **Chapter 979 of the Public Acts of 2022**, Tennessee meets **eight of these nine policies** and joins 26 other states with similar policies for computer science education.

Active Response to the Tennessee Law

The **Tennessee Department of Education (TDOE)** will support all **Local Education Agencies (LEA)** by providing:

- a no-cost Computer Science Endorsement Pathway (CSEP),
- grade-band specific computer science courses, standards, resources and materials, and an online course for middle and high schools,
- targeted professional development modules,
- a computer science education network.

[1] Code.org 2022 State of Computer Science Education: Understanding Our National Imperative

Part B: State Goals

School Year 2024-2025 TN K-12 Computer Science Goals

When new or revised state standards are adopted, measures are put in place to ensure successful standard implementation and sustainability. The Tennessee Department of Education (TDOE) in partnership with the Tennessee STEM Innovation Network (TSIN) have developed three goals for teachers to keep in mind during the 2024-2025 school year when computer science standards have mandatory implementation. Focusing on three goals during the first year of implementation will take what seems overwhelming and decompose it into small achievable chunks that will guarantee sustainability over time.

As an educator in the state of Tennessee, your first-year standard implementation goals should focus on developing an understanding of the standards and their components, deconstructing standards, and using this knowledge to engage all students in standards-based instruction and assessment.

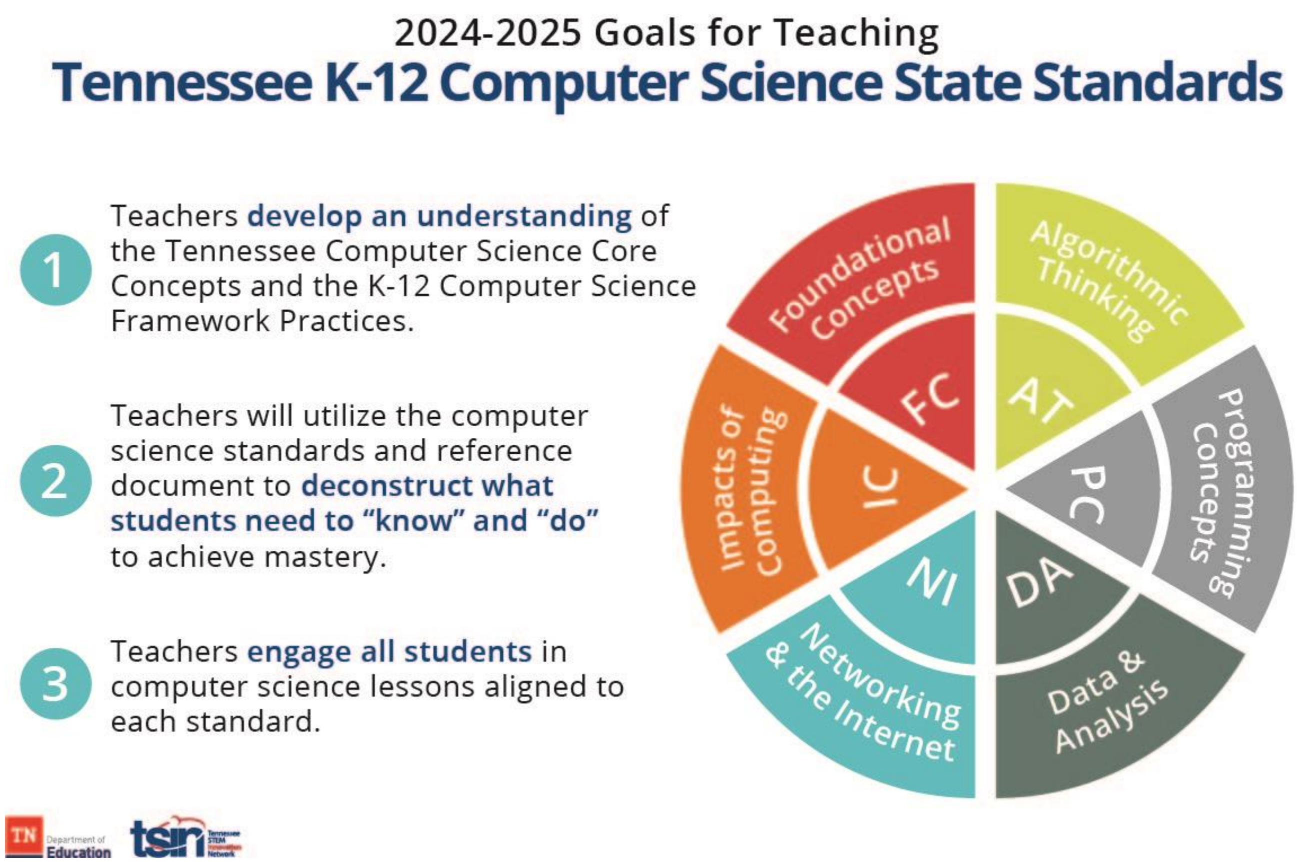

Figure 1. 2024-2025 Tennessee State Goals for Teaching K-12 Computer Science State Standards

Part C: Tennessee K-12 Computer Science State Standards

The **Tennessee Department of Education (TDOE)** in partnership with **the Tennessee STEM Innovation Network (TSIN)**, selected a diverse team of educators in summer 2022 to write the new **Tennessee K-12 Computer Science State Standards**. The driving thought behind these standards is that computer science should become a foundational part of Tennessee K-12 education, accessible to all, rather than a vocational part of education only for those headed to technology-based employment.

The first reading of the **Tennessee K-12 Computer Science State Standards** went before **the Tennessee State Board of Education (SBE)** in **October 2022** and was passed. Over the next six weeks feedback for the standards was solicited and received by TSIN and final revisions were made. The Tennessee K-12 Computer Science State Standards were approved February 10, 2023, with a mandatory implementation date of the **2024-2025** school year.

Computer Science Core Concepts

The **Tennessee K-12 Computer Science State Standards** contain six core concepts, represented in *Figure 2*. Standards are created in themes and terms to seamlessly integrate into instruction and cognitive tasks for every grade level and subject area so students can see logical problem-solving, allowing for enhancement of any topic. They are depicted in equal portions because they each support the other. Therefore, by the time a student completes high school in Tennessee, they will have opportunities to explore, experience, and demonstrate mastery in each of these six core concept areas and to develop the skills and practices each represents.[2]

Computer Science Sub-Concepts

Each of the six Core Concepts contains 3-4 Sub-Concepts. These Sub-Concepts are components of the main concept and decompose (break-down) complex ideas into more manageable parts. Furthermore, these Sub-Concepts help with the organization and understanding of each concept as it progresses from Kindergarten through 12th grade.

[2] Tennessee Department of Education. (2022, October 28). Computer Science Tennessee K-12 Computer Science State Standards. Retrieved January 5, 2023, from https://www.tn.gov/content/dam/tn/stateboardofeducation/documents/2022-sbe-meetings/october-28%2c-2022-sbe-meeting/10-28-22%20III%20A%20Computer%20Science%20Standards%20Framework%20for%20Grades%20K-12%20Clean.pdf

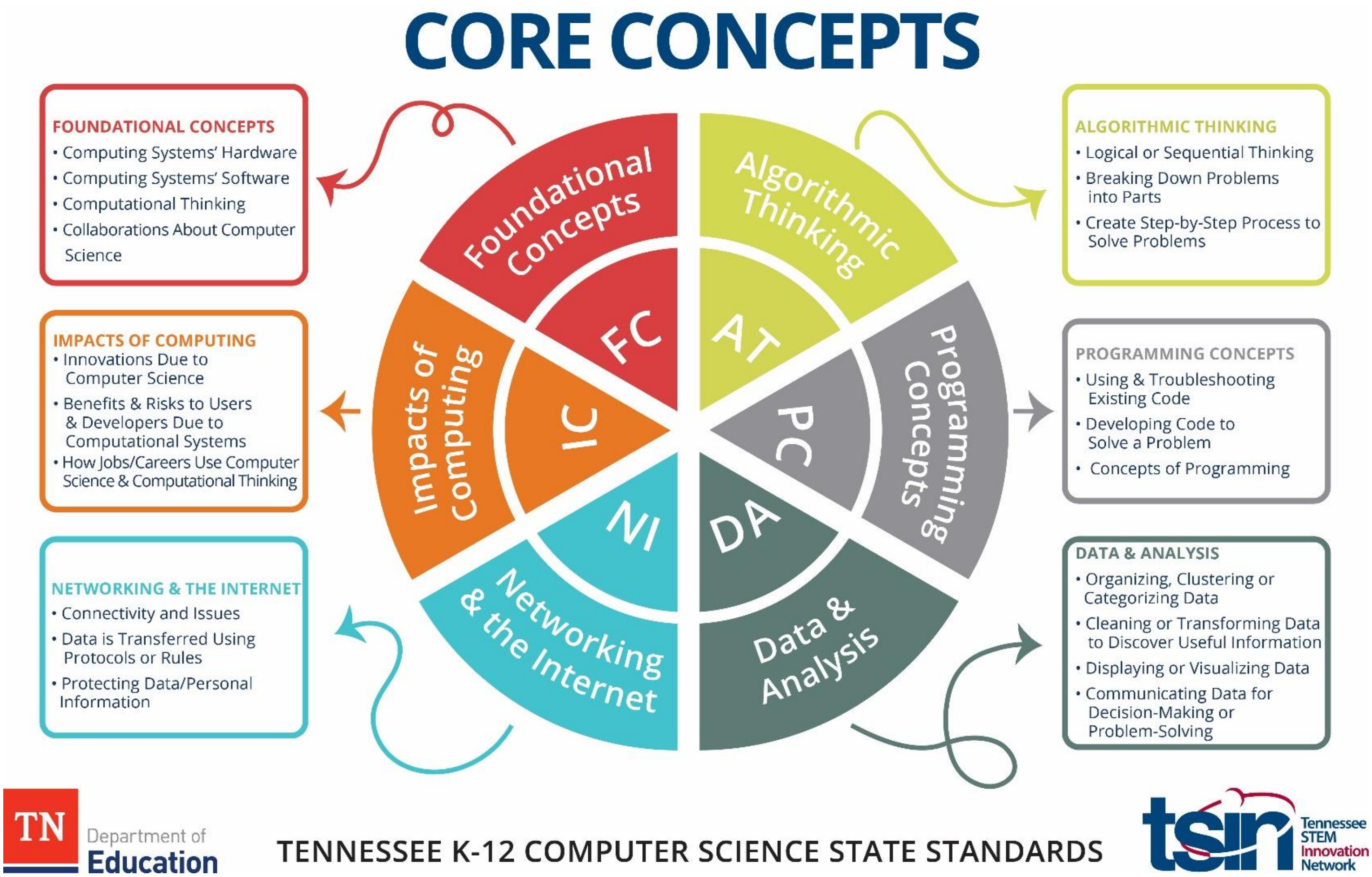

Figure 2. Six Core Concepts and Sub-Concepts of Tennessee's CS K-12 Standards

Curriculum Support Documents

The following Computer Science Curriculum Support Documents are organized by Core Concept and are intended to support teachers when deconstructing the new computer science standards. Each section begins with the **Core Concept Overview**, followed by the **Core Concept Progression Statements**, and finally the **Computer Science Standards Reference Document**.

Each **Core Concept Overview** defines the core concept and gives understanding to why standards may have been classified within that concept. The Core Concept Overview also presents themes that demonstrate how the Sub-Concepts were identified.

The first table after each Core Concept Overview showcases the **Core Concepts Progression Statements**. These statements summarize what students will be mastering for the concept as they progress through grade bands: Early Elementary (K-2), Upper Elementary (3-5), Middle School (6-8), and High School (9-12). It is important to understand the progressions and the role of each grade band within them.

Finally, the **Computer Science Standards Reference Document** is listed by Core Concept and provides an explanation for each standard, connecting the Core Concept and Sub-Concept, and linking the standard to computer science vocabulary. These documents play a significant role in deconstructing the standards to determine what students need to "know" and "do" to achieve mastery of each standard.

To view the Computer Science Standards Reference Document by grade level instead of core concept, please visit **www.computersciencetn.org/resources/**.

Foundational Concepts

Foundational Concepts focus on understanding computing systems (hardware and software). This core concept also includes computational thinking and its applications and emphasizes the importance of collaboration for computer science. Hardware includes the physical components of computing systems while software consists of the programs and data necessary to operate and execute specific tasks using these systems. An understanding of hardware and software and their interactions is a foundational concept and students should learn how systems use both to represent and process information. Computational thinking is a thought process that revolves around solving a variety of problems. In this Implementation Guidebook, we will focus on **four** main components of computational thinking: **1) algorithms**, **2) pattern recognition**, **3) decomposition**, and **4) abstraction**.

Collaborating Around Computing is the *K-12 Computer Science Framework's* Practice #2 and collaboration is one of the 4Cs in the 21st century skills. We should empower students by incorporating collaboration throughout their exploration of computer science concepts and practices to prepare them for their future workspace. These foundational building blocks set the stage for the remaining five concepts including algorithmic thinking, programming concepts, data and analysis, networking and the internet and impacts of computing.

Foundational Concepts

Sub-Concepts: Grade Bands	Early Elementary (K-2)	Upper Elementary (3-5)	Middle (6-8)	High (9-12)
Computing Systems Hardware	Students identify and apply the functions of common physical components of hardware (keyboard, mouse, monitor, etc.) and differentiate between a variety of devices (iPad, desktop, laptop, etc.)	Students perform grade-appropriate tasks with proficiency using a range of digital devices. Students recognize how software and hardware interact with one another.	Students systematically analyze and fix problems that occur with computing devices and their components. Students recognize the advantages and limitations of computer hardware.	

Foundational Concepts

Sub-Concepts: Grade Bands	Early Elementary (K-2)	Upper Elementary (3-5)	Middle (6-8)	High (9-12)
Computing Systems Software	Students identify and apply basic functions to navigate software.	Students locate and utilize grade-appropriate software. Students model proficient use of advanced online tools to communicate key ideas.	Students identify advantages and limitations of software on computing devices in order to improve user experience.	
Computational Thinking	Students use computational thinking to ask questions, conduct investigations, solve problems, and test solutions with teacher guidance.	Students use computational thinking to ask questions, conduct investigations, solve problems, and test solutions independently.	Students use computational thinking to ask questions, conduct investigations, solve problems, and test solutions across multiple disciplines.	

Foundational Concepts					
	Grade Bands	**Early Elementary (K-2)**	**Upper Elementary (3-5)**	**Middle (6-8)**	**High (9-12)**
	Collaboration About Computer Science	Students work collaboratively to connect with other learners using digital tools.	Students use digital tools to communicate key ideas and details collaboratively to inform, persuade, and/or entertain.	Students understand how collaboration is essential to computer science and apply collaborative skills to develop computational solutions.	

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
K.FC.1	Locate letters and numbers on the keyboard.	Hardware is a physical part(s) of a computer. A keyboard is an example of hardware and allows the user to input information. This skill can be introduced through an unplugged activity by printing out a physical copy of a keyboard to familiarize students with the location of keys. Unplugged activities can help students build connections and understand the location of letters and numbers on the keyboard.	Foundational Concepts	Computing Systems Hardware	computer, hardware, keyboard, unplugged activity
K.FC.2	Ask questions to conduct investigations, solve problems, and test solutions.	Computational thinking is a valuable tool for students to approach open-ended problems and create solutions. It involves four key components: algorithmic thinking, pattern recognition, decomposition, and abstraction. Students can work alone or in groups to utilize these strategies to solve issues and test solutions. For instance, if a student is having trouble hearing sound on their device, they can use computational thinking to break down the problem and concentrate on relevant information (such as checking the sound icon or verifying if headphones are plugged in). By investigating and testing possible solutions, they can determine if the volume on their device needs to be adjusted, if they need to disconnect their headset, or if a more significant problem exists.	Foundational Concepts	Computational Thinking	computational thinking

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
1.FC.1	Navigate to applications and documents by using desktop icons, windows, and menus (e.g., open and close the browser window, find/ use bookmark to store the website, recognize and use app on tablet).	Understanding how hardware and software work together is crucial for students to perform tasks accurately and efficiently based on their preferences. Proficiency in using input devices like a mouse or keyboard, and understanding the difference between an icon, window and menu is essential for effectively navigating through software. To practice this, students can select an internet browser icon on their desktop, open the browser, and bookmark a website so they can easily access documents or applications on the website.	Foundational Concepts	Computing Systems Hardware Computing Systems Software	software, application, icon, window, menu
1.FC.2	Demonstrate use of input devices (e.g., mouse, keyboard).	Input refers to signals, data values, or instructions fed into a computer. For instance, a keyboard is used to input text, making it an input device. In addition, microphones are used to input audio or sound. Teachers can leverage input devices in the classroom, such as asking students to log in using a keyboard or dragging and dropping images with a mouse.	Foundational Concepts	Computing Systems Hardware	input, input device, hardware
2.FC.1	Use the menu and toolbar to navigate editing functions.	Applications are used on various computing devices to perform tasks accurately and quickly based on user needs and preferences. Students could use the menu and toolbar to navigate editing functions in various applications (Word, Google Docs, Slides, PowerPoint, etc.). For example, students could add two different images of a bat (the baseball bat and the mammal bat), select a font and size to type bat and type the two meanings of bat, and use the highlight command to make the two meanings of bat a different color.	Foundational Concepts	Computing Systems Software	application, menu, toolbar

Early Elementary (K-2)					
Identifier	Standard	Explanation	Concept	Sub-Concept	Vocabulary
2.FC.2	Use a variety of digital tools collaboratively to connect with other learners.	Digital tools allow students to connect and exchange ideas. By practicing good digital citizenship, students can share their work and receive real-time feedback. For example, students could collaborate on designing a rural or urban community using PowerPoint, Padlet, blogs, Google Docs, or Canva.	Foundational Concepts	Collaboration About Computer Science	digital citizen, digital tool
2.FC.3	Ask questions to conduct investigations, solve problems, and test solutions.	Computational thinking is a valuable tool for students to approach open-ended problems and create solutions. It involves four key components: algorithmic thinking, pattern recognition, decomposition, and abstraction. Students can work alone or in groups to utilize these strategies in order to solve issues and test solutions. For instance, if a student's screen is not responding to information input from the mouse or keyboard, they can use computational thinking to break down the problem and concentrate on relevant information (such as checking to see if the device needs an update or determining whether or not they are connected to the internet). By investigating and testing possible solutions, they can determine if restarting the computer to install updates can help or if the device needs to connect to a network for the device to respond.	Foundational Concepts	Computational Thinking	computational thinking, troubleshooting

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
2.FC.4	Select technology or tools to solve a problem or design a solution.	Using various digital tools, students can determine the appropriateness of a device or software for a given task based on its features. For example, which devices or applications might be ideal for drawing, capturing video or audio, and editing media? Students can understand using different devices and apps are needed to complete individual tasks.	Foundational Concepts	Computational Thinking	software, digital tool

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
3.FC.1	Locate and use appropriate online tools and resources to explore, research, and collect data on specific topics (e.g., applications, web browsers, and online tutorials).	As students explore online with teacher guidance, they should be able to use beginning skills, such as using specific keywords in the search engine, identifying reliable resources using domain names (.com, .net, .edu, etc.), locate appropriate tools and resources for research and collect data on specific topics (e.g., applications, web browsers, and online tutorials). As students progress, they can independently select relevant resources to gather information, support a claim, or communicate information on an advanced level.	Foundational Concepts	Computing Systems Software	domain names, search engine, keyword search
3.FC.2	Communicate key ideas and details collaboratively in a way that informs, persuades, and/or entertains, using digital tools.	Collaborative computing is the process of performing a computational task by working in pairs or on teams and involves asking for the contributions and feedback of others. Effective collaboration can lead to better outcomes than working independently. Students could develop computational artifacts and elicit feedback from teachers and peers that inform, persuade,	Foundational Concepts	Collaboration About Computer Science	digital tools, computational artifact

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
		and/or entertain using digital tools like Google Docs, PPT, Slides, digital animations, storyboards, or other apps.			
3.FC.3	Use basic features of digital tools to communicate key ideas and details in a way that informs and/or persuades.	Students should develop computational artifacts that inform and/or persuade using creative expression and exploration of ideas to create and solve computational problems. Students should understand and use the basic features of various hardware and software. For example, students using Google Docs, may highlight text and select various options to enhance their text (bold, underline, font, size, color).	Foundational Concepts	Computing Systems Hardware Computing Systems Software	software, hardware, computational artifact
4.FC.1	Demonstrate an appropriate level of proficiency in performing tasks using a range of digital devices.	Once a digital device has been appropriately selected, students will use the device to perform a variety of tasks. For example, students will open the correct software (Word), utilizing the appropriate hardware (mouse, keyboard, monitor), to compose text, name the file, and save it in a correct folder. This standard shows how software and hardware complement one another.	Foundational Concepts	Computing Systems Hardware Computing Systems Software	hardware, software
4.FC.2	Use age-appropriate online tools and resources (e.g., learning management systems, grade and assignment record, tutorial, assessment, web browser).	Students can access materials through age-appropriate online tools, web browsers, and resources. Skills include accessing and submitting work online, checking grades, using a student LMS (learning management system) on a dashboard, utilizing tutorials to seek help with programs, communicating with teachers and peers, monitoring their work, using online assessments, and seeking technical support when troubleshooting.	Foundational Concepts	Computing Systems Software	online, Learning Management System

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
4.FC.3	Create a simple digital model of a system and explain what the model shows and does not show.	A model is an artificial representation of an actual set of events or processes. Students could create a digital model that shows information that represents a system. For example, students could use the program "Scratch" to model the phases of the moon. Students would need to apply the concept of abstraction to determine the critical parts of the model. When utilizing digital models, it is important to give students at this age an opportunity to evaluate and critique the digital model.	Foundational Concepts	Computational Thinking	digital model, abstraction
5.FC.1	Use advanced features of digital tools and media-rich resources to communicate key ideas and details in a way that informs, persuades, and/or entertains.	Integrating digital tools into education can improve students' ability to convey key concepts in an engaging, informative, or persuasive way. Tools like PowerPoint provide a range of slide transitions, animations, and design layouts that students can use to communicate important information effectively. Students can share their work with others by saving and uploading their artifacts to a learning management system, sending it as an attachment, or create a shared URL link. Additionally, other digital tools such as Google Docs, video apps, stop motion animation, and storyboards can be used to create media-rich resources that allow students to communicate effectively.	Foundational Concepts	Computing Systems Software Computing Systems Hardware	media-rich, digital tool, computational artifact

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.FC.1	Analyze the advantages and limitations of existing computing devices to improve user experience.	Understanding the advantages and limitations of an existing computing device is important to predict the user experience. Students should consider factors that affect user experience such as processing power, memory, storage capacity, screen size, battery life, connectivity options, input/output capabilities, and user interface. For example, assistive devices provide capabilities such as scanning written information and converting it to speech.	Foundational Concepts	Computing Systems Software Computing Systems Hardware	computing device
MS.FC.2	Demonstrate skills in identifying and solving hardware and software problems that can occur during regular usage.	Although computing systems can differ, common troubleshooting methods apply to all of them. Students need to be able to solve issues like unresponsive devices, power problems, network connectivity issues, app crashes, audio problems, or password errors. Allowing students to develop an algorithm (step-by-step process) for troubleshooting can help them solve a variety of problems. They can use strategies like restarting the device, checking power, ensuring network access, restarting apps, confirming speaker or headphone connections, and verifying the caps lock status to keep computing systems running smoothly.	Foundational Concepts	Computing Systems Hardware Computing Systems Software	algorithm, computing system
MS.FC.3	Apply computational thinking to a variety of problems across multiple disciplines.	Computational thinking is a thought process that can be used to solve a variety of problems. Four main components of computational thinking include algorithmic thinking, pattern recognition, decomposition, and abstraction. For example, in music class students can write an algorithm to sort music into playlists using conditional statements.	Foundational Concepts	Computational Thinking	computer science, computational thinking, abstraction, decomposition, pattern recognition, algorithm

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.FC.4	Understand how collaboration is essential to computer science and apply collaborative skills to develop computational solutions.	Computer science depends on collaboration between team members as it brings different perspectives to solving a common problem. Thus, learning to work as a team is essential. Students should be able to assume roles within teams and manage project workflow using structured timelines. For example, projects could be divided using the design stage of a game and decomposing it into planning the storyboard, a flowchart, and different parts of the game mechanics. The teams then assign tasks and roles among team members, set goals, and assign deadlines. Another example is pair-programming which allows one person to be the driver and one person to be the navigator when coding a program.	Foundational Concepts	Collaboration About Computer Science	pair-programming

Algorithmic Thinking

An **algorithm** is a step-by-step process to complete a task, and algorithms can be executed by both humans and computers. To create an algorithm, students must first decide how to break down the problem or task into manageable parts. After using **decomposition**, students can create logical and sequential steps to solve the task. Flowcharts, conditionals, or visual aids may be utilized to assist in designing and describing the algorithm. As students apply algorithmic thinking to a variety of problems in their K-12 career, they will further develop the conceptual and problem-solving skills necessary to design more efficient and effective algorithms, including sequencing, selection, and iteration.

Algorithmic Thinking

Sub-Concepts	Grade Bands	Early Elementary (K-2)	Upper Elementary (3-5)	Middle (6-8)	High (9-12)
	Logical or Sequential Thinking	Students will classify, sort and categorize events and/or items in a logical order with or without a computing device.	Students will use items or events to predict the outcomes of given algorithms and create ciphers.	Students will analyze algorithmic processes and automation to describe how the use of technology enhances performance and increases efficiency.	Students will create artifacts to collect feedback and demonstrate their knowledge of using algorithms to solve computational problems.
	Breaking Down Problems Into Parts	Students will evaluate problem-solving strategies and communicate effective solutions to a problem.	Students will identify and evaluate problems, pose questions for investigation, and communicate processes to an audience. Students will decompose an existing algorithm to identify areas for improvement.	Students will evaluate and identify the different components of algorithmic processes.	Students will problem solve using effective communication that demonstrates their in-depth knowledge of computer science vocabulary.
	Create Step-by-Step Processes to Solve Problems	Students will brainstorm and organize information to create a visual representation of concepts and elements.	Students will work collaboratively to plan or create an algorithm using a flowchart as well as identifying and eliminating errors through debugging.	Students will utilize variables and data types to create efficient algorithms.	Students will design and develop programs using general computational problems while incorporating feedback.

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
K.AT.1	Construct sequential events step-by-step in a logical order.	Constructing sequential events in a logical order is fundamental to designing and implementing effective algorithms. Students should understand the steps that are necessary to complete a task as well as determine the most effective order in which to complete those steps. For example, when teaching young children, the order of the steps for tying their shoes, students can identify the steps and then create a sequence for what order those steps can be completed.	Algorithmic Thinking	Logical or Sequential Thinking	sequence, algorithm
1.AT.1	Identify and revise problem-solving strategies to solve a simple problem.	Problem-solving refers to the ability to analyze a problem, determine an appropriate strategy, and make adjustments if needed to achieve a solution. It involves a flexible approach to problem-solving, where individuals can recognize when a strategy is not working and revise it accordingly. For example, building off K.AT.1, a student might identify the most efficient way to tie their shoes.	Algorithmic Thinking	Breaking Down Problems Into Parts	strategy
1.AT.2	Classify and sort information into logical order with and/or without a computer.	Organizing and arranging information systematically involves categorizing data based on specific criteria and arranging it in a logical order. Students should practice the skills required to classify and sort information efficiently, both manually and using computer tools or software. For example, provide students with cards that illustrate the steps for brushing your teeth and have them organize the information in order from first to last or last to first order as an unplugged activity.	Algorithmic Thinking	Logical or Sequential Thinking	classify, categorize, data, unplugged activity

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
1.AT.3	Utilize digital tools to illustrate potential solutions to a problem.	Leveraging technology to visualize and present ideas or concepts in a digital format focuses on the ability to use digital tools or software to showcase and explore different possible solutions to a problem. This standard emphasizes the use of digital tools as a creative and effective means of problem-solving and communication. For example, students might complete a drag and drop activity where they dress a person in appropriate clothes for the weather. If it is raining outside, they will drag and drop rainboots and an umbrella.	Algorithmic Thinking	Breaking Down Problems Into Parts	digital
2.AT.1	Plan and create a design document to illustrate thoughts, ideas, and stories in a sequential (step- by-step) manner (e.g., story map, storyboard, sequential graphic organizer).	It is important to teach students how to organize information or narratives in a clear, step-by-step manner to effectively communicate with others. Design documents, such as story maps, storyboards, or sequential graphic organizers, can be used to plan and create visual representations of ideas or stories. For example, students could outline their daily activities and create a graphic timeline representing those events.	Algorithmic Thinking	Create Step-by-Step Processes to Solve Problems	sequence

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
2.AT.2	Compare and evaluate multiple ways to get a solution.	Comparing and evaluating multiple solution paths involves considering a variety of options, identifying their strengths and weaknesses, and making informed decisions based on the evaluation. Students can practice critical thinking and the ability to assess the efficiency of various solutions. For example, divide the students into small groups and assign each group a different strategy to solve the same problem. Provide them with the necessary materials or resources for each strategy. The students will come back together to discuss the strengths and weaknesses of each strategy and determine which would be the best solution.	Algorithmic Thinking	Breaking Down Problems Into Parts	algorithm, efficiency
2.AT.3	Categorize a group of items based on the attributes or actions of each item, with or without a computing device.	Organizing and categorizing information systematically, whether manually or using a computing device, emphasizes skills which are foundational in computer science and information gathering. Students may be asked to create the categories that items can be sorted into. For example, students might sort shapes based on different attributes. First, they might sort by number of sides. Next, they might sort by size or color as an unplugged activity.	Algorithmic Thinking	Logical or Sequential Thinking	attributes, unplugged activity

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
3.AT.1	Discuss the design process and use digital tools to illustrate potential solutions.	Students must understand and engage in the design process which involves exploring the iterative nature of design, discussing problem-solving strategies, and utilizing technology to create visual representations of ideas or solutions. For example, students might be tasked with designing a new and improved layout for the classroom and must use a drawing program, like Google Draw, to showcase their plan for rearranging classroom furniture in a way that would improve classroom flow and efficiency.	Algorithmic Thinking	Logical or Sequential Thinking	iterative, digital
3.AT.2	Create an algorithm to solve a problem as a collaborative team.	An algorithm is a well-defined, step-by-step procedure for accomplishing a specific task. Students working together as a team to design and implement an algorithm to solve a problem emphasizes collaboration, critical thinking, and effective communication within the context of problem-solving in computer science. For example, expanding on K.AT.1 and 1.AT.1, the teacher introduces team collaboration and communication as part of the process for creating an effective step-by-step procedure.	Algorithmic Thinking	Create Step-by-Step Processes to Solve Problems	algorithm
3.AT.3	Identify problems to solve and generate questions for investigation.	In the design process, the first step is to recognize and articulate problems or challenges as a starting point for exploration and discovery. From there, they will formulate questions to investigate. Through this process, students are being introduced to the foundational concept of decomposition. For example, students might identify a challenge in their school that they want to improve and begin developing questions for how to solve the problem.	Algorithmic Thinking	Breaking Down Problems Into Parts	decomposition

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
4.AT.1	Examine logical reasoning to predict outcomes of an algorithm.	Using critical thinking to analyze algorithmic processes allows students the opportunity to practice the skill of explaining how the algorithm gives you the predicted outcome. Through this process, students are being introduced to the foundational concept of pattern recognition. For example, students could run an algorithm and identify the pattern being produced. Once they identify the pattern, they will use logical reasoning to explain the algorithm and predict what the next five outcomes will be. Finally, they can test the algorithm to determine if their prediction is correct.	Algorithmic Thinking	Logical or Sequential Thinking	pattern recognition
4.AT.2	Use flowcharts to create a plan or algorithm.	Students should use flowcharts to organize and sequence an algorithm that addresses a complex problem. For example, students might express an algorithm that produces a recommendation for building a flower bed based on inputs such as size, sun exposure, climate, plants available, and cost. Testing the algorithm with a wide range of inputs and users allows students to refine their recommendation algorithm and to identify other inputs they may have initially excluded.	Algorithmic Thinking	Create Step-by-Step Processes to Solve Problems	flow chart

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
4.AT.3	Construct a basic system of numbers, letters, or symbols to represent information as a cipher.	Computers use a two-number system called binary code to store and process information. A cipher is an algorithm for encrypting or decrypting information important for data security. Students can create a simple encoding system which involves assigning values to symbols to represent specific meaning, allowing for secure communication. For example, students might create a cipher, such as pigpen, to write an encrypted letter to a friend or family member. The recipient of the letter can use the cipher to decode the message.	Algorithmic Thinking	Logical or Sequential Thinking	cipher, binary code, encode, encryption, decode
5.AT.1	Analyze and improve an algorithm that includes sequencing and simple patterns with or without a computing device.	Algorithms should be written to accomplish a task efficiently and with the fewest number of steps. Students can use a computing device or a graphic organizer, such as a flow chart, to represent the algorithm. For example, as an unplugged activity, students can select a daily process, such as getting ready for school, and share their algorithm for completing this task. Student can then share their algorithms with each other and identify ways to make their routine more efficient.	Algorithmic Thinking	Breaking Down Problems Into Parts	algorithm, sequence, flow chart
5.AT.2	Create an algorithm to solve a problem while detecting and debugging logical errors within the algorithm.	Building off of 3.AT.2, teachers will introduce the skill of finding and fixing logical errors or flaws within the algorithm during the creation process. This process is known as debugging. When a student is writing an algorithm, they should consider the specificity of their instructions and the order in which steps are completed. An important step of the process is to test and refine the algorithm if the intended output is not achieved.	Algorithmic Thinking	Create Step-by-Step Processes to Solve Problems	debug

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
5.AT.3	Develop and recommend solutions to a given problem and explain the process to an audience.	A key component of the design process is effectively communicating your process to others through multiple modalities. Students will identify what steps they took to define and solve the problem and create an explanation that can be understood and replicated by anyone. For example, students identify the problems associated with plastic waste, identify possible solutions, such as recycling, and present the step-by-step processes for each solution to an audience.	Algorithmic Thinking	Breaking Down Problems Into Parts	communication

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.AT.1	Use clearly named variables of various data types to create generalized algorithms.	Variables are placeholders in programming that can be used to keep track of a value that can change while a program is running. The use of variables can make algorithms more efficient. Students may understand this concept to be like a backpack. Having your backpack is part of the algorithmic process of getting ready for school each day. On Monday, they may just need a pencil and notebook, but on Tuesday, when they have practice after school, they will need their sports gear. The contents of that backpack will change based on the need of that day, but they will always have their backpack when they get ready for school.	Algorithmic Thinking	Create Step-by-Step Processes to Solve Problems	variable, data type, algorithm

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.AT.2	Create algorithms which include methods of controlling the flow of computation using "if…then…else" type conditional statements to perform different operations depending on the values of inputs.	Conditional statements are like a decision tree or a flow chart and can help make algorithms more efficient. When an algorithm gets to a point where a choice needs to be made, the decision is made based on whether the conditions are met. For example, students might create an algorithm to code a program to appropriately dress a character using conditional statements. IF it is raining outside, THEN I will bring an umbrella; ELSE, I will leave the umbrella at home.	Algorithmic Thinking	Create Step-by-Step Processes to Solve Problems	conditional statement, computation, operation, input
MS.AT.3	Identify algorithms that make use of sequencing, selection, or iteration.	There are many different types of algorithms that produce specific outcomes or have specific purposes. They all have been designed based on sequencing (specific step-by-step order), selection (decision-making), and iteration (repetition). Just as students should identify which operation, they should use to solve an equation, they can identify how an algorithm uses sequencing, selection, or iteration to solve a problem. This is the first step in learning how to develop a more efficient algorithm. For example, students might be given a variety of algorithms and are asked to identify the one demonstrating the best use of sequencing for solving a particular problem.	Algorithmic Thinking	Breaking Down Problems Into Parts	sequencing, selection, iteration

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.AT.4	Describe how algorithmic processes and automation increase efficiency.	Students should understand how algorithmic processes and automation can enhance performance by saving time, reducing errors, optimizing resource usage, ensuring consistency, and enabling scalability to solve a variety of different problems. For example, if students are coding a robot to complete a task and the algorithm has more steps than necessary, it is not efficient. The students should be able to communicate the improvements that need to be made to the algorithm to improve performance.	Algorithmic Thinking	Logical or Sequential Thinking	automation, efficiency

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.AT.1	Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.	Students should be able to identify common features in multiple segments of code and substitute a single segment that uses lists (arrays) to account for the differences. A list is a data structure for storing ordered values. While an individual variable could be created for each value, it would be more efficient to use a list that allows multiple values to be stored as a single variable. This also allows for easier manipulation of the algorithm. For example, students could be tasked with designing and testing an algorithm to simplify the processes of calculating grade averages. Instead of handling each grade as a single variable, the student will use an array called grades for a simplified, adaptable variable in their algorithm.	Algorithmic Thinking	Create Step-by-Step Processes to Solve Problems	list, variables, array, computation, data type

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.AT.2	Systematically design and develop programs for broad audiences by incorporating feedback from users.	Students can begin by simply collecting feedback and revising programs such as games, utilities, and mobile applications. As students progress through this standard, they should revise their program using a systematic process that includes feedback representing multiple perspectives. Students might create a user satisfaction survey and brainstorm distribution methods that could yield feedback from a diverse audience, documenting the process they took to incorporate selected feedback in product revisions.	Algorithmic Thinking	Create Step-by-Step Processes to Solve Problems	process, distribution
HS.AT.3	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	A prototype represents a computational artifact showcasing the functions of a product or procedure. These prototypes serve as valuable tools for acquiring early input in the design phase and can provide insights into the practicality of a product. The process of crafting computational artifacts includes both creativity and the exploration of how to create prototypes to solve computational problems. Students should generate computational artifacts that hold personal significance and take into consideration an algorithm's performance, reusability, and ease of implementation.	Algorithmic Thinking	Logical or Sequential Thinking	prototype, artifacts

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.AT.4	Use effective communication and accurate computer science terminology to explain problem solving when completing a task.	Vocabulary is the foundation of content learning and an important aspect of college and career readiness. Students must understand the terminology and use it appropriately when explaining their problem-solving process as well as use appropriate communication methods. For example, the students might create a presentation to explain how they developed an app. In this presentation, they might identify decomposition as how they break down the parts of a problem and explain their use of algorithmic thinking when discussing the order in which they solve the problem.	Algorithmic Thinking	Breaking Down Problems Into Parts	decomposition, algorithmic thinking

Programming Concepts

A **program** is a set of instructions a computing device executes to achieve a particular objective.[3] These instructions can come in a variety of forms, but programming represents an opportunity for students to expand their knowledge and application of computational thinking, collaboration, hardware and software application, and designing algorithms. In the primary grades, students will build on their understanding of programming concepts by using physical devices that require students to push buttons in sequential order on the device to solve a problem and begin to explore **block-based programming**. In later grades, students will develop even more complex programs using a variety of programming languages, which might include more advanced block-based programming or text-based programming. The focus is not on learning a particular programming language, as these will change by the time the student graduates. Rather, the goal is to build a set of skills that can be applied to programming in general, including the craft of designing, writing, testing, and maintaining programs to solve problems. At various degrees, students will learn how to troubleshoot or debug existing programs, use arithmetic operators, functions, parameters, conditionals, repetition in programs, and engage in best practices around collaborating in program design.

---

[3] *Computer Science Teachers Association (2020). K-12 CS Education Glossary. Retrieved December 15, 2022, from https://csteachers.org/glossary.*

Programming Concepts

Sub-Concepts / Grade Bands	Early Elementary (K-2)	Upper Elementary (3-5)	Middle (6-8)	High (9-12)
Using and Troubleshooting Existing Code		Students will work collaboratively to test simple algorithms and programs to identify errors and practice debugging.	Students will incorporate continuous improvement through the iterative design process. Students will collaborate with others to improve performance and eliminate errors.	As programs become more complex, students will need to rely more heavily on the iterative design process to improve the efficiency of programs and user experiences.
Developing Code to Solve a Problem		Students can decompose problems into simple algorithms and programs and can use pattern recognition to develop simple loops. Students should develop procedures both as individuals and as teams.	Students can decompose problems into sub-problems and isolate the steps to solve each sub-problem. Students can consider ways to reuse procedures to solve similar problems and isolate these steps through abstraction.	Students can apply a variety of control structures (Boolean logic, nest loops, compound conditionals and third-party libraries) in code to solve more complex problems.
Concepts of Programming (Language Exposure)		Students will be able to identify a problem and come up with a solution by decomposing a problem into smaller easier-to-solve steps. Students will also begin to recognize that information can be represented in more than one way.	Students should begin practicing with a programming platform that will allow them to apply various programming concepts. Students will be introduced to coding jargon including comments, libraries, attributions, and functions.	Students learn to develop variable plans to manage data within a program. Students can select the proper attributes of variables to maximize program efficiency and readability. Students can use libraries and open-source code with proper attribution.

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
3.PC.1	Analyze a given list of sub-problems while addressing a larger problem.	When solving sub-problems within a larger problem, students should consider relationships between sub-problems that might impact the overall result. This is an important step in decomposition. For example, if making a sandwich is the larger problem, deciding on which ingredients to put on each side of the sandwich might be a sub-problem. How well those ingredients complement each other would be an important analysis to consider (i.e., putting peanut butter on one side and ham on the other side might lead to an undesirable outcome).	Programming Concepts	Concepts of Programming (Language Exposure)	analyze, problem, decomposition
3.PC.2	Define a problem or task, decompose it into smaller sub-problems.	Problem-solving processes start by investigating the problem to understand it more (3.AT.1). Part of that process is for students to recognize that larger problems can be broken down into smaller sub-problems through decomposition. For example, needing to dress for the weather is the larger problem. The smaller problems might include selecting the right pants, shirt, hat, jacket, or shoes. By solving the smaller problems, the larger problem becomes solved.	Programming Concepts	Concepts of Programming (Language Exposure)	decomposition problem, computational thinking
3.PC.3	Use numbers or letters to represent information in another form.	Programming languages regularly use variables to represent values. Students need to recognize that information can be represented in various ways. For example, on a color-by-numbers activity, spaces with a 10 might be colored red, spaces with a 12 might be colored blue, etc..	Programming Concepts	Concepts of Programming (Language Exposure)	information, programming, variables

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
4.PC.1	Test and debug a given program in a block-based visual programming environment using arithmetic operators, conditionals, and repetition in programs, in collaboration with others.	Students need to follow a program, break it into sub-programs, and be able to test each sub-program to identify the location of the bug and correct it. Bugs can come in the form of misspellings, mislabeled operators or incorrect logic when using conditionals. Using pair-programming, students can work together to program and debug. For example, a virtual robot is supposed to travel in a square by driving in a straight line and turning right, then repeating that pattern 4 times. But the robot only travels 3 times and stops prompting students to debug and re-test.	Programming Concepts	Using and Trouble-shooting Existing Code	Debug, block-based program, repetition, conditionals
5.PC.1	Create simple animated stories or solve pre-existing problems using a precise sequence of instructions and simple loops, collaboratively or individually.	Programs are created after students have created an algorithm or series of steps to solve a problem. Students are introduced to sequential programming during algorithmic thinking (5.AT.1) and this allows students to build on their knowledge to create their own algorithms. Once a student has determined the flow of the steps, they can implement these steps to see the problem resolution. Artifacts can be digital or offline. An unplugged example might include a team of students who create a set of instructions to have a peer walk through an obstacle course or a student uses a coding program to make a character move in sequence to music.	Programming Concepts	Developing Code to Solve a Problem	algorithm, loop

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
5.PC.2	Identify bugs (errors) in basic programming.	Algorithms and programs may have errors and need to be debugged. Students will improve their ability to understand algorithm logic and find errors (5.AT.2). To fix problems in algorithms and programs, students will use various strategies like changing the order of steps, following the algorithm step-by-step, or using trial and error.	Programming Concepts	Using and Troubleshooting Existing Code	bug, debug, algorithm

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.PC.1	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Students need to recognize that problems can be broken down into smaller sub-problems. Through the identification of the subproblems students can evaluate and revise each step of the design process solving potential problems (bugs) as they arise. Students will expand on learning in MS.AT.4 to continue to improve the efficiency of solutions. For example, students are given an image of a Lego brick house and are instructed to create directions for how to recreate the image. They will need to consider multiple aspects of the design including considerations for parts of the house that are not visible.	Programming Concepts	Developing Code to Solve a Problem	decomposition, computational thinking

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.PC.2	Create procedures with parameters that hide the complexity of a task and can be reused to solve similar tasks.	Procedures are algorithms used to solve a problem that can be reused in similar situations. Students use abstraction to develop these procedures by focusing on the important and necessary parameters. For example, we have a procedure for placing an order at a fast-food restaurant. We review the menu, decide on food and drink, approach the cashier (or kiosk), place the order, and pay. Each time we go to a fast-food restaurant we follow a similar procedure regardless of the location and this simplifies our interaction and decision process. Students also use procedures and abstraction in class with formulas used in science or math where one formula can give the answer for different problems based on input.	Programming Concepts	Developing Code to Solve a Problem	parameter, computational thinking, algorithm, procedure, abstraction
MS.PC.3	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Collaborating in computing allows for continuous improvement and more refinement of a solution. Students need to give and receive feedback in a variety of situations. Giving the students time to internalize and make use of the feedback often leads to new perspectives when viewing problems.	Programming Concepts	Using and Troubleshooting Existing Code	feedback, collaboration

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.PC.4	Provide proper attribution when incorporating existing code, media, and libraries into original programs.	Using and incorporating existing code is essential to increase efficiency in coding and developing a new program. A programmer for an application doesn't need to know how the speaker works on the phone but instead will just connect to specific procedures to distribute sound. When borrowing or using code from others, a programmer will give proper attribution or citation to the original developer. For example, if a student uses direct quotes or indirect quotes, they will create a citation giving credit to the author and the source.	Programming Concepts	Developing Code to Solve a Problem	program, code, software, attribute
MS.PC.5	Use the iterative design process to systematically test and refine programs to improve performances and eliminate errors.	Students need to use an iterative design process to change and improve their design as they go. Part of the iterative design process involves changing and improving the design. This should be conducted several times during the process. The iterative design process is a continuous cycle of prototyping, testing, and making adjustments. Improvements can be made to user experience or to code efficiency (i.e. Loops MS.AT.4) For example, traditionally you will build a prototype from start to finish and test it afterwards. Using iterative design process, students will improve the design as they create it to eliminate errors.	Programming Concepts	Using and Troubleshooting Existing Code	iterative design, prototype

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.PC.6	Document programs using comments and/or README files to make them easier to follow, test, and debug.	Documentation (often called "commenting" a program) in a computer program is text in the program that the computer does not execute when running the program. Documentation (and/or README files) are imperative for anyone else to follow the written program and for the original author who may be looking at the program they wrote months or years after the original program was written.	Programming Concepts	Concepts of Programming (Language Exposure)	README, comment, program, debug
MS.PC.7	Design a function using a programming language.	Functions are sections of code that execute a procedure or routine. Many functions return a value important in the program. Using functions in a program helps to simplify the code to make it easier to troubleshoot, read, and execute. For example, there may be a function for turning right that would be reused repeatedly in a maze program.	Programming Concepts	Concepts of Programming (Language Exposure)	function, algorithm, computational thinking, troubleshooting

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.PC.1	Choose and apply an appropriate iterative design process to systematically test and refine a program to increase performance.	Iterative design processes such as the Engineering Design Process and Design Thinking Process involve planning, designing, testing, making corrections, and repeating, as necessary. Given that students often encounter mistakes and need to revise their programs multiple times to achieve optimal efficiency, it's crucial to apply and build on the principles of peer feedback and refinement learned in middle school (MS.PC.3) and through algorithmic thinking in high school (HS.AT.2).	Programming Concepts	Using and Troubleshooting Existing Code	iterative design, Engineering Design Process, efficiency

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.PC.2	Develop a plan to manage and assign data values of different types (strings, numeric, character, integer, and date) to a variable.	Before coding, students should create a clear plan that outlines the information needed by the program. This information can be stored as different variables, which include the type, value, and a purposeful name. Students can further enhance their variable planning by using lists to organize similar variables (referring to HS.AT.1). Meaningful variable names, such as "sideLength" for a polygon side's length instead of vague names like "l," or using integers for numbers instead of strings, simplify program modification and troubleshooting.	Programming Concepts	Concepts of Programming (Language Exposure)	values, variables, coding, lists
HS.PC.3	Create and refine programs with Boolean conditionals to demonstrate the use of branches and logical operators.	A Boolean is an expression that evaluates either true or false and allows for decision making in a program. They can be nested within or combined with conditional statements (if/then, if/else, etc.) to develop complex programs or algorithms. For example, students could develop a career choice algorithm based on multiple inputs including true/false answers to questions such as "I like being outdoors" or "I like working with animals." Answering true to both might indicate a career in zoology while a false to the first and a true to the second might indicate a different career.	Programming Concepts	Developing Code to Solve a Problem	program, Boolean conditionals, branches, logical operators, algorithm

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.PC.4	Design and develop iterative programs that combine control structures, including nested loops and compound conditionals.	Efficient programming may require the application of nested loops (loops within loops) and compound conditionals (complex conditions) to make decisions based on multiple criteria. By achieving proficiency in designing and developing iterative programs with these control structures, individuals can efficiently solve problems that require repetitive actions or intricate decision-making processes.	Programming Concepts	Concepts of Programming (Language Exposure)	iterative programs, control structures, nested loops, compound conditionals
HS.PC.5	Create parameters to organize a program to make it easier to follow, test, and debug.	Parameters are used to pass information from the main program to functions. They should follow the same conventions as variables, with clear types, values, and names that are intuitive to their purpose. For example, if students were to create a function for the force acting on an object, (F=MA), the function would need to know the Mass and the Acceleration of the object to return the correct force.	Programming Concepts	Using and Concepts of Programming (Language Exposure)	parameters, program, debug, variables, function
HS.PC.6	Incorporate existing code, media, and libraries into original programs, and give proper attribution.	The skill of integrating pre-existing code, media, and software libraries into one's own original programs or projects while adhering to proper attribution practices is fundamental to programming. Utilizing available resources can enhance the functionality and features of new software. Proper attribution similar to citations when writing a paper promotes ethical coding practices by requiring individuals to give due credit to the original authors or sources.	Programming Concepts	Concepts of Programming (Language Exposure)	libraries, programs, code, software, attribute

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.PC.7	Debug (identify and fix) errors in an algorithm or program that includes sequences and simple and complex loops following a two-step debugging process.	Troubleshooting or debugging allows students to refine programs to obtain the intended output or solution. These errors (bugs) can be due to errors with logic, sequence, math, or syntax. There are many approaches to debugging a program and several methods should be applied as a student develops their programming skills. Methods may include inserting print statements, paying attention to error messages or exceptions after running the program, or isolating parts of your code to narrow down the error.	Programming Concepts	Concepts of Programming (Language Exposure)	Debug, output, syntax, bugs

Data and Analysis

In our expansive digital world, computing systems are being used to collect, store, organize, explore, analyze, and process large quantities of data. Data can include a variety of information across a range of formats including numbers, images, text, audio or video files, software programs, or apps. Understanding the process of how data is used to make a variety of decisions is a crucial skill for students in the 21st century and beyond.

This core concept begins with data collection, using a variety of age-appropriate tools and processes, followed by data organization and reliable representation. Spreadsheets, databases, tables, charts, graphs, tabulating, and statistical analysis are just a few examples of the tools for data organization and representation. Further analysis of data aids students in discovering relationships within the data, including emerging patterns or evidence of a phenomenon or process. Beginning in middle school and expanding in high school, students can develop and critically evaluate **computational models** that are based on existing data.

Communicating About Computing is the *K-12 Computer Science Framework's* Practice #7, and communication is one of the 4Cs in the 21st Century Skills. Both support the idea that clear communication of data is necessary to articulate ideas responsibly and effectively. Ideally, communication of data can be used to identify trends, make predictions or inferences, and solve problems.

Data and Analysis					
Sub-Concepts	**Grade Bands**	**Early Elementary (K-2)**	**Upper Elementary (3-5)**	**Middle (6-8)**	**High (9-12)**
	Organizing, Clustering, Categorizing Data	Students recognize that data can be grouped in a variety of ways. Students will use age-appropriate tools and processes to collect and organize data.	Students collect and organize more complex sets of data. Students will also identify solutions and connect collected data to real-life events.		
	Cleaning or Transforming Data to Discover Useful Information	Students utilize data by interpreting charts, identifying relationships, and discovering solutions within the organized data.	Students manipulate a data set by rearranging or removing unnecessary data in order to answer questions.	Students examine, improve, clean, or change data by reorganizing or removing elements. Students use data to make adjustments to computational models.	Students identify patterns in data from complex systems by using data analysis tools and techniques.
	Displaying or Visualizing Data	Students collect and organize data to create digital representations.	Students use data to identify relationships, predict outcomes, communicate ideas, and answer questions using a variety of computing and data visualization representations.	Students use encoding schemes to represent data utilizing characters, symbols, and alphabets.	Students analyze data using computing and data visualization methods to find solutions.
	Communicating Data for Decision-Making or Problem-Solving		Students describe examples of data sets from everyday life and produce published results.		Students create computational models that display multiple relationships within data collected from a process or phenomenon.

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
K.DA.1	Collect and organize data.	Data can include a variety of information across a range of formats including: numbers, images, text, audio/visual files, software programs, and apps. Students will use age-appropriate tools and processes to collect and organize data. For example, students will collect data and group the data by making connections.	Data and Analysis	Organizing, Clustering, or Categorizing Data	data
1.DA.1	Interpret data displayed in a chart.	A chart is any type of visual display or diagram that organizes and represents a set of data. Students should be able to understand and interpret data, even if they did not collect it themselves. When given a chart, students can start by identifying what data was collected, how it was organized, and what it is meant to convey. For example, the students could create a title for the data chart and correctly identify or label the categories of data that were collected. Then, they can look for patterns or trends.	Data and Analysis	Cleaning or Transforming Data to Discover Useful Information	data, chart
1.DA.2	Organize data using similarities and differences.	Data can be grouped in a variety of ways by finding attributes that are similar or separating attributes based on differences. Pattern recognition, a component of computational thinking, is important for organizing data. For example, a collection of animals can be grouped based on color or type of animal.	Data and Analysis	Organizing, Clustering, or Categorizing Data	data, pattern recognition
2.DA.1	Use data to make decisions, identify solutions, or determine relationships.	Data helps us make informed decisions. Building on the skills of collecting (K.DA.1) and organizing (1.DA.2) data, students will analyze data to identify relationships and make decisions.	Data and Analysis	Cleaning or Transforming Data to Discover Useful Information	data

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
2.DA.2	Use if/then reasoning to understand relationships with data.	In addition to being an essential component of computer programming (4.PC.1), the concept of cause and effect applies to a variety of academic areas and real-life situations. Students can identify if/then relationships in a set of data and recognize that the outcome changes depending on the conditions. Flow charts are one way to organize data using if/then reasoning.	Data and Analysis	Cleaning or Transforming Data to Discover Useful Information	if/then statements, data, flow chart, conditional
2.DA.3	Collect, create, and organize data in a digital chart or graph.	Raw data needs to be organized to allow for interpretation or clearer communication - raw data has little meaning on its own. Methods for organizing data include tables and charts, such as pie, bar, or line graphs. Students will be able to organize data (whether it is numerical or categorical) using basic digital tools. For example, students can input data about favorite ice cream flavors into a Google sheet or the free online tool "NCES Create a Graph" and create a pie chart or bar graph to display the data.	Data and Analysis	Displaying or Visualizing Data	data, chart, graph

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
3.DA.1	Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.	Once organized properly, data can be analyzed to uncover correlational or cause and effect relationships. Already knowing how to identify cause and effect relationships (2.DA.2), students can extend a pattern or relationship that is shown in the collected data and predict values beyond those collected. They should also be able to refer to data when communicating an idea. For example, data collected on rainfall	Data and Analysis	Displaying or Visualizing Data	data, cause-and-effect relationships

Upper Elementary (3-5)					
Identifier	Standard	Explanation	Concept	Sub-Concept	Vocabulary
		per month may uncover rainy seasons or allow for predictions about future weather patterns. Likewise, students can also identify if the data set includes data that is not useful for predicting outcomes. For example, when collecting data about whether items are magnetic or not, students might be able to identify that the presence of metal in an object does affect its categorization, but the color of the object does not.			
3.DA.2	Describe examples of data sets or databases from everyday life.	Data sets or databases are organized information that can be digital or traditional including numbers, words, images, etc.. For students to be able to describe examples, students must have an understanding of what a data set/database is as well as know where to look for such information. For example, students could explain how their testing scores compiled for the class, grade, or school is an example of a data set. Students could also explain how banks use databases in order to keep customer information.	Data and Analysis	Communicating Data for Decision Making or Problem Solving	data set, database, digital
4.DA.1	Collect, organize, analyze, and interpret data to identify solutions and/or make informed decisions.	Data analysis follows a sequential path that students are increasingly familiar with (K.DA.1, 1.DA.2, 2.DA.1, 2.DA.3, 3.DA.1). First, students collect data using age-appropriate tools. Next, students organize the data using spreadsheets, tables, charts, or graphs. Then students analyze relationships within the data, including recognizing patterns. Finally, students will then interpret data to identify a solution to a problem.	Data and Analysis	Organizing, Clustering, or Categorizing Data	data, data analysis, chart, spreadsheet

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
4.DA.2	Gather data to answer a question using a variety of computing and data visualization methods.	Questions can be answered using a variety of data collection methods, including surveys, interviews, observations, and reviews of existing records. Once gathered and organized, computing methods could be used to get closer to an answer to the given question. Computing methods help solve mathematical processes in ways a computer does - such as through decomposition and abstraction. Finally, data visualization methods provide an accessible way to see trends and allow students to communicate data relationships in ways that are easy to understand. For example, they could use data from the Census Bureau to ask and answer questions about population changes in the United States during Westward Expansion.	Data and Analysis	Displaying or Visualizing Data	data, computing methods, data visualization, decomposition, abstraction
5.DA.1	Manipulate data to answer a question using a variety of computing methods and tools to collect, organize, graph, analyze, and publish the resulting information.	Some questions require that data be transformed (manipulated) using computational methods (4.DA.2) in order to be answered and for students to accurately display the results. In computer science, data transformation or manipulation refers to the process of changing the format, structure, or content of data to make it more useful, relevant, or suitable for a particular purpose or task. This can involve various operations, such as cleaning data which includes removing or correcting errors, inconsistencies, or irrelevant data from a dataset or transforming data such as converting data from one data type to another for use in a program.	Data and Analysis	Cleaning or Transforming Data to Discover Useful Information	data, data cleaning, data transformation

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
5.DA.2	Connect data from a simulation to real-life events.	Simulations are a way of imitating a process or system in real-life events. They can be used to demonstrate how complex systems work, test ideas, and make predictions. Students can manipulate factors and collect data within the simulation, then compare it to data collected in real life. For example, a simulation of life in pioneer times (like The Oregon Trail) could be compared to primary sources, or a simulation of simple circuits could be confirmed with actual circuits.	Data and Analysis	Organizing, Clustering, or Categorizing Data	simulation, data

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.DA.1	Represent data using multiple encoding schemes, such as decimal, binary, Unicode, Morse code, Shorthand, student-created codes.	Students should be able to represent data in multiple ways. Encoding is the process of changing data or a shared order of characters, symbols, alphabets into a connected format, for the ensured communication of data. Communicating the encoding scheme is as important as encoding itself, to ensure other users or machines can accurately interpret the information. Computing systems can use a variety of character encoding schemes. These are some of the first steps of data encryption that lay the foundation for cyber-security. For example, human speech can be stored as an audio file, but also translated into a text file or Morse code.	Data and Analysis	Displaying or Visualizing Data	encoding, computing system, binary, Unicode, Morse code, algorithm, data encryption, cybersecurity

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.DA.2	Refine computational models based on the data they have generated.	Computational models are representations of phenomenon and the factors that influence them in a computing format. After learning how simulations represent real life data (5.DA.2), students now can make changes to computational models if the data does not match what real life would dictate. Students can examine model-generated data and improve the computational model to make it more useful or reliable. For example, students might refine the code in a model of the Pythagorean theorem if it originally included negative numbers. Additionally, students could refine game mechanics based on test outcomes in order to make the game more balanced or fair.	Data and Analysis	Cleaning or Transforming Data to Discover Useful Information	computational models, data
MS.DA.3	Collect, analyze, transform, and refine computational data to make it more useful and reliable.	Computational data is information represented in a digital format or processed using computer techniques. To make data useful, students may have to clean the data. Cleaning data focuses on finding errors then changing, updating, or removing unhelpful, erroneous, or outdated data in order to be left with a corrected data set. For example, after collecting survey responses about extracurricular student interests, to make the data more reliable check for duplicates, incomplete answers, and group similar choices.	Data and Analysis	Cleaning or Transforming Data to Discover Useful Information	computational data, data cleaning, digital

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.DA.1	Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.	Computational models help predict outcomes based on chosen data and features. The quality and amount of data, along with feature selection, affect model quality. Testing predictions validates models. Abstraction is key to creating a computational model as students will need to determine what is most important when modeling a system or relationship. As an example, students could start by making a basic producer-consumer ecosystem model with a computer program based on prey-predator relationships. They could then validate their model based on real-world data collected from nature.	Data and Analysis	Communicating Data for Decision Making or Problem Solving	computational model, data, abstraction
HS.DA.2	Utilize data to answer a question using a variety of computing and data visualization methods.	People transform, generalize, simplify, and present large data sets in different ways to influence how others interpret and understand the underlying relationships. Visual representations of information communicate complicated data interactions and data-driven understandings in a way that is easy to understand. Data visualization is commonly used to drive idea creation across teams. For example, students could simplify data on traffic patterns, and represent their findings in a visual manner in order to determine and communicate necessary mass transit options.	Data and Analysis	Displaying or Visualizing Data	data, data set, data visualization, computing methods

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.DA.3	Use data analysis tools and techniques to identify patterns in data representing complex systems.	Data analysis techniques include the different ways to analyze data (regressive, predictive, cluster, correlation, etc.) while data analysis tools include the resources available to aid in investigating data (excel, sheets, google analytics, etc.). Complex systems are what emerge from many interrelated elements that have complex relationships due to several contributing factors that lead to a whole infrastructure (global climate, ecosystems, human brain, etc.). Students are to analyze data that displays a specific complex system by focusing on a distinct data analysis technique to find patterns. For example, students may use Google Analytics to do a predictive analysis of climate change.	Data and Analysis	Cleaning or Transforming Data to Discover Useful Information	data, data analysis

Networking and the Internet

This core concept allows students to understand the connectivity of their digital world, including how information is packaged and transmitted across networks and the internet. A **network** is a group of computing devices (personal computers, phones, servers, switches, routers, etc.) connected by cables or wireless media that enable the exchange of information and resources. The **Internet** is a global collection of computer networks and their connections, all using shared protocols to communicate.[3] In addition, this concept also focuses on how to troubleshoot these systems. Many of the other core concepts are essential for understanding this concept as it brings together data, hardware and software, algorithmic thinking, and programming to design and securely use these networks and connections.

Embedded in this concept is the understanding of both physical and digital security measures to protect electronic information and intellectual property and laws in the digital space. Students' education around **cybersecurity** can lead to a safer digital environment, including creating more effective passwords, understanding the differences between secure and non-secure websites, and understanding what personal data is collected and shared across these networks.

[3] *Computer Science Teachers Association (2020). K-12 CS Education Glossary. Retrieved December 15, 2022 from https://csteachers.org/glossary.*

Networking and the Internet

Sub-Concepts / Grade Bands	Early Elementary (K-2)	Upper Elementary (3-5)	Middle (6-8)	High (9-12)
Connectivity and Issues			Students identify and solve basic connectivity issues. Students understand the relationship between the type of network, connectivity issues, and troubleshooting methods.	Students identify and solve advanced connectivity issues. Students understand and apply more sophisticated methods for preventing and troubleshooting connectivity issues.
Data is Transferred Using Protocols or Rules		Students understand there are tools and techniques to improve the outcome of an internet search.	Students understand rules for data transfer on the Internet. Students analyze different ways to secure data and enhance cybersecurity methods.	Students evaluate and discuss rules and protocols for transferring data on the Internet and their importance for cybersecurity. Students analyze these processes for efficiency, feasibility, and ethical impacts.
Protecting Data/Personal Information	Students understand behaviors that promote safe usage of digital devices, tools, and the Internet.	Students demonstrate safe behaviors online deeming them a "good digital citizen." Students begin to understand the permanence of actions in the digital world.	Students understand the risks and benefits of online security and data protection. Students understand intellectual property and its relation to accessing and using digital media.	Students understand Internet and technology-related laws, enabling them to make informed decisions when online. Students recognize their digital identity may be permanent and could impact their future opportunities.

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
K.NI.1	Demonstrate age-appropriate methods for keeping personal information private.	As students begin to interact in the digital world, it is important for them to understand what is personal information and the importance of keeping it private. Age-appropriate methods may include passwords, face identification, and fingerprint scans. Students should understand not to share passwords, name, address, phone number or location with others online. Teachers can lead students through appropriate discussions on why this information should remain private.	Networking and the Internet	Protecting Data/Personal Information	personal information, private, passwords
1.NI.1	Advocate, demonstrate, and routinely practice safe, legal, and responsible use of information and technology.	Students should understand how to be a good digital citizen. This includes safely using the internet, being kind and respectful to others online, and knowing what is their work and what is the work of others. For example, students should understand they should only use approved appropriate devices with adult supervision for safe use of the internet.	Networking and the Internet	Protecting Data/Personal Information	digital citizen
2.NI.1	Identify appropriate and inappropriate behaviors for communicating in a digital environment.	Students will explore and discuss online etiquette. Students will use kind words and be respectful when communicating with others online. For example, all caps and bold words indicate yelling.	Networking and the Internet	Protecting Data/Personal Information	online etiquette
2.NI.2	Cite media and/or owners of digital content.	Students can name the author and/or website where they get their digital content, which includes text and media. Students will learn and discuss the difference between content with a copyright versus creative commons.	Networking and the Internet	Protecting Data/Personal Information	citation, media copyright, creative commons

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
2.NI.3	Create a research-based product using online digital tools.	Students can research a topic using a variety of digital tools including databases and guided Internet search. As they research their topic, they can develop research strategies and organize digital information. Students can create a product (digital or non-digital) to explain their research.	Networking and the Internet	Protecting Data/Personal Information	digital tools

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
3.NI.1	Advocate, demonstrate, and routinely practice safe, legal, and responsible use of information and technology.	Students demonstrate their proficiency in digital citizenship by exemplifying respectful and responsible conduct when utilizing the Internet and digital technologies. For example, they understand their digital footprint mirrors their choices, the implications of cyberbullying, and realize reproducing the work of others without proper authorization is not appropriate.	Networking and the Internet	Protecting Data/Personal Information	digital footprint, cyberbullying, digital citizenship
3.NI.2	Conduct basic keyword searches to produce valid, appropriate results, and evaluate results for accuracy, relevance, and appropriate-ness.	Students practice determining the best keyword for online searches and apply basic Internet search operators such as " " to narrow search results. Students can look at domain suffix (.com, .edu, .gov) to determine the reliability and accuracy of the resulting search content from a source.	Networking and the Internet	Data is Transferred Using Protocols or Rules	keyword, search operator, domain suffix, Internet

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
4.NI.1	Identify appropriate and inappropriate uses of communication technology and discuss the permanence of actions in the digital world.	Students understand how email, messaging, and social media platforms are used for communication and contribute to their digital footprint. Digital communication can be stored and does not go way with the delete button. Inappropriate use of technology may include cyberbullying and harassment, which have laws that provide real-world consequences. Students know school and district rules for appropriate communication (refer to acceptable use policy).	Networking and the Internet	Protecting Data/Personal Information	digital footprint
4.NI.2	Conduct advanced keyword searches to produce valid, appropriate results and evaluate results for accuracy, relevance, and appropriate-ness.	Students will expand their keyword search expertise from using basic search operators like quotation marks to more advanced internet search operators. These may include using the minus sign (-) to exclude irrelevant results, the plus sign (+) to ensure both specified words appear in the results, and site-specific queries (site:). They will further their critique of reliable sources of information by learning how to determine potential bias from a website author or contributor to further their understanding of content relevance and appropriateness.	Networking and the Internet	Data is Transferred Using Protocols or Rules	keyword, search engine, search operator

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
5.NI.1	Explain responsible uses of technology and digital information; describe possible consequences of inappropriate use such as copyright infringement and piracy.	Students will understand there is an appropriate way to use technology and interact with digital information. This may include an awareness it's inappropriate to distribute images or messages without obtaining consent from the individuals involved. They also recognize the importance of refraining from unauthorized access to others' accounts, including email and school accounts, as well as avoiding impersonation of others online. Students understand fines and lawsuits may be a consequence of copyright infringement or piracy. For example, copying and pasting without proper attribution is a form of plagiarism.	Networking and the Internet	Protecting Data/Personal Information	copyright, copyright infringement, piracy, impersonate, plagiarism
5.NI.2	Apply copyright principles to real life scenarios.	Students can explain how copyright affects their school and personal lives and use copyright principles. Students understand what is in public domain and what is copyright protected material. For example, recording a movie in the movie theater and uploading to their personal YouTube page is copyright infringement.	Networking and the Internet	Protecting Data/Personal Information	copyright

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.NI.1	Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.	Students evaluate Internet connectivity issues and troubleshoot possible solutions in order to solve issues and resume digital tasks. Troubleshooting techniques may include rebooting computers, reloading tabs, checking network settings, etc. Students can determine if the issue is hardware or software. Students can identify the type of network they are using (wired vs wireless) and the components of that network.	Networking and the Internet	Connectivity and Issues	troubleshooting, connectivity, reboot, network
MS.NI.2	Differentiate between secure and non-secure websites and applications including how they affect and use personal data.	Students should understand the difference between http: and https: scheme in web addresses. Students discover that websites and applications collect personal information and track usage, which can be used to customize advertisements and mask phishing attempts.	Networking and the Internet	Protecting Data/Personal Information	secure, non-secure, applications
MS.NI.3	Describe the causes and effects of intellectual property as it relates to print and digital media, considering copyright, fair use, licensing, sharing, and attribution.	Students can distinguish between copyright, fair use, creative commons, and royalty free media. Students know how to provide attribution to the creator of content. For example, students understand that if they use an image that is not fair use or royalty free, they may be fined.	Networking and the Internet	Protecting Data/Personal Information	copyright, fair use, licensing, attribution

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.NI.4	Compare and contrast common methods of securing data and cybersecurity.	Students will learn about the various strategies and technologies for keeping data secure and how this is important for cybersecurity. They will determine the appropriateness of a strategy for different systems or networks. Examples may include strong vs. weak passwords, two-step authentication, and data encryption.	Networking and the Internet	Data is Transferred Using Protocols or Rules	cybersecurity, encryption, passwords, two-step authentication
MS.NI.5	Analyze different modes of social engineering and their effectiveness.	Social engineering is a form of manipulation where individuals with malicious intent use various tactics to trick people into divulging sensitive information or performing actions that they typically wouldn't. Examples might include phishing, baiting, and malware. Social engineering can be highly effective when targets are unaware or not adequately trained in cybersecurity awareness.	Networking and the Internet	Protecting Data/Personal Information	social engineering, malware, phishing, baiting, cybersecurity

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.NI.1	Explain the tradeoffs when selecting and implementing cybersecurity recommend-ations.	A network refers to a collection of interconnected devices or computers that can communicate and share resources including data with each other. Cybersecurity is crucial for networks because it protects sensitive data, privacy, and business continuity. It plays a vital role in maintaining national security and personal safety, especially for critical infrastructure networks. When selecting and implementing cybersecurity recommendations, tradeoffs involve balancing security measures with factors like cost,	Networking and the Internet	Data is Transferred Using Protocols or Rules	cybersecurity

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
		usability, and performance. Making decisions often means choosing between increased protection and the potential impact on resources and user experience.			
HS.NI.2	Identify laws regarding the use of technology and their consequences and implications.	Identifying laws related to technology means being aware of the rules and regulations that govern various aspects of technology use. For example, consider copyright laws, which dictate how digital content can be used and shared. Students can learn about Federal, State or even School District regulations and Acceptable Use Policies (AUP). Laws and regulations related to technology can change rapidly due to the dynamic nature of the tech industry and the need for policymakers to adapt to emerging technologies and evolving cybersecurity threats.	Networking and the Internet	Protecting Data/Personal Information	liability
HS.NI.3	Evaluate strategies to manage digital identity and reputation with awareness of the permanent impact of actions in a digital world.	Students should be aware that they can manage the visibility settings of their social media and online accounts, choosing between public, private, or selected users. Students can evaluate and select effective strategies for safeguarding their digital identity and reputation. For example, they should understand potential employers and colleges may review their online presence when making hiring or admissions decisions.	Networking and the Internet	Protecting Data/Personal Information	Digital Identity, Digital Reputation
HS.NI.4	Demonstrate how to apply techniques to mitigate effects of user tracking methods.	Students can identify and apply methods of safeguarding their identity and location online. Such methods include limiting location tracking on apps, private browsing online, and the use of VPN (Virtual Private Network) services to mask IP (internet protocol) addresses.	Networking and the Internet	Protecting Data/Personal Information	user tracking, VPN, IP address

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.NI.5	Show an understanding of the ramifications of end-user license agreements and terms of service associated with granting rights to personal data and media to other entities.	A license contract, also known as a license agreement or licensing agreement, is a contract where one party (the licensor) grants another party (the licensee) the right to produce, use, sell, and/or display the licensor's protected material. Students will understand and explain terms of service being a document describing the specific details about what a service provider is responsible for as well as user agreements that must be followed for continuation of the service. Examples include agreements you sign when you download Microsoft Office or get a new iPhone.	Networking and the Internet	Protecting Data/Personal Information	end-user license
HI.NI.6	Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	Security measures like tokens, two-factor authentication, and biometrics keep data safe. It's crucial to protect data because there are risks like cyberattacks (e.g., ransomware and phishing). When using computational tools to solve problems, students should evaluate the feasibility by considering factors like cost-benefit analysis. Additionally, students should consider other aspects such as efficiency and ethical concerns when evaluating a proposed approach.	Networking and the Internet	Data is Transferred Using Protocols or Rules	security measures
HS.NI.7	Demonstrate a fundamental understanding of API (Application Programming Interface).	An API, or Application Programming Interface, is code that allows different software applications to communicate and interact with each other, enabling them to share data and functionality. Students should be able to identify an API and understand its role. Common examples include a Google Map or local weather embedded on a website or a "log in with Facebook/Twitter/Google/GitHub" capability seen on many websites.	Networking and the Internet	Data is Transferred Using Protocols or Rules	Application Programming Interface (API)

Impacts of Computing

Computer science impacts society on a variety of levels, and we should empower students to move from being passive users of computing devices to critically evaluating how computer science affects their daily lives. These impacts might include reflecting on new ways to interact with others through digital media, how we receive up-to-date information about what is happening in our world, the potential of medical devices to improve our health and well-being or using drones and algorithms to determine how to maximize production of a crop. Computer science is everywhere and for it to fully benefit everyone, we must increase our understanding of its applications. Thus, it is essential for students to understand both the benefits and risks of computer science. Finally, *The K-12 Computer Science Framework* suggests the **Impacts of Computing** influences culture, supports networking and social interaction, and students need to know the fundamentals of digital citizenship to interact safely with computing devices.[4]

Impacts of Computing						
Sub-Concepts	**Grade Bands**	**Early Elementary (K-2)**	**Upper Elementary (3-5)**	**Middle (6-8)**	**High (9-12)**
	Innovations Due to Computer Science		Students can explain how software and hardware can be developed to address all user's needs.	Students will analyze computer science innovations and assess their effects on society, both positive and negative.	Students will understand current innovations in computer science and how these might adapt, change, and be developed to meet future needs.
	Benefits and Risks to Users and Developers Due to Computational Systems	Students will learn the appropriate skills to navigate safely in a digital environment.	Students will understand both the positive and negative impacts of computer science and how laws and tools can be used to develop positive digital environments.	Students will critically evaluate the advantages and disadvantages of computer science in regards to data collection, globalization, and internet censorship.	Students will investigate and debate issues surrounding social and economic implications of computer science in relation to safety, law, and ethics.

[4] Creative Commons. (n.d.). *K–12 Computer Science Framework*. k12cs.org. Retrieved January 5, 2023, from https://k12cs.org/

Impacts of Computing					
Sub-Concepts	**Grade Bands**	**Early Elementary (K-2)**	**Upper Elementary (3-5)**	**Middle (6-8)**	**High (9-12)**
	How Jobs/Careers Use Computer Science and Computational Thinking			Students will understand the role computational thinking and computer science play in a variety of educational pathways and careers.	The student will research and analyze the impacts of computing technologies on career fields/pathways to make informed decisions about their future.

Early Elementary (K-2)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
2.IC.1	Recognize and describe the potential risks and dangers associated with various forms of online communications (e.g., cell phones, social media, digital photos).	For students who are new to online communities, navigating the digital landscape can be both exciting and challenging. It is important for these new learners to recognize and understand the potential dangers that can arise when communicating through various devices and platforms. These risks may include encountering strangers, cyberbullying, and seeing offensive content. These dangers can appear in various online settings, including chats and online games. It's important to help new learners understand these risks to stay safe in the online world.	Impacts of Computing	Benefits and Risks to Users and Developers Due to Computational Systems	digital citizenship, online, cyberbullying

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
4.IC.1	Identify laws and tools which help ensure that users of varying abilities can access electronic and information technology.	Laws are established to help users access electronic information safely, ensuring accessibility for all students without gaps or limitations. One such example is the Rehabilitation Act of 1973, a federal law that mandates agencies make electronic and information technology accessible for individuals with disabilities. Students can gain a practical understanding of this by exploring technology tools designed to assist people like the Bigtrack mouse and talk-to-text software. These tools demonstrate how accessibility laws translate into real-world solutions.	Impacts of Computing	Benefits and Risks to Users and Developers Due to Computational Systems	computing device, accessibility
4.IC.2	Explain how hardware and applications can enable everyone, including people with disabilities, to do things they could not do otherwise.	Every student should be able to describe and communicate the suitable technologies and internet resources that can assist individuals from diverse backgrounds and cultures. Each student should analyze hardware and software applications suitable for educational environments and can cater to the needs of all users. This includes, but is not limited to, those requiring visual aids, text-to-speech capabilities, and sensory accessibility features.	Impacts of Computing	Innovations Due to Computer Science	software, hardware, accessibility

Upper Elementary (3-5)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
5.IC.1	Analyze the impact of social media on individuals, families, and society.	Examine and understand the influence of social media on individuals, families, and society, considering both its positive and negative effects. For example, a positive impact is evident when people use social media to maintain connections with distant family and friends, fostering positive social interactions. While a negative impact emerges when individuals share excessive personal information online, potentially compromising their privacy and security.	Impacts of Computing	Benefits and Risks to Users and Developers Due to Computational Systems	social media, privacy, security

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.IC.1	Identify and evaluate the impacts computer science innovations have had on our society.	Advances in computer science speed up progress and innovation, which benefit the economy, sciences, and society. Exploring these different impacts (both positive and negative) is beneficial as students become creative computer science thinkers. For example, innovations in medical technology, like continuous glucose monitors, and Artificial Intelligence developments have significantly changed our daily lives. However, excessive screen time might be reshaping our culture's social dynamics.	Impacts of Computing	Innovations Due to Computer Science	innovation, computer science, artificial intelligence

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.IC.2	Identify how computational systems are being used to collect and analyze information both public and private and understand the benefits and disadvantages of these systems for the user and developer.	Daily use of computing devices and software relies on the gathering and analyzing of huge amounts of data. Understanding when public or private data is being collected and how it is being used is important for students to learn. This data accumulation has both advantages and disadvantages depending on if you are the developer or the user. For example, a fitness tracker collects data about a user's physical activity, heart rate, and sleep patterns to help the user monitor their health and make informed decisions to improve their well-being. However, this same tracker could be sharing a user's location without their consent for targeted advertising.	Impacts of Computing	Benefits and Risks to Users and Developers Due to Computational Systems	data, computing device, software, public, private
MS.IC.3	Cite evidence of the positive and negative effects of data permanence on personal and professional digital identity.	Digital data can be stored for long or short periods of time depending on the type of information. This data permanence is like a digital footprint that can follow an individual with both positive and negative effects. Students should understand what you do today online can have an effect on your tomorrow.	Impacts of Computing	Benefits and Risks to Users and Developers Due to Computational Systems	data integrity, data type, digital, hardware, software

Middle (6-8)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
MS.IC.4	Discuss digital globalization and Internet censorship.	Digital globalization relies on digital technologies and the Internet to connect and integrate economies, cultures, and societies on a global scale. It involves the seamless flow of data, information, and services across national borders. Internet censorship is the practice of controlling, filtering, or regulating the content that can be accessed, published, or viewed on the internet. Governments, organizations, or internet service providers may impose these restrictions for various reasons including national security, cultural norms, or the prevention of harmful content. Students can discuss how these two ideas relate or are impacted by one another. For example, students can examine the impact of censorship on digital globalization, considering how restrictions on online content can hinder global communication and collaboration.	Impacts of Computing	Benefits and Risks to Users and Developers Due to Computational Systems	digital globalization, data, software, Internet censorship
MS.IC.5	Investigate a variety of education pathways and career options that utilize computational thinking and/or computer science skills across the state of Tennessee and the world.	Exploring educational pathways and careers that involve computational thinking and computer science skills can extend beyond the field of computer science itself. These skills have applications in various disciplines, including science, technology, engineering, mathematics, humanities, and arts. Opportunities for learning and pursuing careers related to computational thinking exist in both local and global communities within Tennessee and around the world.	Impacts of Computing	How Jobs/ Careers Use Computer Science and Computational Thinking	computational thinking

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.IC.1	Evaluate and debate the social and economic implications of computing in the context of safety, law, and ethics.	Understanding, assessing, and discussing the effects of computing on society and the economy, particularly within the frameworks of safety, legal considerations, and ethical principles allows students to become critical computer science thinkers rather than passive users. Students are encouraged to explore how computer-related activities impact individuals and businesses in terms of their well-being, rights, and values. For example, students could debate how the collection and use of personal data by companies can have both positive economic implications (targeted advertising) and negative social implications (privacy breaches and surveillance).	Impacts of Computing	Innovations Due to Computer Science	implications
HS.IC.1a	Discuss the ethical ramifications of hacking and its impact on society.	Hacking refers to the act of gaining unauthorized access to computer systems or networks, often with the intention of exploiting, altering, or stealing data. Students should explore how this practice can have profound ethical ramifications and far-reaching effects on society. For example, students can be given a scenario in which someone identifies a vulnerability in a computer system that allows data to be changed. They can then discuss who could be impacted by this vulnerability and steps to decrease its impact on users.	Impacts of Computing	Benefits and Risks to Users and Developers Due to Computational Systems	hacking, data, vulnerability

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.IC.1b	Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users (Bots, Chatbots, Spiders or Crawlers, Web Scraping, keyloggers etc.).	The collection and generation of data through automated processes can lead to privacy issues. Students should understand how data is gathered and used sometimes without their knowledge or consent in the digital world. This understanding can help students become more informed digital citizens.	Impacts of Computing	Benefits and Risks to Users and Developers Due to Computational Systems	privacy, personal information, data, automation
HS.IC.1c	Explain the positive and negative consequences that intellectual property laws can have on innovation.	Intellectual property laws are legal rules designed to protect the rights of creators and inventors by providing them with exclusive authority over their intellectual works. These legal measures encompass patents, copyrights, trademarks, and trade secrets. Students should be aware of intellectual property laws and be able to explain how they are used to protect the interests of innovators and inhibit innovation. For example, laws that mandate the blocking of some file-sharing websites may reduce online piracy but can restrict the right to access information.	Impacts of Computing	Benefits and Risks to Users and Developers Due to Computational Systems	intellectual property laws, patent, copyright, trademark

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.IC.2	Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.	Students should use a variety of innovative resources such as devices, platforms, software, etc. to facilitate collaboration between different groups of people. These innovations have allowed for interaction with those from different perspectives and opportunities to reach out to career professionals. For example, how does a virtual learning community impact those who otherwise could not access education from a brick-and-mortar school setting? How does remote work help with distributing resources and accessing educational/career opportunities?	Impacts of Computing	Innovations Due to Computer Science	software, connectivity
HS.IC.3	Research the impact of computing technology on possible education and career pathways.	Students will investigate how advancements in computing have created new opportunities, changed existing ones, and shaped the skills and knowledge needed for various educational and career paths. Students may analyze how technology has impacted fields like computer science, data analysis, artificial intelligence, and how it's integrated into traditional disciplines like healthcare, finance, and entertainment. By researching this impact, students can make more informed decisions about their educational and career trajectories in a tech-driven world.	Impacts of Computing	How Jobs/ Careers Use Computer Science and Computational Thinking	data analysis, artificial intelligence

High (9-12)					
Identifier	**Standard**	**Explanation**	**Concept**	**Sub-Concept**	**Vocabulary**
HS.IC.4	Predict how computational innovations that have revolutionized aspects of our culture might evolve.	Students are poised to become the inventors of the future, and it's essential for them to envision and assess how computer science is evolving to impact our world. They can discuss how novel innovations will bring about positive changes. For example, they could explore the potential effects of flying cars on our communities, how drones are aiding the agricultural sector or how Human-Computer Interaction (HCI) might impact our lives.	Impacts of Computing	Innovations Due to Computer Science	Human Computer interaction (HCI)

Deconstructing Computer Science Standards

An essential step in ensuring a curriculum, instruction, and assessment are aligned to a set of standards is by deconstructing those standards. [6] **Deconstructing standards** requires analysis of the language and extraction of clues to determine the **essential knowledge** (**what** students need to **know)** and **skills** (**what** students need to **do**) to demonstrate mastery.

Four examples for why working through the deconstruction process brings clarity to effective standard implementation are:

1. **Clarifying Expectations** - Deconstruction supports an educators' understanding of the content that is embedded in the standard so it can be accurately taught and assessed. [6]
2. **Building Understanding and Ownership** - Collectively working through the deconstruction process facilitates understanding of the standards and supports teachers taking ownership of the expected learning outcomes. [6]
3. **Meeting Rigorous Expectations** - Deconstructing standards strengthens teachers' knowledge about how to teach and assess standards at a level of rigor or cognitive demand that is intended. [6]
4. **Creating Learning Progressions** - The deconstruction process helps educators familiarize themselves with the learning prior to and after the standard. [6]

[6] Ressa, V. (2014). Deconstructing Ohio's New Learning Standards. *Journal Ohio School Boards Association, 58*(2), 28-30. Retrieved from: https://static.battelleforkids.org/documents/the_ohio_standard/2_04-01-14_OSBA_Journal_%20April_Ressa.pdf

When deconstructing the computer science standards, teachers will utilize the Tennessee K-12 Computer Science Core Concepts. These six concepts contain essential knowledge for what students need to "know" to master the standard. Teachers will also utilize the K-12 Computer Science Framework Practices, which contain skills students will need to "do" to master the standard. **For more information on the K-12 Computer Science Framework Practices, please visit https://k12cs.org.**

TN K-12 Computer Science Core Concepts (What/Know)	K-12 Computer Science Framework Practices (How/Do)
1. Foundational Concepts 2. Algorithmic Thinking 3. Programming Concepts 4. Data and Analysis 5. Networking and the Internet 6. Impacts of Computing	1. Fostering an Inclusive Computing Culture 2. Collaborating Around Computing 3. Recognizing and Defining Computational Problems 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing

Deconstructing standards can be overwhelming the first time a teacher works through the process. It is recommended teachers collaborate with a team and utilize the Computer Science Standards Deconstruction Activity as a resource. After teachers have worked through this process, they will find that it becomes easier and faster.

It is best practice to revisit both the standards and the deconstruction process annually prior to instruction. What students need to "know" and "do" to master a standard should not change from one year to the next, however the resources teachers utilize to align to the standards may change. Experience from teaching the standards, student observation and feedback, as well as the availability of new resources and materials, will play a role in how teachers adjust their lessons annually to ensure alignment to the standards.

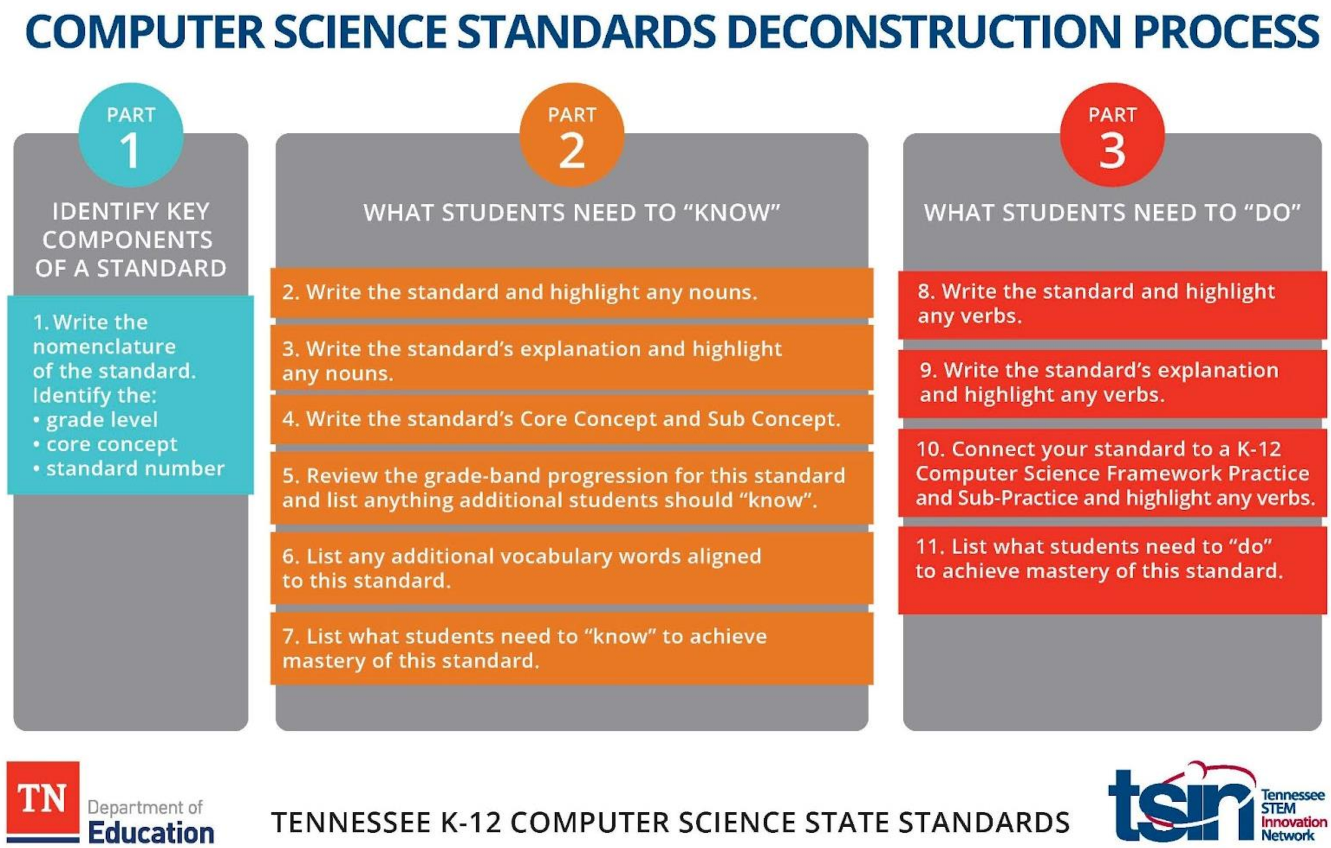

Figure 3. Quick Reference to Standard Deconstruction for the ***[Tennessee K-12 Computer Science Standards]***

Teachers are strongly encouraged to use the Computer Science Standards Deconstruction Activity as a tool to practice deconstructing the standards. The Computer Science Standards Deconstruction Activity can be found **here**.

Part D: Appendices

Appendix A: Resources

The following computer science education resources were compiled to complement this guide and to help teachers add tools to their computer science toolkit. It is not meant to be an exhaustive list. These resources support educators in their journey to network with computer science education professional organizations, access lessons, support implementation of programming, and deepen their computer science pedagogy.

Tennessee Department of Education (TDOE) and Tennessee STEM Innovation Network (TSIN) Computer Science Resources

Title	Link
Computer Science Standards Implementation Guidebook (online edition)	Click Here
Computer Science Core Concepts Progressions	Click Here
Computer Science Reference by Grade Level	Click Here
Computer Science Reference by Core Concept	Click Here
2024-2025 Computer Science Goals (11x17 poster)	Click Here
K-12 Tennessee Computer Science State Standards and Core Concepts (11x17 poster)	Click Here
Deconstruction Process (8.5x11)	Click Here
Deconstruction Process Activity PDF	Click Here
Deconstruction Process Activity Word	Click Here
Computer Science Short Courses	Click Here
Computer Science Endorsement Pathway	Click Here
TDOE Frequently Asked Questions	Click Here

Computer Science Professional Organizations

Title	Description	Link
Computer Science Teachers Association (CSTA)	CSTA supports K-12 computer science education by sharing best practices through professional learning, building computer science communities at the local and national level, and providing courses and materials that will take your teaching to the next level.	Click Here [CSTA] Click Here [TN Chapter]
International Society for Technology in Education (ISTE)	ISTE is home to a community of global educators who believe in the power of technology to transform teaching and learning, accelerate innovation, and solve tough problems in education. Multiple computer science teacher resources are available on their website.	Click Here
Tennessee STEM Innovation Network (TSIN)	TSIN's mission is to expand the teaching and learning of Science, Technology, Engineering, and Math education in K-12 schools across Tennessee. With the passing of legislation in support of advancing computer science	Click Here [TSIN] Click Here [TSIN Computer Science]

	education in the state of Tennessee, TSIN is partnering with TDOE to support educators with Computer Science.	

Online Computer Science Resources

Title	Description	Link	Grade Band
Birdbrain Technologies	Resources from Birdbrain Technologies include lesson plans, videos, and information on how to incorporate the Finch Robot into your classroom. The reading list contains fiction and nonfiction stories centered around computational thinking and computer science.	Click Here [Finch Robot Resources] Click Here [Reading List]	K-12
CSTA K-12 Standards	The K-12 Computer science standards from the Computer Science Teachers Association have been used to inspire other state standards. It has a searchable database with helpful descriptions.	Click Here	All
Code.org	Code.org is a great resource for teaching computer science (Code.org). They recently added a CS Connections section which has specific coding activities/lessons for integration.	Click Here [Code.org] Click Here [CS Connections]	K-12
CS First - Google	Google's computer science curriculum focuses on computer programming. The second link has computational thinking activities. Both have applications to other content areas.	Click Here [Programming] Click Here [Computational Thinking]	K-12
CSforAll	CSforAll is an organization devoted to bringing computer science education to all students. Their curriculum directory is especially helpful when identifying appropriate grade-band leveled lesson plans for your students.	Click Here	K-12
K-12 CS Framework	States, districts, and organizations can use the framework to inform the development of standards and curriculum, build capacity for teaching computer science, and implement computer science pathways. The framework is designed to guide computer science from a subject for the fortunate few to an opportunity for all.	Click Here	All

Microsoft: Guide to Inclusive Computer Science Education	This guide contains information about best practices for computer science education.	Click Here	K-12
Micro:Bit	This site has classroom resources, lesson plans, professional development and projects for use with the Micro:Bits.	Click Here	4-12

Publications

Title	Author / Publisher	Link	Grade Band
Big Book of Computing Pedagogy	Raspberry Pi Foundation	Click Here [Free Download]	9-12
Big Book of Computing Content	Raspberry Pi Foundation	Click Here [Free Download]	9-12
Computer Science in K-12 An A-Z Handbook on Teaching Programming	Shuchi Grover Edfinity	Click Here	K-12
Coding and the Arts	Josh Caldwell ISTE Publishing	Click Here	K-12
Computational Thinking {and Coding} for Every Student	Jane Krauss Kiki Prottsman Corwin Publishing	Click Here	K-12
Computational Thinking Meets Student Learning	Kiki Prottsman ISTE Publishing	Click Here	K-12
Creative Coding: Lessons and Strategies to Integrate Computer Science Across the 6-8 Curriculum	Josh Caldwell ISTE Publishing	Click Here	6-8
Everything You Need to Ace Computer Science and Coding in One Big Fat Notebook	Workman Publishing	Click Here	K-12

Appendix B: Glossary

The Computer Science Standards Implementation Guidebook Glossary serves as a curriculum resource tool for teachers and students. This Glossary is essential because "vocabulary is at the heart of oral language comprehension and sets the foundation for domain-specific knowledge and later reading comprehension."[7]

A critical component to effective vocabulary instruction is selecting appropriate terms for direct instruction, and that teaching terms in categories is an effective way to help students develop rich, contextual understanding of their meaning and use. [8] Therefore, each term and definition show its connection to one or more of the computer science concepts as indicated on the left of the table: **Foundational Concepts (FC)**; **Algorithmic Thinking (AT)**; **Programming Concepts (PC)**; **Data & Analysis (DA)**, **Networking & the Internet (NI)**; and **Impacts of Computing (IC)**. On the right side of the table, each term and definition are matched to the grade band where this word is explicitly connected to the standards and explanations in the Computer Science Standards Reference Document: **Early Elementary School (K-2)**; **Upper Elementary School (3-5)**; **Middle School (MS)**; and **High School (HS)**.

Vocabulary can be taught through direct and indirect instruction. Direct vocabulary instruction is when terms are explicitly identified for instruction and focused on with students. Indirect vocabulary instruction is less straightforward, requiring context clues, such as reading passages to students which contain the vocabulary, but not their definitions. [8] It is important to note that direct vocabulary instruction "must be more than merely identifying or labeling words. Rather, it should be about helping children to build word meaning and ideas that these words represent." [9]

To support teachers with direct vocabulary instruction, we suggest considering Marzano's six-step process when designing computer science lessons. The six steps include:

1. Provide a description, explanation, or example of the new term.
2. Ask students to restate the description, explanation, or example in their own words.
3. Ask students to construct a picture, symbol, or graphic representing the term or phrase.
4. Engage students periodically in activities that help them add to their knowledge of the terms in their vocabulary notebooks.
5. Periodically ask students to discuss the terms with one another.
6. Involve students periodically in games that allow them to play with the terms.

Each of the words listed in the glossary were intentionally selected because they are words students are less likely to encounter and learn on their own but will occur frequently in computer science.

---

[7] Marulis, L. M., & Neuman, S. B. (2010). The effects of vocabulary intervention on young children's word learning: A meta-analysis. *Review of Educational Research*, *80*(3), 300-335.

[8] Marzano, R. J., Rogers, K, & Simms, J. A. (2015). Vocabulary for the New Science Standards. Bloomington, IN: Marzano Resources.

[9] Neuman, S. B., & Dwyer, J. (2009). Missing in action: Vocabulary instruction in pre-K. *The Reading Teacher, 62*(5), 384-392.

These definitions were influenced by the Computer Science Teachers Association (CSTA) glossary. Please visit https://csteachers.org/k12standards/glossary/ for a complete list of their glossary resources.

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
✓		✓	✓			**abstraction**	The process of reducing complexity by focusing on the main idea. By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the problem.		✓	✓	✓
		✓				**analyze**	Examine in detail for the purposes of interpretation and explanation.		✓		
	✓	✓	✓			**algorithm**	A step-by-step process to complete a task.	✓	✓	✓	✓
✓				✓		**application**	A type of application software designed to run on a mobile device, such as a smartphone or tablet computer. Also known as a *mobile application.*	✓		✓	
					✓	**accessibility**	Appropriate measures to ensure that persons with disabilities access information and communications, on an equal basis with others, both in urban and rural areas.		✓		
				✓		**Application Programming Interface (API)**	A set of rules, protocols, and tools that allows two or more computers to communicate with each other. APIs define the methods and data formats that developers can use to request and exchange information or functionality between different software systems.				✓
	✓					**array**	A data structure comprising a collection of values of the same type accessible through an index. Data is of fixed size. *Example: [A, B, C, D] is an array of letters; the second element of the array is B.*				✓
	✓					**artifacts**	Anything created by a human. See also computational artifact.				✓
					✓	**artificial intelligence (AI)**	A wide range of techniques and technologies that enable computers and machines to perform tasks that typically require human intelligence. Examples			✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
							include visual perception, speech recognition, and decision making.				
					✓	**assistive technology**	Any device, software, or system that is used to increase, maintain, or improve functional capabilities of a person with a disability.		✓		
	✓	✓				**attribute**	A piece of information which determines the properties of a field or tag in a database or a string of characters in a display. *Example: The "color" attribute of a red car would have the value "red."*	✓		✓	✓
				✓		**attribution**	Assigning proper credit to a particular source, origin, or author.			✓	
					✓	**automation**	Controlling a process by automatic means, reducing human intervention to a minimum.				✓
				✓		**baiting**	A type of social engineering in the form of enticing behavior aimed at luring someone into a trap, confrontation, or harmful situation to reveal personal or financial information or introduce malware.			✓	
	✓		✓			**binary**	A method of encoding data using two symbols, 1 and 0. *Example: the number 4 written in binary is 100.*		✓	✓	
		✓				**block-based programming**	Drag and drop layout, where code fits together like puzzle pieces, making skilled typing and attention to syntax unnecessary.		✓		
		✓				**Boolean**	A type of data or expression with two possible values: true and false.				✓
		✓				**branches**	In programming, branches refer to when the decision-making process occurs due to an algorithm choosing between several different actions to take. *Example: when a decision is made with an If/Then statement.*				✓

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
							In software development, branches are separate lines of development that allow multiple versions of a software project to coexist. Developers can work on different branches to implement new features, debug, or experiment with changes without affecting the main or stable version of the software.				
		✓				**bug**	An error in a software program. It may cause a program to unexpectedly quit or behave in an unintended manner. The process of finding and correcting errors is called debugging.		✓		✓
			✓			**cause-and-effect relationship**	An action or event, or the *cause*, produces an outcome or result, or the *effect*.		✓		
			✓			**chart**	Information in the form of a table, graph, or diagram.	✓	✓		
	✓					**cipher**	An algorithm for encrypting or decrypting information.		✓		
				✓		**citation**	A reference in a body of work to a source.	✓			
		✓				**code**	Any set of instructions expressed in a programming language.			✓	✓
			✓			**coding**	The act of writing computer programs in a programming language.				✓
		✓				**comment**	A programmer-readable annotation in the code of a computer program, added to make the code easier to understand. Comments are generally ignored by machines.			✓	
	✓					**communicating about computing**	Communicating with diverse audiences about the use and effects of computation and the appropriateness of computational choices. This is one of the seven CS practices in the *K-12 CS Framework*. *Examples: writing clear comments, documenting work, communicating ideas using precise language and multiple forms of media.*		✓		

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
		✓				**compound conditionals**	Used in computer programming to combine two conditions into a single statement. These compound conditionals are often used in decision-making processes, such as in conditional statements (*if/then statements*) and loops.				✓
	✓					**computation**	The process of performing calculations, operations, or data processing using a computer, calculator, or any other device capable of executing mathematical or logical tasks.			✓	✓
✓						**computational artifact**	Anything created by a human using a computational thinking process and a computing device. A computational artifact can be, but is not limited to, a program, image, audio, video, presentation, or web page file.		✓		
			✓			**computational data**	Data sets stored, manipulated, or visualized using algorithms and data structures.			✓	
			✓			**computational model**	A representation of some part of a problem or a system using computer science or computational thinking.			✓	✓
✓		✓			✓	**computational thinking**	A thought process that revolves around solving a variety of problems. The four main components of computational thinking are algorithmic thinking, pattern recognition, decomposition, and abstraction. Computational thinking is at the heart of the computer science practices and is delineated by the practices in the *K-12 Computer Science Framework*: • Practice 3: Recognizing and Defining Computational Problems • Practice 4: Developing and Using Abstractions • Practice 5: Creating Computational Artifacts	✓	✓	✓	

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
							• Practice 6: Testing and Refining Computational Artifacts				
✓						**computer**	See computing device.	✓			
					✓	**computer science**	Computer science is a way to combine creative thinking and the power of computers to solve a wide variety of problems. This includes designing algorithms, hardware and software implementation, data analysis, and the impacts of computers on society.			✓	
					✓	**computing device**	A physical device that uses hardware and software to receive, process, and output information. *Examples: computers, mobile phones, and computer chips inside appliances.*		✓	✓	
			✓			**computing method**	A systematic way to perform tasks or solve problems using a computer. *Examples: algorithms and computation.*		✓		✓
			✓			**computing system**	A computing system can be limited to a single computer or computing device *however*, it more commonly refers to a collection of multiple connected computers, computing devices, and hardware.			✓	
	✓	✓				**conditional**	A feature of a programming language that performs different computations or actions depending on whether a programmer-specified Boolean condition evaluates to true or false. These include conditional statements, conditional expressions, or conditional constructs that help to control the flow of a program or make decisions. *Examples: If/Then or If/Else statements.*			✓	✓
				✓	✓	**connectivity**	A program or device's ability to link with other programs and devices.			✓	✓
		✓				**control structure**	A programming (code) structure that implements control.				✓

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
							Examples: conditionals and loops.				
				✓	✓	**copyright**	A form of intellectual property that protects the creators of original works of authorship. These exclusive rights generally include the right to reproduce, distribute, perform, display, and create derivative works based on the original work.	✓	✓	✓	✓
				✓		**copyright infringement**	The unauthorized use of copyrighted materials without permission of the copyright holder. This can lead to legal action, including injunctions and monetary damages.		✓		
				✓		**Creative Commons (CC)**	A licensing system that provides a legal framework for creators to grant the public permission to use their creative works while allowing them to retain certain rights and control over how their works are used.	✓			
				✓	✓	**cyberbullying**	The use of electronic communication to bully a person typically by sending messages of an intimidating or threatening nature.	✓	✓		
			✓	✓		**cybersecurity**	The protection against access to, or alteration of, computing resources through the use of technology, processes, and training.			✓	✓
	✓	✓	✓		✓	**data**	Information that is collected and used for reference or analysis. Data can be digital or nondigital and can be in many forms, including numbers, text, show of hands, images, sounds, or video.	✓	✓	✓	✓
			✓		✓	**data analysis**	Data analysis is a process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis includes identifying trends and making predictions or inferences.		✓		✓

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
			✓			**data cleaning**	The process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a data set.		✓	✓	
					✓	**data integrity**	The overall completeness, accuracy, and consistency of data.			✓	
			✓			**data set**	An ordered collection of data.		✓		✓
			✓			**data transformation**	The process of removing errors, highlighting, or exposing relationships, and/or making it easier for computers to process data.		✓		
	✓					**data type**	A classification of data that is distinguished by its attributes and the types of operations that can be performed on it. *Examples: integer, string, Boolean (true or false), and floating-point.*			✓	✓
			✓			**data visualization**	The representation of data through use of common graphics, such as charts, plots, infographics, and even animations. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.		✓		✓
			✓			**database**	An integrated and organized collection of logically related records or files or data that are stored in a computer system.		✓		
	✓	✓				**debug**	The process of finding and correcting errors (***bugs***) in programs.		✓	✓	✓
	✓					**decode**	The process of converting encoded or encrypted information back into its original, human-readable form.		✓		
	✓	✓	✓			**decomposition**	Breaking down a problem or system into components.		✓	✓	✓
	✓		✓		✓	**digital**	A characteristic of electronic technology that uses discrete values, generally 0 and 1, to generate, store, and process data.	✓	✓	✓	
✓				✓	✓	**digital citizenship**	The norms of appropriate, responsible behavior with regard to the use of technology.	✓	✓		

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
							Digital citizenship topics include instruction on media balance, privacy and security, cyberbullying, news and media literacy, and digital identity and footprint.				
				✓		**digital identity**	The body of information about an individual, organization, or device that exists online. Key components of digital identities may include personal information, usernames and IDs, authentication factors, digital certificates, digital footprints, authorizations, and permissions.				✓
				✓		**digital footprint**	A digital footprint, also known as a digital trail, refers to the digital records and traces that individuals, organizations, or devices leave behind as a result of their online activities.		✓		
					✓	**digital globalization**	The process by which digital technologies enable and accelerate the global exchange of information, goods, services, and culture.			✓	
✓						**digital model**	(*see* ***computational model***)		✓		
✓				✓		**digital tools**	Digital tools refer to computer programs, software applications, or online platforms that are designed to perform specific tasks or functions in a digital environment.	✓	✓		
	✓					**distribution**	The process of delivering software from a developer to the end user.				✓
✓						**domain**	An internet domain is a unique, human-readable address used to identify a specific website or resource on the internet.		✓		
				✓		**domain suffix**	A domain suffix, also known as a Top-Level Domain (TLD), is the last part of the domain name in an internet address and appears after the last dot. Examples include .com, .net, and .org.		✓		
	✓	✓				**efficiency of an algorithm**	The minimum number of resources, such as memory, time, or messages, needed to solve a problem or execute an algorithm.	✓		✓	
	✓		✓			**encode**	To assign a code to represent data.		✓	✓	

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
	✓		✓	✓		**encryption**	The conversion of electronic data into another form, called ciphertext, which cannot be easily understood by anyone except authorized parties.		✓	✓	
				✓		**end user**	A person for whom a hardware or software product is designed (as distinguished from the developers).				✓
		✓				**Engineering Design Process**	A problem-solving process that follows a series of steps and focuses on creating a functional product or process.				✓
				✓		**fair use**	Copyright laws include the doctrine of fair use, which allows limited use of copyrighted materials without permission from the copyright owner for purposes such as criticism, commentary, news reporting, education, and research.			✓	
	✓		✓			**flow chart**	A diagram of the sequence of steps or actions in a complex system or activity.	✓	✓		
		✓				**function**	A type of procedure or routine. Some programming languages make a distinction between a function, which returns a value, and a procedure, which performs some operation, but does not return a value.			✓	✓
			✓			**graph**	A pictorial representation or diagram that represents data or values in an organized manner.	✓			
					✓	**hacking**	Appropriately applying ingenuity or using a computer to gain unauthorized access to data within a system.				✓
✓					✓	**hardware**	The physical components that make up a computing system, computer, or computing device.	✓	✓	✓	
					✓	**human–computer interaction (HCI)**	The study of how people interact with computers and to what extent computing systems are or are not developed for successful interaction with human beings.				✓
✓						**icon**	A symbol or graphical representation of a file, folder, program, or application on the desktop or in file directories. Clicking	✓			

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
							directly on the icon usually opens the associated file or application.				
				✓		**impersonate**	Pretending to be someone else, typically for deceptive or fraudulent purposes.		✓		
					✓	**implications**	The potential consequences that result from a specific action, decision, event, statement, or situation.				✓
		✓				**information**	What is conveyed or represented by a particular arrangement or sequence of things.		✓		
					✓	**innovation**	Creating something new or improved that results in positive change.			✓	
✓	✓					**input**	The signals, data values, or instructions sent to a computer.	✓		✓	
✓						**input device**	Hardware accessory that receives signals or instructions sent to a computer. *Examples: keyboard, mouse, microphone, touchpad, touchscreen, sensors.*		✓		
					✓	**intellectual property laws**	A set of legal principles, regulations, and protection designed to safeguard the creation and innovations of individuals and entities. *Examples: patents, copyright laws, trademark laws*.				✓
✓				✓		**internet**	The global collection of computer networks and their connections, all using shared protocols to communicate.	✓	✓		
					✓	**internet censorship**	The practice of controlling, restricting, or limiting what can be accessed, published, or viewed on the internet by governments, organizations, or internet service providers.			✓	
				✓		**internet protocol (IP) address**	A unique numeric value assigned to a computer or other device connected to the Internet so that it may be identified and located.				✓
	✓	✓				**iterative**	The repeating of a process with the aim of approaching a desired goal, target, or result.		✓	✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
							*Example: Program development is an **iterative** process.*				
		✓				**iterative design**	Repeatedly refining and improving a product, system, or process through a series of cyclic iterations. *Example: Engineering Design Process*			✓	✓
✓						**keyboard**	A hardware input device that is used to enter text, characters, and commands into a computer.	✓			
✓				✓		**keyword**	A specific word or phrase that is essential in information retrieval and search engine optimization.		✓		
✓						**Learning Management System (LMS)**	A software application or platform designed to facilitate, deliver, and assess educational and training content, resources, and activities in an organized manner.		✓		
				✓		**liability**	The legal responsibility of an individual or entity to settle a debt, perform a duty, or provide compensation for any harm, loss, or damage they may have caused to another party.				✓
				✓		**licensing**	A legal arrangement that grants permission from one party (*the licensor*) to another party (*the licensee*) to use, possess, or access certain assets, intellectual property, or rights under specific conditions.			✓	
		✓				**libraries**	A collection of files, programs, routines, scripts, or functions that can be referenced in a computer program.				✓
	✓	✓				**list**	A data structure for storing ordered values.				✓
		✓				**logical operators**	A symbol or word used to connect two or more expressions such that the value of the compound expression produced depends only on that of the original expression and on the meaning of the operator.				✓

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
							Examples: AND, OR, NOT				
		✓				**loop**	A programming structure that repeats a sequence of instructions, as long as a specific condition is true. Infinite (*forever*) loops repeat the same steps endlessly, and it has no terminating condition. Count-controlled (*for*) loops repeat the same steps a specific number of times, regardless of the outcome. Condition-controlled (*while, for... while*) loops will keep repeating the steps over and over, until it gets a specific result.		✓		✓
				✓		**malware**	Any software specifically designed to harm, exploit, or compromise computer systems, networks, or user data.			✓	
✓				✓		**media**	Various forms of communication, information, and entertainment designed to reach and influence a wide variety of audiences.	✓	✓		
✓						**menu**	A structured and organized way for users to interact with a program's features, functions, and settings.	✓			
			✓			**Morse code**	A system of communication that's composed of combinations of short and long tones that represent letters of the alphabet. The tones are sometimes called dots and dashes.			✓	
		✓				**nested loop**	A loop which exists inside the body of another loop.				✓
				✓		**network**	A group of computing devices (*personal computers, phones, servers, switches, routers, etc.*) connected by cables or wireless media for the exchange of information and resources.			✓	
✓					✓	**online**	A device, system, or entity that is connected to or accessible via the internet or other telecommunications systems.	✓	✓		
				✓		**online etiquette**	A set of social conventions, guidelines, and norms governing behavior and interactions in the digital realm. Also referred to as "*netiquette*."	✓			

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
	✓					**operation**	An action that is carried out to accomplish a given task. Five basic types of computer operations are: inputting, processing, outputting, storing, and controlling. Arithmetic operations are addition, subtraction, multiplication, and division.			✓	
		✓				**output**	Any information that is processed by and sent out from a computing device. *Example: anything viewed on your computer monitor screen, such as the words you type on your keyboard.*				✓
		✓				**pair programming**	Collaborating with a partner(s) to develop and/or execute a program.		✓		
		✓				**parameter**	The name of a piece of information passed into a procedure to customize if for a specific need.			✓	✓
					✓	**patent**	A legal document that grants an inventor exclusive rights to an invention for a specific period of time.				✓
	✓		✓			**pattern recognition**	Identifying similarities or patterns across a problem or when evaluating data.	✓	✓		
				✓		**passwords**	A secret combination of characters, numbers, symbols, or words that individuals use to authenticate their identity and gain access to a computer system, online account, or digital resource.	✓		✓	
				✓	✓	**personal information**	Details that can be used to identify, describe, or contact an individual. Examples may include name, birthdate, gender, social security number, address, height, weight, financial account numbers, and health information.	✓		✓	✓
				✓		**phishing**	A malicious act where someone pretends to deceive individuals into revealing confidential information such as PIN or credit card numbers.			✓	
				✓		**piracy**	The illegal copying, distribution, or use of software.		✓		

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
				✓		**plagiarism**	Presenting someone else's work, ideas, words, or intellectual property as your own without giving credit to the original author or source.		✓		
					✓	**privacy**	The right of an individual to keep certain aspects of their personal life free from being observed or disturbed by others.				✓
				✓	✓	**private information**	*(See **personal information**.)*	✓		✓	
		✓				**problem**	A specific task, challenge, or puzzle, that requires a computational solution or algorithmic approach.		✓		
		✓				**procedure**	An independent code module that fulfills some concrete tasks and is referenced within a larger body of program code. The fundamental role of a procedure is to offer a single point of reference for some small goal or task that the developer or programmer can trigger by invoicing the procedure itself. Procedure is used as a general term that may refer to an actual procedure or a method, function, subroutine, or module of any other name by which modules are known in other programming languages. Some programming languages make a distinction between a function, which returns a value, and a procedure, which performs some operation, but does not return a value.			✓	
	✓					**process**	To perform a series of operations on a set of data.				✓
		✓				**program**	A set of instructions that the computer executes to achieve a particular objective. Instructions can be written in different programming languages. *Examples: HTML, CSS, Python, JavaScript, block-based.*			✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
		✓				**programming**	The craft of analyzing problems and designing, writing, testing, and maintaining programs to solve the problems.		✓		
	✓					**prototype**	An early approximation of a final product or information system, often built for demonstration purposes.				✓
		✓				**readme**	A file containing information about other files in a directory or archive of computer software.			✓	
				✓		**reboot**	The process of restarting a computer, device, or system to refresh its operating state and software.			✓	
				✓		**secure / non-secure**	The level of protection, safety, or confidentiality that a system or network possesses.			✓	
✓				✓		**search engine**	A software application or online service that allows users to search and retrieve information.		✓		
				✓		**secure operator**	A string of characters used in a search engine query to narrow the focus of the search. Three basic operators are AND, OR, and NOT.		✓		
				✓	✓	**security measures**	A set of practices in place to protect individuals from threats, risks, or unauthorized access.		✓		✓
	✓					**selection**	Using conditions to control the flow of a program. Algorithms can utilize selection to decide which set of actions to perform based on the evaluation of a ***Boolean*** expression.			✓	
	✓					**sequence**	sequence (noun): An ordered set of instructions. sequence (verb): To arrange instructions in a particular order.	✓	✓	✓	
			✓			**simulation**	Imitation of the operation of a real-world process or system over time.		✓		
			✓			**spreadsheet**	A computer application for computation, organization, analysis, and storage of data in tabular form.		✓		

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
				✓		**social engineering**	A deceptive technique used to trick individuals into making decisions that compromise security or privacy. Unlike hacking, which exploits technical vulnerabilities, social engineering exploits human psychology, trust and social interactions to gain unauthorized access.			✓	
					✓	**social media**	Platforms designed to enable individuals, groups, and organizations to connect, communicate and share content.		✓		
✓		✓			✓	**software**	Programs that run on a computing system, computer, or other computing device.	✓	✓	✓	✓
	✓					**strategy**	A high-level plan for solving a complex problem.	✓			
		✓				**syntax**	A set of rules that govern the structure, arrangement, and organization of elements within a programming language.				✓
✓						**toolbar**	A vertical or horizontal row of icons, buttons, or controls designed to provide the user with quick and convenient access to functions and features of a program enhancing the user experience.	✓			
					✓	**trademark**	Any recognizable word, phrase, symbol, design, or combination of these things that identifies your goods and services.				✓
✓		✓	✓			**troubleshooting**	A systematic approach to problem solving that is often used to find and resolve a problem, error, or fault within software or a computing system.	✓		✓	
				✓		**two-step authentication**	An extra layer of security to protect online accounts that requires users to provide two separate forms of authentication before granting access.			✓	
			✓			**Unicode**	An informational technology standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems.			✓	
✓	✓	✓				**unplugged activity**	An activity used for learning computer science or computational thinking and does not require a device.	✓	✓		

FC	AT	PC	DA	NI	IC	Term	Definition	K-2	3-5	MS	HS
				✓		**user tracking**	The practice of monitoring online activities of individuals as they interact with websites, applications, and digital services.				✓
✓						**window**	The framed area on a screen that displays the content of an application, document, or program.	✓			
	✓	✓				**variable**	A symbolic name that is used to keep track of a value that can change while a program is running. Variables are not just used for numbers; they can also store text, including whole sentences (*strings*) or logical values (*true or false*).		✓	✓	✓
				✓		**Virtual Private Network (VPN)**	A service that provides secure communication over a public network by creating an encrypted connection between the user's device and a remote server operated by the VPN service provider.				✓
					✓	**vulnerability**	A weakness, flaw, or gap in a computer system, software application, hardware device, or network that could compromise a system's security.				✓

Appendix C: Computer Science Resource Team

The Computer Science Standards Resource Team (CSSRT) was formed in the summer of 2023 for the purpose of creating a Standards Reference Document that would support each individual grade level standard by providing explanations and connections to concepts, sub-concepts, and vocabulary. Additionally, the resource team created the Computer Science Standards Progressions which provides K-2, 3-5, 6-8, and 9-12 learning progressions for each of the six core-concepts and their sub-concepts.

The thirty members of the Computer Science Standards Resource Team represent a variety of districts across the three grand divisions of Tennessee. Members of the CSSRT are listed below.

Foundational Concepts		
Name	**Position/Role**	**District**
Annemarie Lampright	District Leader / FC Team Lead	Lawrence County Schools
Michael Newman	Educator / FC Team Member	Henry County Schools
Katie Robertson	Technology Coach / FC Team Member	Murfreesboro City Schools
Kathryn Vaughn	Educator / FC Team Member	Tipton County Schools
Teresa Wise	Educator / FC Team Member	Rutherford County Schools

Algorithmic Thinking		
Name	**Position/Role**	**District**
Amanda Gray	Technology Coach / AT Team Lead	Clarksville-Montgomery County Schools
Kevin Ayers	Educator / AT Team Member	Roane County Schools
David Freeman	District Leader / AT Team Member	Jefferson County Schools
Joni Stone	Educator / AT Team Member	Wilson County Schools
Janet Williams	Educator / AT Team Member	Memphis-Shelby County Schools

Programming Concepts		
Name	**Position/Role**	**District**
Richard Hawkins	Educator / PC Team Lead	Hamblen County Schools
Michael Chaffin	Educator / PC Team Member	Putnam County Schools
Lakata Green	District Leader / PC Team Member	Memphis-Shelby County Schools
Bobbie Meredith	Educator / PC Team Member	Rutherford County Schools
Jeanie Phillips	Instructional Coach/ PC Team Member	Oak Ridge City Schools

Data & Analysis		
Name	**Position/Role**	**District**
Rebecca Smith	Educator / DA Team Lead	Private School
Michelle Bettis	Instructional Coach / DA Team Member	Hamilton County Schools
Claytarcia Holliday	Educator / DA Team Member	Memphis-Shelby County Schools
Crystal Lock	Educator / DA Team Member	Dyer County Schools
Logan Phillips	Educator / DA Team Member	White County Schools

Networking & the Internet		
Name	**Position/Role**	**District**
Jessica Holloway	Instructional Coach/ NI Team Lead	Hamilton County Schools
Lillian Grant	Instructional Coach / NI Team Member	Memphis-Shelby County Schools
Jessica Hernandez	District Leader / NI Team Member	University of Memphis
Nikki Russell	Administrator / NI Team Member	Hamilton County Schools
Chad Ward	Educator / NI Team Member	Clarksville-Montgomery County Schools

Impacts of Computing		
Name	**Position/Role**	**District**
Amy Haney	District Leader / NI Team Lead	Roane County Schools
Stephanie Cooper	Educator / NI Team Member	Wilson County Schools
Jerry Lynn Recker	Instructional Coach / NI Team Member	Memphis-Shelby County Schools
Shelby Woods	Educator / NI Team Member	Sevier County Schools
Bixiao Zhao	Educator / NI Team Member	Knox County Schools

Integration Guidebook Authors	
Karen Harper, Ed. D.	Tennessee STEM Innovation Network, Computer Science Professional Learning Consultant
Stephanie Zeiger, Ph. D.	Tennessee STEM Innovation Network, Computer Science Education Content Consultant
Integration Guidebook Editor	
Misty Brown	Tennessee STEM Innovation Network, Communications and Relationship Manager

Tennessee Department of Education	
Audra Block	Director of STEM/STEAM and Computer Science

Appendix D: References

[1] 2022 State of Computer Science Education: Understanding Our National Imperative. Retrieved, December 15, 2022 from https://advocacy.code.org/2022_state_of_cs.pdf

[2] Tennessee Department of Education. (2022, October 28). Computer Science Tennessee K-12 Computer Science State Standards. Retrieved January 5, 2023, from https://www.tn.gov/content/dam/tn/stateboardofeducation/documents/2022-sbe-meetings/october-28%2c-2022-sbe-meeting/10-28-22%20III%20A%20Computer%20Science%20Standards%20Framework%20for%20Grades%20K-12%20Clean.pdf

[3] Computer Science Teachers Association Glossary. Retrieved December 15, 2022 from https://www.csteachers.org/page/glossary

[4] Creative Commons. (n.d.). K–12 Computer Science Framework. k12cs.org. Retrieved January 5, 2023, from https://k12cs.org/

[5] Tennessee is Top State in US for Advanced Industry Job Growth, Brookings Institution Report Finds. (n.d.). www.tn.gov. Retrieved September 14, 2022, from https://www.tn.gov/ecd/news/2016/8/5/tennessee-is-top-state-in-us-for-advanced-industry-job-growth-brookings-ins.html

[6] Ressa, V. (2014). Deconstructing Ohio's New Learning Standards. *Journal Ohio School Boards Association, 58*(2), 28-30. Retrieved from: https://static.battelleforkids.org/documents/the_ohio_standard/2_04-01-14_OSBA_Journal_%20April_Ressa.pdf

[7] Marulis, L. M., & Neuman, S. B. (2010). The effects of vocabulary intervention on young children's word learning: A meta-analysis. *Review of Educational Research*, *80*(3), 300-335.

[8] Marzano, R. J., Rogers, K, & Simms, J. A. (2015). Vocabulary for the New Science Standards. Bloomington, IN: Marzano Resources.

[9] Neuman, S. B., & Dwyer, J. (2009). Missing in action: Vocabulary instruction in pre-K. *The Reading Teacher, 62*(5), 384-392.