Computer Science

Integration Guidebook

Tennessee Department of Education | February 2023

Contents

Part A: Introduction to Tennessee Computer Science Law 3

Public Chapter #979 of the Public Acts of 2022 3

Active Response to the Tennessee Law 4

Computer Science Endorsement Pathway (CSEP) 4

Tennessee K-12 Computer Science State Standards and Core Concepts 5

Computer Science Core Concepts 5

Reach Them All 9

Part B: How to Integrate Computer Science 10

Why Computer Science? 10

Computational Thinking 11

Integration 13

Computer Science Practices 14

Integration Process© 15

Model Lessons 19

Resource Kits 29

Additional Resources 31

Appendices 35

Appendix A: Glossary 35

Appendix B: Model Lesson Teacher Team 48

Appendix C: References 49

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 take action 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 will 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]

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

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.

Computer Science Endorsement Pathway (CSEP)

To ensure teachers are properly trained and licensed to deliver computer science courses in middle school and high school, the **Computer Science Endorsement Pathway (CSEP)*** was created. The CSEP provides teachers with a no-cost route to computer science endorsement. The endorsement pathway contains **six self-paced modules** that allow teachers to demonstrate their knowledge and skills of the Tennessee K-12 Computer Science State Standards by embedding these practices into their current instruction. Teachers who obtained an employment standard or completed Code.org training will only be required to successfully complete three of the six modules. For additional information about the CSEP, please visit https://www.computersciencetn.org/. For additional questions regarding the endorsement and your licensure, please email educator.licensure@tn.gov.

Please note: The Computer Science Endorsement Pathway (CSEP) is a **separate initiative from Reach Them All. Reach Them All will be expanded on later in this guidebook.*

Tennessee K-12 Computer Science State Standards and Core Concepts

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 the 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. To access the drafted proposal of the Tennessee K-12 Computer Science State Standards, please click **here**.

Computer Science Core Concepts

The Tennessee K-12 Computer Science Standards contain six core concepts, represented in the graphic below. To achieve this vision, the Standards are created in themes and terms to be seamlessly integrated into instruction and cognitive tasks in every grade level and subject area so students can see how the logical problem solving allows for enhancement of any topic. They are shown as similarly sized sections because they each support the other. One section cannot be removed without compromising the integrity of the structure. Therefore, by the time a student has completed high school in Tennessee, they should 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]

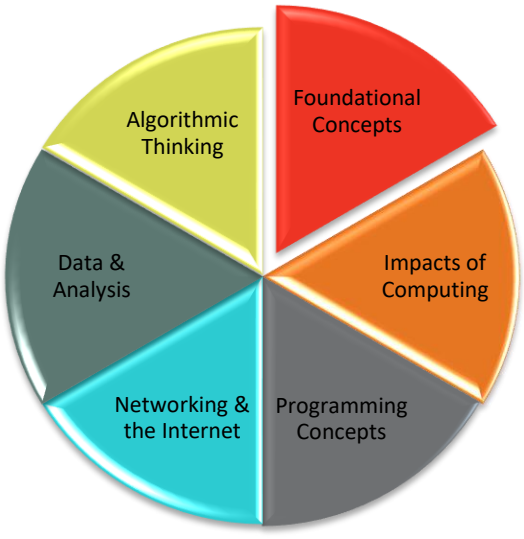


Figure 1. Six Core Concepts of Tennessee's CS K12 Standards

[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*

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 Integration 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.

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 pattern recognition, debugging, and abstraction.

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

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

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.

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.

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 which 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,

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

understanding the differences between secure and non-secure websites, and understanding what personal data is collected and shared across these networks.

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]

Computer science is one of the fastest growing industries and computer programmers are needed within every field, including healthcare, transportation, entertainment, and banking.[5] All students will benefit from learning computer science concepts and practices, allowing them to better understand the world around them, improve their logical reasoning and problem-solving skills, and increase their creativity and collaboration. Rather than being avocational part of education only for students interested in a computer science career, computer science is now a foundational part of Tennessee K-12 education that is accessible to all students.[2]

[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*

[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

Reach Them All

The Reach Them All initiative was launched in September 2022, in support of Computer Science Education Legislation (Chapter 979 of the Public Acts of 2022).

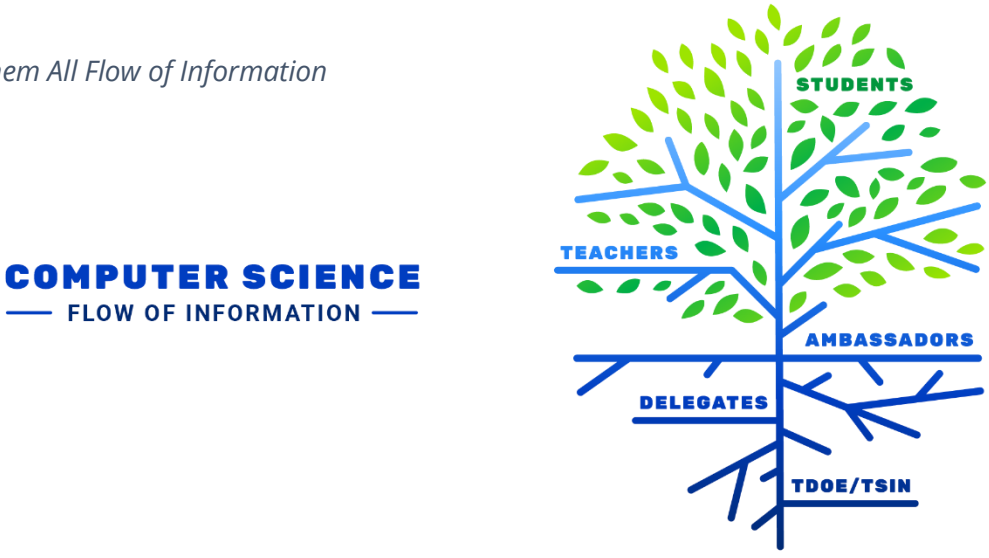
Reach Them All utilizes a train-the-trainer model that allows each district to select one **Computer Science District Delegate** that will serve as a liaison and support their district by offering professional development opportunities in computer science. Computer Science District Delegates recruit **Computer Science District Ambassadors** from within their district to join them in providing high-quality computer science support. The number of Ambassadors depends on the size of the district. The Ambassadors will attend the same meetings as Delegates and they will also be responsible for re-delivering the training to teachers and schools.

Computer Science District Delegates and Ambassadors will receive training from November 2022 to March 2023, and Ambassadors will re-deliver the training to schools within their district from April 2023 through September 2023. These training sessions will empower schools and teachers to promote integration of computer science into all Tennessee classrooms, understand new computer science legislation and expectations, and create a network of best practices regarding computer science statewide. This timeline gives schools one full year after training before state accountability measures take effect.

Each school participating in the Reach Them All training provided by their District Ambassadors will receive a grade-band specific kit with materials and resources that will complement their professional learning. To access a list of Tennessee Computer Science District Delegates, please click here. For additional information please email tsin@battelle.org.

Once Tennessee K-12 teachers begin integrating computer science and computational thinking practices into their lessons, we recommend that districts focus on computer science standard implementation. This will require a deep dive into standard analysis and deconstruction, crosswalks between computer science standards and contents, exploration or resources, and access to quality curriculum and computer science courses.

Figure 2. Reach Them All Flow of Information



Part B: How to Integrate Computer Science

Why Computer Science?

Computer science is one of the fastest growing industries today, and it impacts a variety of sectors including healthcare, transportation, entertainment, and banking.[5] More than **7.7 million Americans** use computer science at their jobs, with close to half of them in Science, Technology, Engineering, and Math (STEM) fields[4] outside of computer science, but where computation has become fundamental to innovation and even to everyday practices in the field. In fact, being an engaged and active citizen in this digital world increasingly requires an understanding of computer science and the skills behind it. All students will benefit from learning computer science concepts and practices to allow them to better understand and critically reflect on the world around them, improve their logical reasoning and problem-solving skills, and increase their creativity and collaboration. Rather than being a vocational part of education only for those headed to a specific computer science field, computer science is a foundational part of Tennessee K-12 education.[2]

Computer science is defined by the **Association of Computing Machinery** as "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their implementation, and their impact on society."[6] Thus, **computer science is not** technology integration, being a 1:1 school, or simply using a computing device. It is also more than writing code or programming. It is a creative endeavor that combines a variety of skills - including logical reasoning, collaboration, problem solving and designing with empathy - with the power of computing. It can range from designing an app to locating the ideal restaurant to applying computational biology to understand the complexities of the human body. Thus, computer science represents a way for students to express their ideas and apply their creativity to solve a variety of different problems.

---

[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

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

[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*

[6] *Tucker, Allen, (editor), Deek, F., Jones, J., McCowan, D., Stephenson, C., and Verno, A. A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee. Association for Computing Machinery (ACM), New York, New York, October, 2003. (2nd ed., 2006.)*

Computational Thinking

To build a foundation of computer science pedagogy, teachers need to gain confidence in their understanding of computer science and be able to see how it connects with their individual content area. Computational thinking serves as the bridge between content and computer science.

Figure 3: Connecting Computer Science to Core-Content

Introduced as procedural thinking by Papert in the 1960s and later expanded and popularized by J.M. Wing in 2006 in the publication *Communications of ACM*, the definition of computational thinking has evolved throughout the years by researchers and educators.[7] Across all definitions, though, it is a strategic thought process that combines skills and practices fundamental to computer science, which all K-12 students can learn. Moreover, its application moves beyond computer science and can be a powerful tool for many content areas.

While there are several different components to computational thinking, the four main components we will focus on in this integration guide include decomposition, pattern recognition, algorithms, and abstraction (Figure 4). **Decomposition** allows students to break complex problems into more manageable steps while **pattern recognition** helps students identify similarities and differences within a

[7] Wing, J. M. (2006, March). *Computational Thinking*. CMU School of Computer Science. Retrieved January 6, 2023, from https://www.cs.cmu.edu/

problem or across data. Combining these with **algorithmic thinking** (step-by-step process) and **abstraction** (focusing on the problem versus all the details) can lead to powerful ways to approach complex problems with or without a computer. In 2011, an ISTE and CSTA collaboration led to further defining computational thinking in the following ways:[8]

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

Ideally, teachers and students should be able to see how computational thinking connects to content and skills already being developed in the classroom setting. It is a relatable entry point for teachers as they consider how to further bring computer science into the K-12 classroom and to help students develop systems of thinking around solving complex problems. While computational thinking helps to develop the skills that underpin computer science, educators need to be intentional with highlighting how these skills relate to computer science even in elementary grades. You will see these connections as we further explore how to integrate computer science or computational thinking into content classes within this Guidebook.

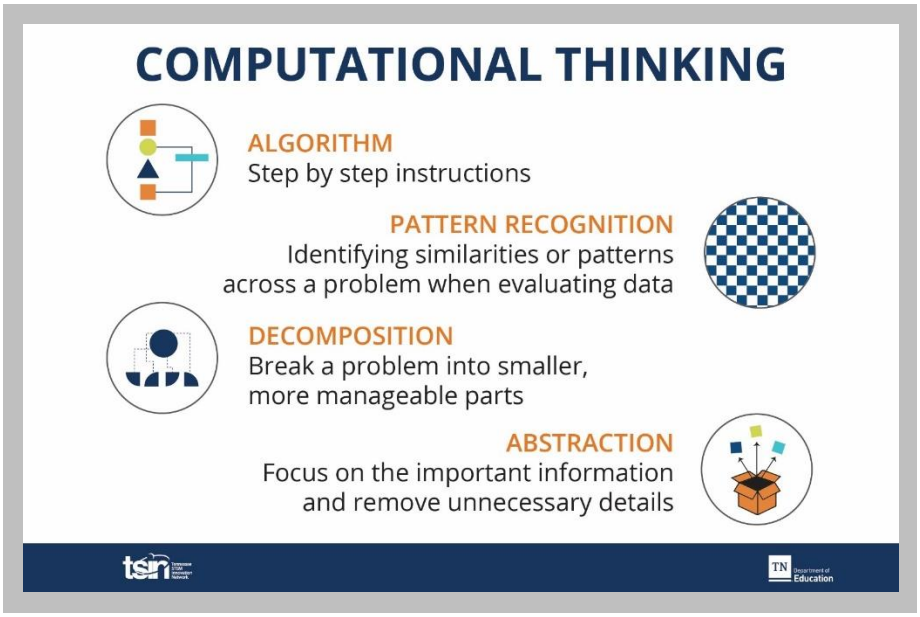


Figure 4: Four Main Components of Computational Thinking

[8]International Society for Technology in Education. (2011). *Operational Definition of Computational Thinking for K-12 Education*. www.iste.org. Retrieved January 6, 2023, from https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf

Integration

Computer Science is a foundational component necessary for all TN students to be successful as future contributing citizens to our state and its economy and community. All efforts must be grounded in equitable access to all students regardless of grade level or geographic location which means that our efforts must equip all teachers in all schools, not just one grade band, content area or geographic grand division of our state.

Tennessee educators will provide all students with access to computer science education from kindergarten through high school graduation. The focus on integration will create a framework to ensure the state teaching force has the capacity, disposition, and sustainability to provide accessible and high-quality computer science education.

The purpose of this section of the integration guide is to demonstrate how content teachers can weave computational thinking and computer science practices into their daily lessons. With an empathetic approach to the workload of educators, we want to demonstrate with this resource that teaching computer science is not “one more thing” teachers are being asked to do, but rather a way to complement the teaching of their content lessons with the pedagogy of computer science. Often teachers are already integrating computational thinking and computer science practices in their lessons, and the different components in this integration guide will help teachers to recognize those connections and integrate with confidence and efficacy.

The main components in this section include:

1. **Computer Science Practices** from the K-12 Computer Science Framework.
2. A **process** for how to integrate computational thinking and computer science into content.
3. **Model lessons** that demonstrate how computational thinking and computer science practices have been integrated into content lessons, along with supporting materials in a resource kit.
4. **Additional resources** listed for teachers to access computer science professional learning, quality lessons, and tools for integration.

By integrating computational thinking and computer science practices into lessons, teachers can actively play a part in supporting the momentum of computer science education in the state of Tennessee. Although the law specifies required computer science courses, these courses alone do not fully address the need for our students to be creative, critical participants in society and leaders in computationally intensive careers.

Through the Reach Them All initiative, you will be able to leverage the collective wisdom of those around you including your **Computer Science District Delegates and Ambassadors**. As we move forward from our trainings on integration and begin implementing what we have learned, we ask all Tennessee educators to keep the following goals in mind:

2023-2024 Goals for Computer Science Integration

1. Teachers develop a foundational understanding of computational thinking and computer science concepts.
2. Teachers discover connections between what they teach, and computational thinking and computer science concepts. Teachers use these connections to integrate computational thinking and computer science concepts into their curriculum.
3. Teachers cultivate a mindset that expects all students to participate in computational thinking and computer science.

Computer Science Practices

The K-12 Computer Science Framework is a national document created by The Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative as a high level framework for K-12 computer science education and identifies core concepts and practices of computer science. Additionally, the K-12 Computer Science Framework serves as a guide to states, districts, and organizations who are implementing computer science education.[4]

Tennessee's K-12 Computer Science Standards, currently under consideration by the State Board of Education, were strongly influenced by this framework and readers may recognize several of the practices and skills that were incorporated into the TN standards document. The K-12 Framework states, "The seven core practices of computer science describe the behaviors and ways of thinking that computationally literate students use to fully engage in today's data-rich and interconnected world."[4] The practices are heavily influenced by computational thinking, and this is apparent in practices 3-6.
The seven **K-12 Computer Science Framework practices** are:

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

The K-12 Computer Science Framework also demonstrates how its seven practices relate to practices in other contents. This overlap in practices with other subject areas affirms the ability and ease to integrate computational thinking and computer science into content lessons. For additional information please visit the K-12 Computer Science Framework at https://k12cs.org/.

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

Integration Process©

Below is an **algorithm**, a step-by-step process to consider as you begin integrating **computer science (CS)** and/or **computational thinking (CT)** into your lessons. This process is based on the experiences of the model lesson teacher team, recommendations from the International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA) resources, and the iterative design process. Like most new things, when you begin, you may need to return to resources and videos to help you feel confident in the process. The model lessons in this Integration Guidebook are a great place to start.

Remember, the integration of computational thinking or computer science is a process that takes time and practice. Be patient with yourself and collaborate with other teachers as you begin this process. Collaboration with colleagues, including brainstorming, will be integral to successful computer science integration in your school.

Based on your preferences and needs, you can begin the process by identifying a lesson you already teach in your classroom or finding connections to your curriculum in a brainstorming session. Like most iterative design processes, this is not meant to be linear progression, as you should find yourself moving back and forth between steps depending on your needs *(Figure 5).*

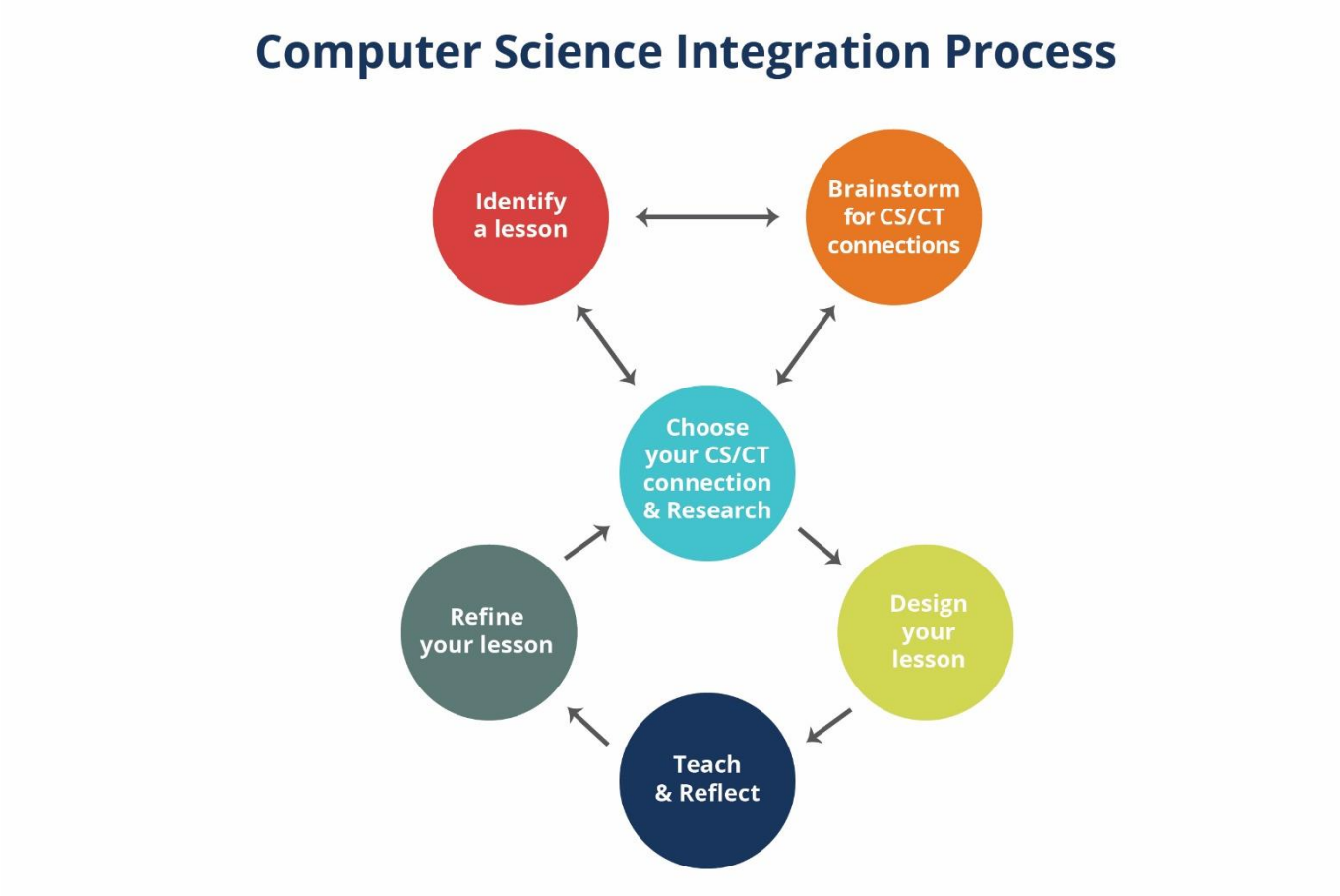


Figure 5: Computer Science Integration Process©

Identify a Lesson

As you begin this process, start with a lesson you are familiar with and already teach rather than creating a new lesson. Frequently, you will find that by intentionally including computer science vocabulary or thinking into the activities you do with a computer science or computational thinking lens that your lesson only needs slight modifications. Remember the following are best practices to ensure your lesson fits with your content standards and practices.

- Select your content and grade level standard. Consider any specific practices for your content that can be supportive in teaching that standard.
- Deconstruct the standard by asking yourself, "What do students need to know and do to achieve mastery of that standard?".
- Review the curriculum and resources you might use to teach this standard.

Brainstorm Computer Science or Computational Thinking Connections

As you consider where there are natural connections between your current lesson(s) and computer science, take a moment to brainstorm. This is a great time to collaborate with fellow teachers. The following checklists for Computational Thinking Connections and Tennessee Computer Science Concept Connections are meant to be prompts and are not an exhaustive list. You may notice similarities between the lists as computational thinking is important for computer science education. Choose the list you are most comfortable with as you begin this process.

Computational Thinking Connections

- **Decomposition** – Does the problem or task assigned to students require to break it into smaller, more manageable parts?
- **Pattern Recognition** - Will my students be finding similarities or differences across something? This might include numbers, words, pictures, a process, or a problem.
- **Algorithmic Thinking** - Am I asking my students to think in a step-by-step way or tackle a problem in a logical way? Consider anything that asks you to do one thing before you can do another.
- **Abstraction** - Will my students be reducing the complexity of something by focusing on a main idea? Do students need to generalize a concept, model, or process?

TN Computer Science Concept Connections

Foundational Concepts

- Can my students use computational thinking in this lesson?
- Are students using hardware or software in this lesson? Help students move beyond just using the hardware/software and consider how it might be helpful for learning, ways to improve it, or deciding the best option for creating something or representing an idea.
- Are students collaborating and working as a team to develop a **computational artifact** (anything created by a human using a computational thinking process and a computing device) or seeking user input when designing something?

Algorithmic Thinking

- Are students using logical sequencing in this lesson or do I ask students to be detailed in solving a problem or task?

- Will a flowchart or decision matrix be helpful for their learning? Students may use **conditional statements** which help in making choices. Example: an if/ then statement
- Can students reflect on another person's **algorithm**, step-by-step process, to make it more efficient or help solve another problem?

Programming Concepts

- Once students design an **algorithm**, how can they use it to make something happen?
- Is there a coding or unplugged activity that will complement my lesson?
- Can students extend or enhance their learning of a concept by programming a physical device?

Data and Analysis

- Are students collecting, organizing, or using data in this lesson?
- Is there an opportunity for students to use a computational model to understand a concept?
- Will data visualization help students discover relationships or connections for understanding?

Networking and the Internet

- Can the topic of **networking**, **internet**, or **cybersecurity** be used for your lesson? This can include researching, essay writing, or debating around these topics and their real-world impacts.
- Is there an opportunity for students to practice or consider appropriate cybersecurity measures in this lesson? Examples: Secure passwords, backing up data, using firewalls, https sites
- Consider moments in the lesson of how to teach students how to troubleshoot devices or when having connectivity issues. This is a powerful opportunity to build problem-solving skills and perseverance.

Impacts of Computer Science

- How has computer science impacted your content area? How did people understand this concept before and after a new computing technology was introduced?
- Is there an industry or career students can relate to that uses these computer science skills? Move beyond just a computer when considering this concept. Think about the many different fields that use computer science to solve problems. Examples: computational biology, data mining, digital art, communications, or even fashion
- What are the real-world connections to computer science in your lesson?

Choose a Computer Science or Computational Thinking Connection & Research

This step asks you to focus on one connection you made during your iterative or brainstorming process. We highly recommend you choose one or two connections at the most when beginning to integrate these concepts. If too many are selected, your lesson may be confusing to students and become overly complicated. As you further build your background knowledge of computer science and computational thinking, you will begin to identify more connections. At this point, further research about the computer science connection you have chosen, and your lesson content area may be required before you begin a lesson design. **Resources, including a glossary, are available in Appendix A.**

- Have you chosen one or two connections? Keep it simple.
- What resources are available to help you further understand this computer science or computational thinking concept?
- How can your students relate to the chosen connection? Are there things in their lives that are impacted by computer science? Think of the technology they use and what matters to them.
- What computer science vocabulary words would be important for this connection?
- Do you need to learn more about available hardware, software, or coding resources?

Design Your Lesson

You may use the **Computer Science Model Lesson Template** or adapt your current lesson template as you begin to design your lesson. Below are a few things to consider as you plan.

- Don't forget to stay focused on your content standards. You might want to take a moment to go back and review the main focus of your lesson. The goal is for computer science or computational thinking integration to **support** your student's learning of the content standard rather than your content supporting the computer science.
- What content and computer science vocabulary will my students need to be successful in this lesson?
- How will I differentiate this lesson to make it equitable for all students?
- How will I assess learning, and will the computer science integration be a part of that assessment?
- How will I ensure all students have access to the computational thinking and computer science practices in this lesson?

Teach and Reflect

This is your opportunity to try your lesson in the classroom. As you begin the integration process, it is essential that you reflect on your lesson after implementation and make adjustments where necessary to refine the lesson. You might ask students for feedback in a short survey as they may have additional insights you have not considered. Make sure to reflect not only on the student experience, but your own experience as well.

- What worked well in the lesson?
- Does the pacing need any modification?
- Was there enough scaffolding for students to learn and use the new concepts and skills?
- Were the connections between the content and the computer science skills clear to students? How could you tell?
- What areas of the lessons were you less confident about?

Refine Your Lesson

Make modifications to the lesson for next time. Remember to use those reflection notes and reach out to a fellow teacher for support. You may find yourself having to go back to other steps in this process as you redesign including at the beginning with brainstorming. Here are some things to consider.

- How can you strengthen the computer science or computational thinking connection?
- Do you need to reach out to a computer science teacher or do further research?

- Is there an industry or career connection to be made for this lesson? Consider allowing students to meet someone doing this in their job either virtually or in person. There may also be videos through various organizations for students to learn more.
- Is there a way to emphasize the computer science/computational thinking content from this lesson to other lessons in your curriculum or across a project?

Model Lessons

The goal for providing model lessons in this integration guide is to provide teachers with examples of how computational thinking and computer science practices integrate into their curriculum and enhance the teaching of Tennessee Academic Standards. The model lessons provided demonstrate a balance between **plugged** (needing a computing device) and **unplugged** (not needing a computing device) activities. Many of the strategies you will see demonstrate the use of computational thinking and can be applied across contents. We hope you are inspired from the model lessons to make meaningful computational thinking and computer science connections with your own curriculum.

Elementary School Model Lessons Access: https://bit.ly/3IvYMcG					
Lesson Name	**Content**	**Grade Level**	**Content Standard**	**Computer Science & Computational Thinking**	**Overview**
Pattern Recognition and Rhyming	English Language Arts	K	**K.FFL.PA.2** Demonstrate understanding of spoken words, syllables, and sounds (phonemes). a. Recognize and begin to produce rhyming words.	• Pattern Recognition • Algorithmic Thinking	Using Margaret Wise Brown's book *Goodnight Moon*, students will use pattern recognition to identify rhyming words in a story and algorithmic thinking to explain their step-by-step process
Algorithms in the Design Process	English Language Arts and Social Studies	1	**1.26** Identify and describe the events or people celebrated during the following national holidays, and examine why we celebrate them: Martin Luther King, Jr. Day, Columbus Day, Presidents' Day, Veterans' Day, Memorial Day, Thanksgiving Day, and Independence Day. **1.RI.RRTC.10** With prompting and support, read informational texts of appropriate complexity for grade 1. **1.SL.CC.2** Ask and answer questions about key details in a text read aloud or information presented orally or through other media.	• Algorithms	The students will listen to the text Balloons Over Broadway written by Melissa Sweet. This story tells the true story of the origins of the Macy's Thanksgiving Day Parade and information about the puppeteer who designed the first balloons. Students will write a step-by-step process to create a balloon for the parade.

Lesson Name	Content	Grade Level	Content Standard	Computer Science & Computational Thinking	Overview
Classifying Polygons and Quadrilaterals	Math	3	**3.G.A.1** Understand that shapes in different categories may share attributes and that the shared attributes can define a larger category. Recognize rhombuses, rectangles, and squares as examples of quadrilaterals and draw examples of quadrilaterals that do not belong to any of these subcategories.	• Algorithms • Flow Charts • Conditionals	Students will determine if a shape is a polygon, and students will apply their knowledge of a **flowchart** to determine the correct order of thinking when analyzing the attributes of various quadrilaterals.
Finding the Area of Rectilinear Decomposition	Math	3	**3.MD.C.7.d** Recognize area as additive. Find areas of rectilinear figures by decomposing them into non-overlapping rectangles and adding the areas of the non-overlapping parts, applying this technique to solve real-world problems.	• Decomposition • Algorithmic Thinking	Students will break apart and/or decompose rectilinear figures into two rectangles. They will find the area of each rectangle and add them to find the total area of the original figure. **(3.MD.C.7d)**
The Case of Cause and Effect	English Language Arts and Science	3	**3.RI.KID.3** Describe the relationship between a series of historical events, scientific ideas or concepts, or steps in technical procedures in a text, using language that pertains to time, sequence, and cause/effect. **3.RI.KID.1** Ask and answer questions to demonstrate understanding of a text, referring explicitly to the text as a basis for the answers. **3.LS4.1** Explain the cause and effect relationship between a naturally changing environment and an organism's ability to survive and change.	• Conditionals	Students will use cause and effect statements to create **conditionals** which include if-then statements.
Cardinal Directions and Algorithms	Social Studies	3	**3.02** Use cardinal directions, intermediate directions, map scales, legends, and grids to locate major cities in Tennessee and the U.S. **3.03** Examine major physical features on globes and maps, including: Basin, Bay, Canal, Canyon, Delta, Desert, Gulf, Island, Isthmus, Mountain, Ocean, Peninsula, Plain, Plateau, River, Sea, Strait, Stream, and Valley.	• Algorithms • Pair-programing • Debugging	Students will write an **algorithm** incorporating cardinal directions to move students on a grid to a specific destination. Students will use **pair-programming** when they test, compare, and debug their algorithms.

Lesson Name	Content	Grade Level	Content Standard	Computer Science & Computational Thinking	Overview
Wall Art and Algorithmic Thinking	Visual Art	4	**4.VA.Cr2.A** Explore and invent art-making techniques and approaches using developmentally appropriate craftsmanship. **4.VA.R1.B** Compare responses to a work of art before and after experimenting with similar processes.	• Algorithmic Thinking • Developing and Using Abstractions	In this lesson, students will explore Sol LeWitt's Wall Art in which he created the art by writing a list of instructions that were later executed in museums and galleries by curators. This form of artmaking parallels the creation of **algorithms** by computer scientists.
Understanding Earth's Seasons Using Patterns and Models	Science	5	**5.ESS1.5** Relate the tilt of the Earth's axis, as it revolves around the sun, to the varying intensities of sunlight at different latitudes. Evaluate how this causes changes in day-lengths and seasons.	• Pattern Recognition • Computational Modeling	In this lesson students identify **patterns** within data from an investigation and **computational models** to determine which season Earth is experiencing. Students will be able to explain how the tilt of the Earth causes seasons.

Middle School Model Lessons Access: https://bit.ly/3QkbQnB					
Lesson Name	**Content**	**Grade Level**	**Content Standard**	**Computer Science & Computational Thinking**	**Overview**
Exponents and Pattern Recognition	Math	6-7	**6.EE.A.2** Write, read, and evaluate expressions in which variables stand for numbers. a. Write expressions that record operations with numbers and with variables. c. Evaluate expressions at specific values of their variables. Include expressions that arise from formulas used in real-world problems. Perform arithmetic operations, including those involving whole number exponents, in the conventional order when there are no parentheses to specify a particular order (Order of Operations). **6.EE.C.9** Use variables to represent two quantities in a real-world problem that change in relationship to one another. a. Write an equation to express one quantity, thought of as the dependent variable, in terms of the other quantity, thought of as the independent variable. **7.EE.B.4** Use variables to represent quantities in a real-world or mathematical problem and construct simple equations and inequalities to solve problems by reasoning about the quantities. **AM.A.PS.A.1** Apply problem solving strategies to real-world situations. Strategies include, but are not limited to: making orderly lists or tables, drawing diagrams, considering simpler problems, looking for patterns, working backwards, guess and check, using logical reasoning, etc.	• Pattern Recognition • Algorithms • Abstraction	Students will use **pattern Recognition**, **algorithmic thinking**, and **abstraction** to develop an understanding of exponents.

Lesson Name	Content	Grade Level	Content Standard	Computer Science & Computational Thinking	Overview
Algorithms and Music	Music	6-8	**General Music Standards** **6.GN.R3.A** Apply teacher-provided criteria to evaluate musical works or performances. **7.GM.R3.A** Select from teacher-provided criteria to evaluate musical works or performances. **8.GM.R3.A** Apply appropriate personally-developed criteria to evaluate musical works or performances. **8.GM.R1.B** Compare how the elements of music and expressive qualities relate to the structure within programs of music. **Band Standards** **6/7.IMR1.A** Identify and justify reasons for selecting music based on characteristics found in music, context, and student interest. **8.IM.R3.A** Identify and justify musical preferences using appropriate vocabulary, context, student opinion, and personal research gathered from varied sources. **Choir Standards** **6/7.VM.R1.A** Identify and justify reasons for selecting music based on characteristics found in music, context, and student interest. **8.VM.R3.A** Identify and justify musical preferences using appropriate vocabulary, context, student opinion, and personal research gathered from varied sources.	• Algorithms • Conditional Statements • Impacts of Computer Science	Based on their understanding of musical elements, students will write **algorithms** to sort music into playlists using **conditional statements**. They will make real world connections for how algorithms are used by them daily.

Lesson Name	Content	Grade Level	Content Standard	Computer Science & Computational Thinking	Overview
Data Mining and Civilization Traits	Social Studies	6-7	**Social Studies** **6.04 Identify and explain the importance of the following key characteristics of civilizations:** • Culture • Stable food supply • Government • Technology • Religion • Writing • Social structure **English/Language Arts** **6.RI.CS.4** Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings. **6.RI.IKI.7** Integrate information presented in different media or formats, such as in tables, images, diagrams, and words to develop a coherent understanding of a topic or issue. **6.RI.KID.2** Determine a central idea of a text and how it is conveyed through details; provide an objective summary. **6.RI.RRTC.10** Read and comprehend a variety of literary nonfiction throughout the grades 6-8 text complexity band proficiently, with a gradual release of scaffolding at the high end as needed.	• Data Analysis • Pattern Recognition • Impacts of Computer Science	Using **data analysis** of a text, students focus on organizing and looking for **patterns** within **data** to determine important concepts around civilizations.
Catching the Big Bass with Data Analysis	Science	6	**6.LS2.1** Evaluate and communicate the impact of environmental variables on population size.	• Decomposition • Algorithmic Thinking • Data Analysis	Students **decompose** a problem, collect and **analyze** water quality **data**, and create an **algorithm** to determine the best environment for a fish species.

Lesson Name	Content	Grade Level	Content Standard	Computer Science & Computational Thinking	Overview
Using Computational Thinking to Write an Argumentative Essay	English Language Arts	7	**7.7.W.TTP.1** Writing arguments to support claims with clear reasons and relevant evidence.	• Decomposition • Pattern Recognition • Algorithmic Thinking	Students will learn to write an argumentative essay with a focus on breaking the essay into more manageable parts (**decomposition**), **identifying patterns** within the essay, and recognizing this is a way to use **algorithmic thinking** in a new context.
Computational Thinking and Evaluating a Crime Scene	STEM	6	**Science & Engineering Practice** • Asking Questions • Obtaining, Evaluating, & Communicating Information **6.L.VAU.6** Acquire and accurately use grade-appropriate general academic and domain specific words and phrases; develop vocabulary knowledge when considering a word or phrase important to comprehension or expression. **6.RI.KID.1** Analyze what a text says explicitly and draw logical inferences. **6.RI.KID.2** Determine a central idea of a text and how it is conveyed through details; provide an objective summary.	• Algorithms • Abstraction	Students will practice **abstraction** as they determine the important information needed to solve a crime.

High School Model Lessons Access: https://bit.ly/3ImeOWC					
Lesson Name	**Content**	**Grade Level**	**Content Standard**	**Computer Science & Computational Thinking**	**Overview**
Computational Modeling and Gas Laws	Science - Chemistry 1	9-11	**CHEM1.PS1** Matter and Its Interactions (5) Conduct investigations to explore and characterize the behavior of gasses (pressure, volume, temperature), develop models to represent this behavior, and construct arguments to explain this behavior. Evaluate the relationship (qualitatively and quantitatively) at STP between pressure and volume (Boyle's law), temperature and volume (Charles's law), temperature and pressure (Gay-Lussac law), and moles and volume (Avogadro's law), and evaluate and explain these relationships with respect to kinetic-molecular theory. Be able to understand, establish, and predict the relationships between volume, temperature, and pressure using combined gas law both qualitatively and quantitatively.	• Algorithms • Flow Charts • Conditionals	Combining **pattern recognition** and **data analysis** with a **computational model**, students will explore the relationships between temperature, pressure, and volume to understand the Gas Laws.
Patterns and the Periodic Table	Science - Chemistry 1	9-11	**CHEM1.PS1.12** Explain the origin and organization of the Periodic Table. Predict chemical and physical properties of main group elements (reactivity, number of subatomic particles, ion charge, ionization energy, atomic radius, and electronegativity) based on location on the periodic table. Construct an argument to describe how the quantum mechanical model of the atom (e.g., patterns of valence and inner electrons) defines periodic properties. Use the periodic table to draw Lewis dot structures and show understanding of orbital notations through drawing and interpreting graphical representations (i.e., arrows representing electrons in an orbital).	• Pattern Recognition • Algorithmic Thinking • Impacts of Computer Science	Using **pattern recognition** and **algorithmic thinking**, students will develop an understanding of the arrangement of the periodic table of elements.

Lesson Name	Content	Grade Level	Content Standard	Computer Science & Computational Thinking	Overview
Patterns, Algorithms, and the New Deal	Social Studies – U.S. History	11	**US.43** Analyze the impact of the relief, recovery, and reform efforts of President Franklin D. Roosevelt's New Deal programs, including: • Agricultural Adjustment Act • Civilian Conservation Corps • Securities and Exchange Commission • Fair Labor Standards Act • Social Security • Federal Deposit Insurance Corporation • Tennessee Valley Authority • Works Progress Administration National Recovery Administration	• Pattern Recognition • Algorithmic Thinking	Students will use **pattern recognition** and **algorithmic thinking** to analyze President Franklin D. Roosevelt's New Deal programs.
Analysis of Informational Texts with Algorithms	English Language Arts	9-10	**9-10.RI.KID.1** Analyze what a text says explicitly and draw inferences; cite the strongest, most compelling textual evidence to support conclusions. **9-10.RI.IKI.7** Evaluate the topic, subject, and/or theme in two diverse formats or media. **9-10.W.TTP.2** Write informative/ explanatory texts to analyze and convey complex ideas, concepts, and information clearly and accurately through the effective selection and organization of content. **9-10.W.RBPK.9** Support and defend interpretations, analysis, reflections, or research with evidence found in literature or informational texts, applying grade band 9-10 standards for reading and source material.	• Algorithms • Impacts of Computer Science	Students will examine how **algorithms** impact what they view on social media. By using a variety of sources, they will present their understanding and write an informational summative response.

Lesson Name	Content	Grade Level	Content Standard	Computer Science & Computational Thinking	Overview
Using Computational Thinking to Understand Geometric Sequences	Math – Algebra I	9-12	**A2.F.BF.A.1** Write a function that describes a relationship between two quantities. **A2.F.BF.A.2** Define sequences as functions, including recursive definitions, whose domain is a subset of the integers. Write explicit and recursive formulas for arithmetic and geometric. Sequences in context and connect them to linear and exponential functions.	• Pattern Recognition • Algorithmic Thinking	Through **pattern recognition** and designing **algorithms**, students will recognize both explicit and recursive formulas for geometric sequences. They will use **abstraction** to write the explicit and recursive formulas.

Resource Kits

Reach Them All resource kits were created to accompany the Model Lessons found in this **Integration Guidebook** and on the **Tennessee Computer Science Website**. All kits contain grade level appropriate devices that can be used to teach programming and enhance the Model Lessons. Each school that receives Reach Them All training from their Computer Science District Ambassadors will receive the appropriate grade-band kit. The content of each kit is listed below.

Grade Band	Item	Quantity
Elementary School Kit	Integration Guide	1
	2023-2024 Computer Science Goals for Integration Poster (11x17)	1
	Computational Thinking Poster (11x17)	1
	Computational Thinking- Bridging Core Content/Subjects to Computer Science Poster (11x17)	1
	Bee-Bots	2
	Blue-Bots	2
	Pro-Bots	2
	Command Card Decks	4
	Bee-Bot Lesson License for 1 Year	1
	Pro-Bot Lesson License 1 Year	1
	Vernier Probe- Go Temp	1
	Book: Balloons Over Broadway	1
	Book: Goodnight Moon	1
	Book: The Case of the Vanishing Little Brown Bats	1
	15 cm X 15 cm White Cardstock Template	1

Grade Band	Item	Quantity
Middle School Kit	Integration Guide	1
	2023-2024 Computer Science Goals for Integration Poster (11x17)	1
	Computational Thinking Poster (11x17)	1
	Computational Thinking- Bridging Core Content/Subjects to Computer Science Poster (11x17)	1
	Finch Robots Classroom Set (contains 5 Finch robots, 5 USB cables, USB charging hub, 5 markers, 5 micro:bits, caring case)	1
High School Kit	Integration Guide	1
	2023-2024 Computer Science Goals for Integration Poster (11x17)	1
	Computational Thinking Poster (11x17)	1
	Computational Thinking- Bridging Core Content/Subjects to Computer Science Poster (11x17)	1
	Finch Robots Classroom Set (contains 5 Finch robots, 5 USB cables, USB charging hub, 5 markers, 5 micro:bits, caring case) (an additional Finch robot is included in only the High School Kit)	1
	Micro:bits	5
	Franklin D. Roosevelt New Deal Graphic	1

Additional Resources

The following computer science education resources were compiled to compliment this integration 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 integrated lessons, support implementation of programming, and deepen their computer science pedagogy.

Computer Science Professional Organizations

Organization	Description	Website 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.	www.csteachers.org Tennessee Chapter https://tennessee.csteachers.org/
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.	www.iste.org
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 education in the state of Tennessee, TSIN is partnering with TDOE to support educators with the Computer Science Endorsement Pathway (CSEP), integration through Reach Them All's initiative, computer science professional learning, and resources and tools to support the newly adopted computer science standards.	www.tsin.org https://www.computersciencetn.org/

Online Computer Science and Computational Thinking Educational Resources

Name	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 for Finch Robot Resources Click Here for Reading List	K-12

Name	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 for Finch Robot Resources Click Here for Reading List	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	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	K-12
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 for Code.org main site Click Here for 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 for Programming Click Here for Computational Thinking	K-12
CTLessons.org	These computational thinking lessons were created by a teacher for a variety of subjects. This resource can be used to INSPIRE you as you consider how you might be using computational thinking in your classroom. This resource has "unplugged" activities that do not require a computing device.	Click Here	K-10
Micro:Bit	This site has classroom resources, lesson plans, professional development and projects for use with the Micro:Bits.	Click Here	4-12
No Fear Coding	Heidi Williams, author of No Fear Coding: Computational Thinking Across the K-5 Curriculum, offers a variety of free resources and materials that accompany her book and support the implementation of Bee and Blue-Bot.	Click Here	K-5

Name	Description	Link	Grade Band
Terrapin: Tools for Thinking	Terrapin resources that support the Bee-Bot, Blue-Bot, and Pro-Bot. Use discount code, **TENNAccessories**, for 20% off Bee-Bot, Blue-Bot and Pro-Bot accessories. Bee-Bot and Pro-Bot Online Resources and Emulator Bee-Bot Lessons (1 year subscription) Discount Code at Checkout: TENNBEELESSONS Pro-Bot Lessons (1 year subscription) Discount Code at Checkout: TENNPROLESSONS	Click Here Click Here Click Here Click Here	K-5

Publications

Title	Author/Publisher	Link	Grade Band
Big Book of Computing Pedagogy	Raspberry Pi Foundation	Free Download-Click Here	9-12
Big Book of Computing Content	Raspberry Pi Foundation	Free Download-Click Here	9-12
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
No Fear Coding: Computational Thinking Across the K-5 Curriculum	Heidi Williams ISTE Publishing	Click Here	K-5

Tennessee Department of Education (TDOE) and Tennessee STEM Innovation Network (TSIN)
Reach Them All Resources

Title/Item	Link
Computer Science Integration Guidebook	Click Here
2023-2024 Computer Science Goals for Integration Poster (11x17)	Click Here
Computational Thinking Poster (11x17)	Click Here
Computational Thinking Bridging Core Content to Computer Science-Poster (11x17)	Click Here
Reach Them All Training Resources	Click Here

Additional resources will be available on the TSIN website as they are developed.

Appendices

Appendix A: Glossary

A strong vocabulary forms the foundation for developing knowledge and skills in computer science. The following computer science terms are not all-inclusive. These vocabulary words were selected and emphasized in this guide due to their strong connection to integration. These terms are utilized in the computer science model lessons and show how computer science and computational thinking integrate into all content.

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)**. While many of these words will fit into multiple categories given the overlap of these concepts, categories were selected for where each word may be introduced or emphasized.

On the right side of the table, each term and definition are matched to the grade band you would most often see this word utilized: **Elementary School (ES)**; **Middle School (MS)**; and **High School (HS)**.

To access computer science vocabulary broken down by concepts, please click here. This vocabulary list was heavily influenced by the Computer Science Teachers Association (CSTA) vocabulary glossary. Please visit https://www.csteachers.org/page/glossary for a complete list of their glossary resources.

FC	AT	PC	DA	NI	IC	Term	Definition	ES	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.	✓	✓	✓
✓	✓					**algorithm**	A step-by-step process to complete a task.	✓	✓	✓
			✓			**analog**	The defining characteristic of data that is represented in a continuous, physical way. Whereas digital data is a set of individual symbols, analog data is stored in physical media, such as the surface grooves on a vinyl record, the magnetic tape of a VCR cassette, or other non-digital media.		✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
✓						**app**	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.	✓	✓	✓
			✓			**array**	A data structure comprising a collection of values of the same type accessible through an index. Data are of fixed size. For example, [A, B, C, D] is an array of letters; the second element of the array is B.	✓	✓	✓
					✓	**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.”	✓	✓	✓
				✓		**authentication**	The verification of the identity of a person or process.	✓	✓	✓
		✓				**binary**	A method of encoding data using two symbols, 1 and 0. Example: the number 4 written in binary is 100.	✓	✓	✓
		✓				**binary number**	A number written in the Base-2 Number System.	✓	✓	✓
		✓				**Boolean**	A type of data or expression with two possible values: true and false.		✓	✓
✓						**central processing unit (CPU)**	The device within a computer that executes instructions.		✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
			✓	✓		**cipher**	An algorithm for encrypting or decrypting information.	✓	✓	✓
✓		✓				**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.	✓	✓	✓
✓						**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 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. Four main components of computational thinking include 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	ES	MS	HS
							• Practice 6. Testing and Refining Computational Artifacts			
✓						**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. Computers, mobile phones, and computer chips inside appliances are all examples of computing devices.	✓	✓	✓
✓						**computing system**	Although a computing system can be limited to a single computer or computing device, 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.	✓	✓	✓
✓						**configuration**	The specific hardware and software details that tell exactly what the system is made up of, especially in terms of devices attached, capacity, or capability.		✓	✓
				✓		**connectivity**	A program or device's ability to link with other programs and devices.		✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
		✓				**control**	The use of elements of programming code to direct which actions take place and the order in which they take place.		✓	✓
		✓				**control structure**	A programming (code) structure that implements control. Conditionals and loops are examples of control structures.		✓	✓
				✓		**cyberbullying**	The use of electronic communication to bully a person typically by sending messages of an intimidating or threatening nature.	✓	✓	✓
				✓		**cyber harassment**	The use of the Internet or other electronic means to harass an individual, a group, or an organization.	✓	✓	✓
				✓		**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.	✓	✓	✓
			✓			**data integrity**	The overall completeness, accuracy, and consistency of data.		✓	✓
			✓			**data structure**	A particular way to store and organize data within a computer program to suit a specific purpose so that it can be accessed and worked with in appropriate ways. Examples of data structures include		✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
							arrays, queues, linked lists, trees, and graphs.			
			✓			**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. Some common data types are integer, string, Boolean (true or false), and floating-point.		✓	✓
			✓			**database**	An integrated and organized collection of logically related records or files or data that are stored in a computer system.	✓	✓	✓
		✓				**debugging**	The process of finding and correcting errors (bugs) in programs.	✓	✓	✓
✓						**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. Digital citizenship topics include instruction on media balance, privacy and security, cyberbullying, news and media literacy, and digital identity and footprint.	✓	✓	✓
	✓	✓				**efficiency of an algorithm**	The minimum amount 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.		✓	✓
			✓			**encryption**	The conversion of electronic data into another form, called ciphertext,		✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
							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).	✓	✓	✓
		✓				**execute**	execute: To carry out (or "run") an instruction or set of instructions (program, app, etc.).		✓	✓
	✓					**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.	✓	✓	✓
✓						**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.	✓	✓	✓
			✓			**index**	A common method for keeping track of data so that it can be accessed quickly. Like an index in a book, it is a list in which each entry contains the name of the item and its location. However, computer-based indexes may point to a physical location on a disk or to a logical location that points elsewhere to the actual location.			✓
✓						**input**	The signals, data values, or instructions sent to a computer.	✓	✓	✓
✓						**input device**	Hardware accessory that receives signals or instructions sent to a	✓	✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
							computer. Examples include keyboard, mouse, microphone, touchpad, touchscreen, and sensors.			
				✓		**internet**	The global collection of computer networks and their connections, all using shared protocols to communicate.	✓	✓	✓
				✓		**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. Example: Program development is an iterative process.	✓	✓	✓
		✓	✓			**list**	A data structure for storing ordered values.			✓
	✓	✓				**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.	✓	✓	✓
✓						**memory**	The physical storage space in computing devices, where data is to be processed and instructions required for processing are stored. Types of memory are RAM (Random Access Memory), ROM (Read Only Memory), and secondary storage	✓	✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
							such as a hard drive, a removable drive, and cloud storage.			
				✓		**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.	✓	✓	✓
				✓		**networking device**	Hardware units that help with connections on a network so that they can share data or resources. Examples include hubs, switches, routers, and bridges		✓	✓
✓						**operating system (OS)**	A set of programs that manage the functioning of, and other programs' access to hardware.	✓	✓	✓
✓						**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. An example of output is anything viewed on your computer monitor screen, such as the words you type on your keyboard.	✓	✓	✓
				✓		**packet**	The unit of data sent over a network.		✓	✓
		✓				**pair programming**	Collaborating with a partner(s) to develop and/or execute a program.	✓	✓	✓
✓						**pattern recognition**	Identifying similarities or patterns across a problem or when evaluating data.	✓	✓	✓
✓						**pixel**	The smallest controllable element of a picture/display.	✓	✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
							Also known as picture element.			
			✓			**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 include HTML, CSS, Python, JavaScript, and Block based.	✓	✓	✓
		✓				**programming**	The craft of analyzing problems and designing, writing, testing, and maintaining programs to solve the problems.	✓	✓	✓
				✓		**protocol**	A set of rules or procedures for transmitting data between electronic devices, such as computers. In order for computers to exchange information, there must be a preexisting agreement as to how the information will be structured and how each side will send and receive it.	✓	✓	✓
✓						**prototype**	An early approximation of a final product or information system, often built for demonstration purposes.	✓	✓	✓
	✓	✓				**pseudocode**	An informal high-level description of the operating principle of a computer program or other algorithm.	✓	✓	✓
✓						**redundancy**	A system design in which a component is duplicated, so if it fails, there will be a backup.		✓	✓
✓						**reliability**	An attribute of any system that consistently produces the same results, preferably meeting or exceeding its requirements.	✓	✓	✓
		✓				**repetition**	The process of repeating a task a set number of times or until a condition is met.	✓	✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
				✓		**router**	A device that connects networks to one another. Routers determine the path that data packets travel from source to destination.		✓	✓
				✓		**scalability**	The capability of a network to handle a growing amount of work or its potential to be enlarged to accommodate that growth.	✓	✓	✓
	✓	✓				**sequence**	sequence (noun): An ordered set of instructions. sequence (verb): To arrange instructions in a particular order.	✓	✓	✓
				✓		**server**	A computer or program dedicated to a particular set of tasks that provides services to other computers or programs on a network.		✓	✓
✓						**simulation**	Imitation of the operation of a real-world process or system over time.	✓	✓	✓
✓						**software**	Programs that run on a computing system, computer, or other computing device.	✓	✓	✓
		✓				**source code**	Programming statements and instructions written by a computer programmer. Source code is what a programmer writes, but it is not directly executable by the computer.		✓	✓
		✓				**statement**	A descriptive phrase that generates one or more instructions in a computer program.	✓	✓	✓
✓			✓			**store**	A process through which digital data is saved within a data storage device by means of computing technology. Storage is a mechanism that enables a computer to retain data, either temporarily or permanently.	✓	✓	✓
✓			✓			**storage**	A place, usually a device, into which data can be entered, in which the data can be held, and from which	✓	✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
							the data can be retrieved at a later time.			
		✓	✓			**string**	A sequence of letters, numbers, and/or other symbols. A string might represent, for example, a name, address, or song title. Some functions commonly associated with strings are length, concatenation, and substring.		✓	✓
		✓	✓			**structure**	A data type used to represent information about something more complicated than a single number, character, ~~or bool~~46oolean can do (and more complicated than an array of the above data types can do).		✓	✓
				✓		**switch**	A device that connects devices on a computer network by receiving, processing, and forwarding data to the destination device.		✓	✓
		✓	✓			**tag**	A field that identifies the contents of a data record.		✓	✓
✓						**test case**	A set of conditions or variables under which a person will determine whether the system satisfies requirements or works correctly.	✓	✓	✓
✓						**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.	✓	✓	✓
✓						**unplugged activity**	An activity used for learning computer science or computational thinking and does not require a device.	✓	✓	✓
		✓	✓			**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,	✓	✓	✓

FC	AT	PC	DA	NI	IC	Term	Definition	ES	MS	HS
							including whole sentences (strings) or logical values (true or false).			

Appendix B: Model Lesson Teacher Team

Name	**Position/Role**	**District**
Glenn Arnold	Educator	Knox County Schools
Nathan Bailey	Educator	Kingsport City Schools
Lea Bartch	District Leader	Murfreesboro City Schools
Andrea Burnette	Educator	Sevier County Schools
Cassie Cate	Educator	Sevier County Schools
Jessica Chambers	Educator	Sevier County Schools
Amanda Gray	Instructional Coach	Clarksville Montgomery County Schools
Alexis Hall	Post-Secondary	University of Tennessee, Knoxville
Amy Haney	District Leader	Roane County Schools
Brynne James	Instructional Coach	Clarksville Montgomery County Schools
Shelby Johnson	Educator	Sumner County Schools
Latisha King	Educator	Arlington Community Schools
Nikki Morgan	District Leader	Bartlett City Schools
Samantha O'Connor	Educator	Private School
Katie Robertson	Instructional Coach	Clarksville Montgomery County Schools
Becky Smith	Educator	Private School
Pamela Spinelli	Educator	Williamson County Schools
Kattie Stevens	Educator	Putnam County Schools
Jack Witt	Educator	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 C: 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] Tucker, Allen, (editor), Deek, F., Jones, J., McCowan, D., Stephenson, C., and Verno, A. A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee. Association for Computing Machinery (ACM), New York, New York, October, 2003. (2nd ed., 2006.)

[7] Wing, J. M. (2006, March). Computational Thinking. CMU School of Computer Science. Retrieved January 6, 2023, from https://www.cs.cmu.edu/

[8] International Society for Technology in Education. (2011). Operational Definition of Computational Thinking for K-12 Education. www.iste.org. Retrieved January 6, 2023, from https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf