



VIRGINIA DEPARTMENT OF
EDUCATION

Copyright © 2025

Virginia Department of Education
P.O. Box 2120
Richmond, Virginia 23218-2120
<http://www.doe.virginia.gov>

All rights reserved. Reproduction of these materials for instructional purposes in public school classrooms in Virginia is permitted.

Superintendent of Public Instruction

Emily Anne Gullickson

Assistant Superintendent of Teaching and Learning

Michelle Wallace, Ph.D.

Office of Educational Technology and Classroom Innovation

Calypso Gilstrap, Associate Director
Keisha Tennessee, Computer Science Coordinator

NOTICE

The Virginia Department of Education does not unlawfully discriminate on the basis of race, color, sex, national origin, age, or disability in employment or in its educational programs or services.

Table of Contents

2024 Computer Science <i>Standards of Learning</i>	3
Introduction	3
Foundational Principles	3
First Grade: 2024 Computer Science <i>Standards of Learning</i>	5
Algorithms and Programming (AP)	5
Computing Systems (CSY)	6
Cybersecurity (CYB)	6
Data and Analysis (DA)	6
Impacts of Computing (IC)	7
Networks and the Internet (NI)	7
Computer Science Instructional Guide Framework	9
Grade 1: Computer Science Instructional Guide	10
Algorithms and Programming (AP)	10
Computing Systems (CSY)	26
Cybersecurity (CYB)	36
Data and Analysis (DA)	44
Impacts of Computing (IC)	53
Networks and the Internet (NI)	68
Appendix A	72
K-5 Computer Science Skills and Practices Continuum	72
Appendix B	80
Grade 1 Computer Science Vocabulary	80

2024 COMPUTER SCIENCE *STANDARDS OF LEARNING*

Introduction

Virginia’s Computer Science *Standards of Learning* aim to raise our aspirations for computational instruction to enable students to engage and thrive in a digital world. Beginning in the earliest grades and continuing through 12th grade, students must develop a foundation of computer science knowledge and learn new approaches to problem solving that harness the power of computational thinking to become both users and creators of computing technology.

It is important for every student to engage in computer science education from the earliest ages. This early and sustained access equips students with foundational problem-solving practices, develops their understanding of how current and emerging computer science technologies work, and fosters curiosity, interest, and innovation with computer science.

Foundational Principles

Computer Literacy is foundational to learning and post-secondary success as technology becomes increasingly incorporated into all aspects of everyday life. Computer Literacy provides critical knowledge and skills for all subject areas including mathematics, science, history, English, and fine arts. By applying computer science as a tool for learning and expression in a variety of disciplines and interests, students will actively and proficiently participate in a world that is increasingly influenced by digital technology.

Computer Science fosters problem solving skills that are essential to all educational disciplines and post-secondary employment opportunities. Understanding how multi-step solutions are executed within computer programs allows students the opportunity to use metacognitive strategies with tasks they are performing as they work and study in any topic area. Computer Science should become an essential part of Virginia K-12 education, accessible by all, rather than a vocational part of education only for those headed to technology-based employment.

Computer Science instruction must maintain the pace of technology evolution to prepare students for the workforce. Computer science is a core technology component for students to have the ability to adapt to the future evolution of work. The workforce of the future will increasingly require that all adults effectively work in digital environments and utilize technology both ethically and responsibly. As a result, we must prioritize preparing all students with integral computer science learning opportunities throughout their academic career to ensure they are prepared for a post-secondary success in a digital world that includes computer-based problem solving, artificial intelligence and communication rooted in the use of digital tools.

Students should gain specific digital and computational concepts to harness the power of computer science and derivative applications, such as machine learning, online programming, virtual reality, and Artificial Intelligence (AI), to embrace innovation and chart the future of individuals, business, and government responsibly.

Instructional Intent and Integration

Computer science is an academic discipline that encompasses both conceptual foundations and applied practices. It can be taught effectively with or without computing devices, as many key skills, such as logical reasoning, pattern recognition, decomposition, and sequencing can be developed through with or without a computing device.

In primary grades, overlapping concepts between computer science and other content areas may be taught within the same instructional context. When doing so, it is essential that educators intentionally align instruction to ensure that the full intent and specifications of the computer science standard are addressed, even when the learning experience is shared with another content area.

As students' progress into upper elementary and beyond, instruction should be explicit, ensuring students are able to identify and understand the computer science concepts and practices embedded within those shared experiences. By naming the connections and explicitly stating the domain specific elements of computer science, students can deepen their disciplinary understanding, build metacognitive awareness, and transfer their knowledge and skills across contexts.

It is important to recognize that not all computer science concepts will naturally overlap with other subjects. Concepts such as algorithms, data representation, networks, and programming require dedicated instructional time and may be taught independently of other content areas. Whether through integration or stand-alone instruction, computer science should be approached with the same level of intentionality and rigor as other academic subjects, ensuring students develop a coherent and comprehensive understanding from kindergarten through grade 12.

Disclaimer: The Virginia Department of Education (VDOE) does not endorse or recommend any commercial products, services, or platforms. Any trademarks, logos, or images displayed in this instructional guide are used solely for educational and illustrative purposes to support conceptual understanding. Their inclusion does not constitute an endorsement by the VDOE of the referenced products, services, companies, or organizations.

First Grade: 2024 Computer Science *Standards of Learning*

In First Grade, students learn how computing components work together to form systems for collecting and sharing data/information. Students continue to develop problem-solving skills that apply computational thinking and can be applied with addressing hardware and software issues. Students refine computational thinking practices through activities such as planning, document creation, and construction of programs that consist of events and sequences. Students will further their knowledge on the importance of simple safeguards to protect private information and apply responsible behaviors and proper use of computing devices.

Algorithms and Programming (AP)

1.AP.1 The student will apply computational thinking by sorting items into categories based on multiple attributes and create patterns.

- a. Describe attributes of a set of objects.
- b. List the attributes a set of objects have in common.
- c. Sort and classify concrete objects based on multiple attributes.
- d. Create repeating and increasing patterns.

1.AP.2 The student will plan and implement algorithms that include the use of sequence and an event based on a predetermined task.

- a. Plan and create a design document that illustrates thoughts, ideas, and stories in a sequential manner.
- b. Construct step-by-step instructions that include decision-making and repetition.
- c. Identify and explain the role of events that are used in an algorithm.
- d. Test algorithms that are sequential and contain an event.

1.AP.3 The student will use the iterative design process to construct, test, and debug algorithms that include sequencing and an event.

- a. Discuss and describe the concept of debugging.
- b. Analyze and explain the results of an algorithm.
- c. Revise and improve an algorithm to produce desired outcomes.

Computing Systems (CSY)

1.CSY.1 The student will describe how computing components work together to create a computing system.

- a. Identify and define hardware, software, and computing systems.
- b. Identify common components of computing systems in different types of computing devices.
- c. Describe how hardware and software work together to form a computing system.

1.CSY.2 The student will use accurate terminology to describe when a computing system might not work as expected.

- a. Identify and describe a problem with a device or computing system when it does not work as expected.
- b. Propose a solution to simple hardware or software issues.

Cybersecurity (CYB)

1.CYB.1 The student will demonstrate safe and responsible use of computing technologies.

- a. Describe safe and responsible uses of computing technologies based on the school rules and acceptable use policy (AUP).
- b. Demonstrate safe and responsible behaviors when using computing technologies and online communication.
- c. Discuss the process for reporting inappropriate technology use at school or home.
- d. Classify appropriate and inappropriate uses of technology at school or at home.
- e. Explain the consequences of inappropriate uses of computing technologies.

1.CYB.2 The student will discuss the importance of using a password to protect private information.

- a. Describe the purpose of usernames and passwords.
- b. Discuss how passwords are private information and are used to protect the privacy of information.

Data and Analysis (DA)

1.DA.1 The student will explore how data can be stored and retrieved from various computing devices.

- a. Identify data formats used for various purposes, including audio, images, text, and video.
- b. Explore and identify computing devices that collect, store, and/or display data.

1.DA.2 The student will create representations of data to make predictions and draw conclusions.

- a. Collect and organize data with or without a computing device.
- b. Create tables, object graphs, picture graphs, and models using abstraction.
- c. Identify patterns and describe trends in data visualizations of various formats.
- d. Use data to answer questions, draw conclusions, and make predictions.

Impacts of Computing (IC)

1.IC.1 The student will describe how computing technologies impact daily tasks and communication.

- a. Determine when tasks should be completed with or without computing devices.
- b. Describe how computing devices are used in communication.
- c. Describe healthy habits for using computing technologies.

1.IC.2 The student will describe tasks and activities that use screens and categorize them based on their use.

- a. Identify daily routines and activities that can be completed with or without screens.
- b. Classify the different uses of screen time as learning, entertainment, or communication.

1.IC.3 The student will compare and contrast ways people complete tasks with and without computing technologies.

- a. Identify tasks that can be completed with and without computing technologies.
- b. Discuss advantages and disadvantages of using and not using computing technologies.
- c. Describe how the appropriate use of computing technologies can improve efficiency.
- d. List computing technologies used in various careers.

Networks and the Internet (NI)

1.NI.1 The student will explain that computing devices and the use of the Internet allow people the ability to gather information and connect with others.

- a. Describe how the Internet can be used to gather information.
- b. Explain ways people communicate using computing devices and the Internet.

Computer Science



Instructional Guide

This instructional guide, a companion document to the 2024 Computer Science *Standards of Learning*, amplifies each standard by defining the core knowledge and skills in practice, supporting teachers and their instruction, and serving to transition classroom instruction from the 2017 Computer Science *Standards of Learning* to the newly adopted standards.

Computer Science Instructional Guide Framework

This instructional guide includes, Understanding the Standard, Concepts and Connections, Opportunities for Integration, and Skills in Practice aligned to each standard. The purpose of each is explained.

Understanding the Standard

This section is designed to unpack the standards, providing both students and teachers with the necessary knowledge to support effective instruction. It includes core concepts that students are expected to learn, as well as background knowledge that teachers can use to deepen their understanding of the standards and plan standards-aligned lessons.

Concepts and Connections

This section outlines concepts that transcend grade levels and thread through the K through 12 computer science as appropriate at each level. Concept connections reflect connections to prior grade-level concepts as content and practices build within the discipline as well as potential connections across disciplines. The connections across disciplines focus on direct standard alignment, where concepts and practices in computer science overlap with similar ideas in other disciplines.

Computer Science connections are aligned to the: 2024 English *Standards of Learning*, 2023 History and Social Science, 2023 Mathematics *Standards of Learning*, 2020 Digital Learning Integration *Standards of Learning*, and 2018 Science *Standards of Learning*.

These cross-disciplinary concepts and practices are foundational for effective interdisciplinary integration.

Opportunities for Integration

This section provides lesson ideas for integrating computer science with English, history and social science, mathematics, and science through multidisciplinary, interdisciplinary, and transdisciplinary approaches. Lesson ideas may involve the integration of standards that may or may not be directly aligned yet are strategically taught together to achieve a purposeful and authentic learning experience that supports meaningful student outcomes such as deeper understanding, skill transfer, and real-world application.

Skills in Practice

This section focuses on instructional strategies that teachers can use to develop students' skills, deepen their conceptual understanding, and encourage critical thinking. These practices are designed to support curriculum writers and educators in weaving pedagogical approaches that deepen student understanding of unit and course objectives, ultimately enhancing learning outcomes. This section provides a framework for planning effective and engaging lessons.

Grade 1: Computer Science Instructional Guide

In Grade 1 students learn how computing components work together to form systems for collecting and sharing data/information. Students continue to develop problem-solving skills that apply computational thinking and can be applied with addressing hardware and software issues. Students refine computational thinking practices through activities such as planning, document creation, and construction of programs that consist of events and sequences. Students will further their knowledge on the importance of simple safeguards to protect private information and apply responsible behaviors and proper use of computing devices.

Algorithms and Programming (AP)

1.AP.1 The student applies computational thinking by sorting items into categories based on multiple attributes and create patterns.

- a. Describe attributes of a set of objects.
- b. List the attributes a set of objects have in common.
- c. Sort and classify concrete objects based on multiple attributes.
- d. Create repeating and increasing patterns.

Understanding the Standard

Computational thinking (CT) is a logical and systematic problem-solving process that uses decomposition, pattern recognition, abstraction, and algorithm thinking to foster creativity and develop solutions. It is universally applicable across various fields and allows individuals to break down complex problems and develop efficient solutions. Its role in computer science is particularly important, as it serves as the foundation for designing algorithms, analyzing data, and solving real-world challenges through the use and development of technology. Computational thinking is an integral part of Virginia's computer science standards.

Computational thinking consists of four key components: decomposition, pattern recognition, abstraction, algorithm thinking.

- Decomposition is the process of breaking apart a problem, process, or task into smaller, more manageable components. This involves identifying and recognizing relationships among the parts.
- Pattern recognition involves identifying commonalities, similarities, or differences in recurring elements.
- Abstraction is a filtering process. It enables one to focus on important and relevant information, while excluding or hiding irrelevant or less important details.
- Algorithmic thinking is the process of developing algorithms in a logical, systematic, and procedural way to solve problems or complete tasks.

Computational Thinking: Increasing by Twos	
Decomposition	Break the task into smaller steps by recognizing that each number follows a pattern of increasing by 2. Identify the rule guiding the sequence, which starts at 2 and progresses by adding 2 each time.
Pattern recognition	When looking at a sequence or arrangement, pattern recognition helps notice repetitions. For example, if the sequence is "red, blue, red, blue," pattern recognition may identify that the colors alternate. A prediction could be made that the next color will be "red."
Abstraction	Consider a pattern of shapes: circle, square, circle, square, circle, square. Use abstraction to ignore the individual shapes or color of the shapes and focus on the core rule: alternating shapes.
Algorithmic thinking	Identify the specific rule or logic behind a pattern. For example, if there is a sequence like "2, 4, 6, 8," use algorithmic thinking to identify that the pattern is increasing by 2 each time. The algorithm here is "start at 2 and add 2 repeatedly."

[1.AP.1a] Objects can be observed and described based on their attributes; these attributes allow people to group items. An attribute refers to a characteristic or quality that describes and differentiates objects. An attribute refers to a characteristic or quality that describes and differentiates objects or data. Attributes of objects may include properties such as color, size, shape, weight, position, number, or texture. At this grade level, students will identify simple and observable attributes within a set of objects, such as color, size, shape, or texture. Students should recognize characteristics among objects and explain how an object is unique or fits into a group.

[1.AP.1b] Through observation and teacher guidance students should list and label attributes within a set of objects. Listing attributes reinforce students' ability to determine commonalities and differences, which is essential in sorting and classification.

[1.AP.1c] Sorting is a process that involves comparing a set of objects to identify similarities and differences. This process strengthens logical thinking and helps students observe recurring structures, which is essential for pattern recognition. Classification is the ability to group objects based on specific categories or attributes, such as size or type. In science and mathematics, the term classify is used when grouping objects or organisms based on one or more attributes. In computer science, students use pattern recognition to classify objects based on given attributes (e.g., color, shape, size, computing system, hardware or software). Classification builds upon students' ability to sort, as it relies on careful observation of similarities and differences. These similarities and differences can be used to categorize and label objects with or without a computing device.

[1.AP.1d] Repeating patterns follow a sequence that repeats itself in a predictable way, this is known as patterning. Patterning is a fundamental cornerstone of mathematics, particularly algebra. The process of generalization leads to the foundation of algebraic reasoning. Opportunities to

identify, describe, extend, create, and transfer patterns are essential to the primary school experience and lay the foundation for algebraic thinking (Mathematics 1.PFA.1).

Patterning should include:

- creating a given pattern using objects, sounds, movements, and pictures.
- describing a pattern, to include identifying the core of the pattern and labeling the pattern.
- recording a pattern with pictures or symbols.
- transferring a pattern into a different form or different representation (e.g., blue–blue–red–green to an AABC repeating pattern).
- analyzing patterns in practical situations (e.g., calendar, seasons, days of the week) (Mathematics 1.PFA.1).

Examples of how students can create repeating and increasing patterns include:

Type of Pattern	Repeating Patterns	Increasing Patterns
Color	Use colored blocks or paper to create a sequence like red, blue, red, blue, repeating.	Shading can gradually darken a color from light to dark across a row or column.
Shape	Arrange shapes like circles, squares, circles, squares in a row.	Draw shapes where the size increases step-by-step (small circle, medium circle, large circle).
Number	Write numbers like 1, 2, 1, 2 in a repeating sequence.	Start with 1 and add 1 each time (1, 2, 3, 4, 5...).

In the science and mathematics standards the term classify is used when grouping objects or organisms based on one or more attributes. In computer science, patterning may include the creation of sequences in algorithms or code, where actions repeat in a predictable manner. Whereas, increasing patterns are the growth or change in a sequence in a predictable and consistent way over time. Consider the following examples for pattern-based algorithms:

- Using pattern blocks to create a repeating pattern as a jumpstart on creating algorithms, regarding sequence.
- Creating simple animations that repeat patterns.
- Create sound patterns.
- Create and extend patterns using blocks and beads, then students can draw using a spreadsheet or computer.

In block-based programming environments, commands are grouped into categories based on function. In higher level programming languages, the kind or type of data determines the classification.

Patterning helps students recognize and create repeated sequences. This computational skill supports reasoning and problem-solving by decomposing tasks into smaller and simple components, which is an important in the development of algorithms.

Concepts and Connections

CONCEPTS

Classification is the foundational concept of sorting and grouping objects based on shared attributes. Students learn to organize objects in multiple ways and identify repeating characteristics, a foundational skill in recognizing and creating patterns.

CONNECTIONS

Within the grade level/course: At this grade level, students use decomposition, pattern analysis, and algorithmic thinking to sort items into categories based on multiple attributes and create patterns (1.AP.1).

Vertical Progression: In Kindergarten, students learn how to sort items into categories based on attributes (K.AP.1). In Grade 2, students engage in the computational thinking practices of decomposition, pattern analysis, and algorithmic thinking as they identify patterns, and design algorithms to compare and contrast objects based on attributes (2.AP.1).

ACROSS CONTENT AREAS

English

- **1.RL.1A** Retell familiar stories sequentially and demonstrate an understanding of the story structure, the central message or lesson, and the details.
- **1.C** The student will develop effective oral communication and collaboration skills to build a community of learners that process, understand, and interpret content together.
- **1.C.2D** Retell, create, and dictate stories, rhymes, poems, and events in sequential order using drama, props, and/or pictures indicating first, next, and last events in a story.

History and Social Science

- **1.5** The student will develop map skills by a) recognizing basic map symbols, including references to land, water, cities, and roads; b) using cardinal directions on maps; and e) constructing simple maps, including a title, map legend, and compass rose.

Mathematics

- **1.PFA.1** The student will identify, describe, extend, create, and transfer repeating patterns and increasing patterns using various representations. This includes a) Identify and describe repeating and increasing patterns; b) Analyze a repeating or increasing pattern and generalize the change to extend the pattern using objects, colors, movements, pictures, or geometric figures; c) Create a repeating or increasing pattern using objects, pictures, movements, colors, or geometric figures; and d) Transfer a repeating or increasing pattern from one form to another.

- **1.MG.2** The student will describe, sort, draw, and name plane figures (circles, triangles, squares, and rectangles), and compose larger plane figures by combining simple plane figures.
b) Sort plane figures based on their characteristics (e.g., number of sides, vertices, angles, curved).

Science

- **1.1** The student will demonstrate an understanding of scientific and engineering practices by planning and carrying out investigations. *1.1 standard is integrated within science content and not taught in isolation. Potential science concepts to apply 1.1 include: 1.4 (plants), 1.5 (animals), 1.6 (relationship between the sun and the Earth), 1.7 (weather and seasonal changes), 1.8 (natural resources)*

DIGITAL LEARNING INTEGRATION

- **K-2.CT** Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods, including those that leverage assistive technologies, to develop and test solutions.
A. Formulate problem definitions suited for technology assisted methods such as data analysis, modeling and algorithmic thinking in exploring and finding solutions.

Opportunities for Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students describe attributes of a set of objects, such as color, size, or shape using adjectives.
- Students sort and classify objects based on multiple attributes, like grouping items by color and size.

History and Social Science

- Students describe attributes of objects or items from the past, such as toys or tools, comparing how they were used by people in the past and now.
- Students sort and classify historical objects based on multiple attributes, such as grouping old and new tools by material or function.
- Students create patterns related to historical events, like repeating holidays or seasons, and increasing patterns in timelines (e.g., “first, second, third” event).

Mathematics

- Students describe and identify different attributes of objects, like size, shape, or color, and use those attributes to classify objects.
- Students list common attributes, such as shape or color, and use them to group objects together.
- Students sort objects based on multiple attributes, such as sorting shapes by both color and size.
- Students create repeating patterns, such as “red, blue, red, blue” with colored blocks, and increasing patterns, such as “1, 2, 3, 4” with objects or numbers.

Science

- Students describe attributes of objects they encounter in nature, such as the color, shape, and size of plants or animals.
- Students list common attributes of plants or animals.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

B. FOSTERING COMPUTATIONAL THINKING PRACTICES:

1. Decompose Real-World Problems
2. Explore Common Features and Identify Patterns
3. Use Abstraction to Simplify, Represent, and Problem Solve
4. Apply Algorithmic Thinking to Problem Solve and Create
5. Apply Computational Thinking Practices to Select, Organize, and Interpret Data

1.AP.2 The student will plan and implement algorithms that include the use of sequence and an event based on a predetermined task.

- a. Plan and create a design document that illustrates thoughts, ideas, and stories in a sequential manner.
- b. Construct step-by-step instructions that include decision-making and repetition.
- c. Identify and explain the role of events that are used in an algorithm.
- d. Test algorithms that are sequential and contain an event.

Understanding the Standard

At school and at home, students regularly engage in step-by-step activities. These routines mirror algorithms which are step-by-step instructions designed to solve problems or perform tasks. Algorithms are finite and sequential, with each step following a specific order. They can be created with or without the use of a computing device. As students learn to construct algorithms, they can reflect on daily routines or experiences that follow a sequence. Understanding algorithms helps students strengthen their logical thinking and problem-solving skills—critical abilities for success in everyday life.

[1.AP.2a] Design documents are used in the planning process to organize thoughts, determine methods of implementation, propose solutions, and track progress.

- Design documents are detailed plans that outline the structure, features, and implementation strategy of a project. It serves as a blueprint, providing clear specifications, goals, and guidelines for developers, designers, and stakeholders. Design documents often include diagrams, technical requirements, workflows, and rationale to ensure a shared understanding of the project's direction and implementation.

Design documents include descriptions of the problem being solved, the input and output requirements, and the step-by-step processes the algorithm will follow. They can include flow charts, diagrams, or story maps that students use to brainstorm the algorithm they want to create.

Creating an algorithm is comparable to planning a narrative. Authors often utilize design tools—such as story maps and storyboards—to organize the structure of a story, including the beginning, middle, and end. Similarly, students can apply these tools to outline the logical flow of an algorithm. Both processes involve the use of clearly defined, sequential steps to systematically achieve the desired outcome.

[1.AP.2b] In kindergarten, students created sequential, task-based algorithms. Learning sequencing supports the development of logical thinking, fosters an understanding of cause and effect, and establishes a foundation for future coding and computational skills. At this grade level, students build upon that foundation by designing decision-making algorithms. These algorithms consist of instructions that determine the

next step or outcome in the algorithm’s execution flow based on a specific condition or set of criteria. They incorporate keywords such as “if” and “if-then,” which are known as conditional statements.

[1.AP.2c] Events in an algorithm refer to specific actions or changes that trigger subsequent actions within a process. These events act as ‘signals’ or ‘triggers’ that initiate, alter, or terminate steps in a sequence. Events can be user actions, such as clicking a mouse button or pressing a key, or system occurrences, such as a timer or low battery.

[1.AP.2d] The practice of reviewing work should be taught early and can be applied across disciplines, including computer science. Students should check that the sequence of steps that compose an algorithm works as intended. Testing an algorithm is the process of verifying that its steps are clear, accurate, and lead to the intended outcomes. To test an algorithm, the teacher should guide students in tracing, the act of following an algorithm step-by-step to ensure each action works as expected. If any step causes an error or doesn’t produce the intended outcome, the algorithm must be modified to resolve the issue. If the algorithm does not work as intended, the students should determine the changes to make in the algorithm in order to complete the task. These changes may include adding, deleting, rearranging, or changing a step in order to obtain the intended outcome.

To test an algorithm that includes events, students need to make sure each event happens the way it's supposed to and that it correctly leads to the next step. If everything happens in the right order and the events cause the right actions, then the algorithm is working properly.

This kind of testing helps determine whether the instructions are functioning by observing how the program responds to actions like clicks, key presses, or timers. It's part of understanding how interactive programs work—programs that respond to users or changes in the system. Through testing and refinement, students strengthen problem-solving skills and improve the reliability of the program.

Concepts and Connections

CONCEPTS

To implement an algorithm, students must create step-by-step instructions that follow a clear sequence to complete a specific task. Events are used to trigger certain actions, instructing the algorithm to respond to user input or changes. Testing the algorithm is important to make sure the algorithm works as intended.

CONNECTIONS

Within the grade level/course: At this grade level, students plan and implement algorithms that included the use of sequence, and an event based on a predetermined task (1.AP.2).

Vertical Progression: In Kindergarten, students followed and created step-by-step instructions to complete tasks, shared ideas in order, and designed and tested simple algorithms (K.AP.2). In Grade 2, students will continue to build upon their algorithm design by planning and implementing algorithms that consist of events and loops using a block-based programming language (2.AP.2).

ACROSS CONTENT ALIGNMENT

English

- **1.RL.1A** Retell familiar stories sequentially and demonstrate an understanding of the story structure, the central message or lesson, and the details.
- **1.C.1** The student will develop effective oral communication and collaboration skills to build a community of learners that process, understand, and interpret content together.
A. Participate in a range of collaborative discussions (one-on-one, in groups, and teacher-led) on grade one topics and texts.
- **1.C.2A** Describe people, places, things, and events with relevant details and using appropriate vocabulary.
- **1.C.2B** Speak audibly with appropriate pacing, prosody, and voice level.
- **1.C.2C** Participate in a variety of oral language activities, including choral speaking and recitation of short poems, rhymes, songs, and stories with repeated patterns and refrains.
- **1.C.2D** Retell, create, and dictate stories, rhymes, poems, and events in sequential order using drama, props, and/or pictures indicating first, next, and last events in a story.

History and Social Science

- **1.5** The student will develop map skills by a) recognizing basic map symbols, including references to land, water, cities, and roads; b) using cardinal directions on maps; and e) constructing simple maps, including a title, map legend, and compass rose.

Mathematics

- **1.CE.1** The student will recall with automaticity addition and subtraction facts within 10 and represent, solve, and justify solutions to single-step problems, including those in context, using addition and subtraction with whole numbers within 20. a) Recognize and describe with fluency part-part-whole relationships for numbers up to 10 in a variety of configurations; b) Demonstrate fluency with addition and subtraction within 10 by applying reasoning strategies (e.g., count on/count back, one more/one less, doubles, make ten); c) Recall with automaticity addition and subtraction facts within 10; e) Solve addition and subtraction problems within 20 using various strategies (e.g., inverse relationships: if $9 + 3 = 12$ then $12 - 3 = 9$; decomposition using known sums/differences: $9 + 7$ can be thought of as 9 decomposed into 2 and 7, then use doubles, $7 + 7 = 14$; $14 + 2 = 16$ or decompose the 7 into 1 and 6; make a ten: $1 + 9 = 10$; $10 + 6 = 16$); f) Represent, solve, and justify solutions to single-step addition and subtraction problems (join, separate, and part-part-whole) within 20, including those in context, using words, objects, drawings, or numbers.

- **1.MG.3** The student will demonstrate an understanding of the concept of passage of time (to the nearest hour and half-hour) and the calendar. g) Identify specific days/dates on a calendar (e.g., What date is Saturday? How many Fridays are in October?); and i) Determine the day/date before and after a given day/date (e.g., Today is the 8th, so yesterday was the ?) and a date that is a specific number of days/weeks in the past or future (e.g., Tim's birthday is in 10 days, what will be the date of his birthday?)

Science

- **1.1** The student will demonstrate an understanding of scientific and engineering practices by b) planning and carrying out investigations with guidance, conduct investigations to produce data identify characteristics and properties of objects by observations use tools to measure relative length, weight, volume, and temperature of common objects. e) developing and using models use physical models to demonstrate simple phenomena and natural processes *1.1 standard is integrated within science content and not taught in isolation. Potential science concepts to apply 1.1 include: 1.2 (object movement), 1.3 (physical properties), 1.4 (plants), 1.6 (relationship between the sun and the Earth), 1.7 (weather and seasonal changes), 1.8 (natural resources).*

DIGITAL LEARNING INTEGRATION

- **K-2.CT** Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods, including those that leverage assistive technologies, to develop and test solutions.
A. Formulate problem definitions suited for technology assisted methods such as data analysis, modeling and algorithmic thinking in exploring and finding solutions.
C. Break problems into component parts, extract key information, and develop descriptive models, using technologies when appropriate, to understand complex systems or facilitate problem-solving.
- **K-2.ID** Students use a variety of technologies, including assistive technologies, within a design process to identify and solve problems by creating new, useful or imaginative solutions or iterations.
B. Select and use appropriate technologies to plan and manage a design process that considers design constraints and calculated risks.
C. Use appropriate technologies to develop, test, and refine prototypes as part of a cyclical design process.

Opportunities for Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students plan and create a design document that illustrates their thoughts, ideas, or stories in a sequential order.
- Students construct step-by-step instructions for tasks that include decision-making and repetition.

- Students identify and explain the role of events in a simple algorithm.
- Students test sequential algorithms that include a specific event to see if the algorithm works as expected.

History and Social Science

- Students plan and create a design document that outlines a historical event or story in sequential order, such as the steps leading to the signing of a historical document.
- Students identify and explain events in history that fit into a sequence, such as the events leading up to a major discovery or turning point in history.

Mathematics

- Students plan the sequence of steps needed to solve math problems, like counting objects in a particular order or arranging numbers in a sequence.
- Students create step-by-step instructions for a repetitive process, such as counting by twos or fives, and decision-making, such as choosing whether to add or subtract based on a given problem.
- Students test algorithms with simple mathematical tasks that follow a sequence, like organizing numbers or adding objects in a specified order.

Science

- Students plan and create a design document that outlines how to conduct a simple science experiment in a step-by-step sequence, such as how to grow a plant from a seed.
- Students construct instructions for science experiments that involve decision-making and repetition.
- Students identify events in a scientific process and explain how they are part of a sequence.
- Students test algorithms in science, such as conducting a sequence of steps to observe a natural phenomenon and checking if the outcome matches expectations.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

B. FOSTERING COMPUTATIONAL THINKING PRACTICES:

1. Decompose Real-World Problems
2. Explore Common Features and Identify Patterns

3. Use Abstraction to Simplify, Represent, and Problem Solve
4. Apply Algorithmic Thinking to Problem Solve and Create
5. Apply Computational Thinking Practices to Select, Organize, and Interpret Data

C. FOSTERING ITERATIVE DESIGN PRACTICES:

1. Identify, Define, and Evaluate Real-world Problems
2. Plan and Design Artifacts
3. Create, Communicate and Document Solutions
4. Test and Optimize Artifacts

1.AP.3 The student will use the iterative design process to construct, test, and debug algorithms that include sequencing and an event.

- a. Identify and describe the concept of debugging.
- b. Analyze and explain the results of an algorithm.
- c. Revise and improve an algorithm to produce desired outcomes.

Understanding the Standard

The iterative design process is a systematic approach to creating and refining products, systems, or solutions through repeated cycles of design, evaluation, and improvement. It emphasizes flexibility and responsiveness to feedback, allowing students to make incremental enhancements based on real-world testing and user input.

[1.AP.3a] If the algorithm does not work as intended, the students should determine the changes to make in the algorithm. These changes may include adding, deleting, rearranging, or changing a step in order to obtain the intended outcome. Debugging is a technique used to find errors or “bugs” within a program and fixing these errors. Revisions may be made regardless of the presence of errors, as revisions may contribute to improving the overall effectiveness and efficiency. In this grade, debugging might involve reviewing steps in a sequence to ensure each part is correct. If incorrect, students can rearrange steps in a task or adjusting actions to achieve the intended outcome. It is a foundational problem-solving skill that helps students learn resilience and critical thinking.

[1.AP.3b] Analyzing an explaining the results of an algorithm is the observation of each step, confirming the logical flow, and verification of intended outcome. This can be done with or without a computing device.

- For example: The goal of an algorithm is to sort blocks by size; therefore, students would observe each step in the algorithm and its logical flow. The student would then explain whether the intended outcome was achieved by verifying if the blocks were sorted correctly.

[1.AP.3c] Students should test algorithms to check the sequence of steps work as intended. Testing is a critical step in the iterative design process. If an algorithm does not produce the intended outcome during testing, students should revise and improve the algorithm. Revisions to the algorithm will occur until the outcome meets the intended results. This cycle of testing and refining helps build stronger algorithms and deeper understanding of how each step contributes to the overall outcome. Through testing and refinement, students strengthen problem-solving skills, by showing that mistakes and adjustments are part of the problem-solving process and improve the reliability of the program.

Concepts and Connections

CONCEPTS

Iterative design process helps students understand that solving problems often takes multiple tries and that improvement comes through testing and making changes. Debugging is a technique that is used to identify mistakes, try new solutions, and resolve errors.

CONNECTIONS

Within the grade level/course: At this grade level, students will use the iterative design process to construct, test, and debug algorithms that include sequencing and an event (1.AP.3).

Vertical Progression: In Grade 2, students build their use of the iterative design process to create, test, and debug a program containing events and loops in a block-based programming tool (2.AP.3).

ACROSS CONTENT AREAS

English

- **1.C.2D** Retell, create, and dictate stories, rhymes, poems, and events in sequential order using drama, props, and/or pictures indicating first, next, and last events in a story.

History and Social Science

- **1.5** The student will develop map skills by a) recognizing basic map symbols, including references to land, water, cities, and roads; b) using cardinal directions on maps; and e) constructing simple maps, including a title, map legend, and compass rose.

Mathematics

- **1.CE.1** The student will recall with automaticity addition and subtraction facts within 10 and represent, solve, and justify solutions to single-step problems, including those in context, using addition and subtraction with whole numbers within 20. a) Recognize and describe with fluency part-part-whole relationships for numbers up to 10 in a variety of configurations; b) Demonstrate fluency with addition and subtraction within 10 by applying reasoning strategies (e.g., count on/count back, one more/one less, doubles, make ten); c) Recall with automaticity addition and subtraction facts within 10; e) Solve addition and subtraction problems within 20 using various strategies (e.g., inverse relationships: if $9 + 3 = 12$ then $12 - 3 = 9$; decomposition using known sums/differences: $9 + 7$ can be thought of as 9 decomposed into 2 and 7, then use doubles, $7 + 7 = 14$; $14 + 2 = 16$ or decompose the 7 into 1 and 6; make a ten: $1 + 9 = 10$; $10 + 6 = 16$); and f) Represent, solve, and justify solutions to single-step addition and subtraction problems (join, separate, and part-part-whole) within 20, including those in context, using words, objects, drawings, or numbers.
- **1.MG.3** The student will demonstrate an understanding of the concept of passage of time (to the nearest hour and half-hour) and the calendar. g) Identify specific days/dates on a calendar (e.g., What date is Saturday? How many Fridays are in October?); and

i) Determine the day/date before and after a given day/date (e.g., Today is the 8th, so yesterday was the __.), and a date that is a specific number of days/weeks in the past or future (e.g., Tim’s birthday is in 10 days, what will be the date of his birthday?) .

Science

- **1.2** The student will investigate and understand that objects can move in different ways. Key ideas include a) objects may have straight, circular, spinning, and back-and-forth motions; and b) objects may vibrate and produce sound.

DIGITAL LEARNING INTEGRATION

- **K-2.CC** Students communicate clearly and express themselves creatively for a variety of purposes using appropriate technologies (including assistive technologies), styles, formats, and digital media appropriate to their goals.
B. Create original works or responsibly repurpose or remix digital resources into new creations.
- **K-2.EL** Students leverage technologies, including assistive technologies, to take an active role in choosing, achieving, and demonstrating competency in their learning goals, informed by the learning sciences.
C. Use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.

Opportunities for Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students discuss and describe the concept of debugging, such as how they fix mistakes in their writing or steps in a sequence of events.
- Students analyze and explain the results of an algorithm by talking about what happened after following the steps, like when a simple event (like pressing a button) caused something to happen.
- Students revise and improve their algorithms to get the desired outcomes, such as changing the sequence of steps in a story to make more sense or adjusting how they follow a pattern to match the correct order.

Mathematics

- Students discuss debugging in math, like checking their work to see if the answer to an addition or subtraction problem makes sense.
- Students analyze the results of an algorithm in math, like seeing if following a sequence of numbers (e.g., counting by twos, fives, or tens) leads to the correct solution.
- Students revise and improve their math algorithms, such as adjusting the steps in a problem-solving process to ensure the right outcome.

Science

- Students discuss debugging in science, such as identifying mistakes in an experiment.
- Students analyze and explain the results of a scientific experiment by looking at the data collected and figuring out if the outcome matches their hypothesis.
- Students revise and improve an experiment or algorithm based on their analysis to produce the desired outcomes, such as adjusting the amount of sunlight for plants to see if it helps them grow better.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

C. FOSTERING ITERATIVE DESIGN PRACTICES:

1. Identify, Define, and Evaluate Real-world Problems
2. Plan and Design Artifacts
3. Create, Communicate and Document Solutions
4. Test and Optimize Artifacts

[Back to Algorithms and Programming \(AP\)](#)

Computing Systems (CSY)

1.CSY.1 The student will describe how computing components work together to create a computing system.

- a. Identify and define hardware, software, and computing systems.
- b. Identify common components of computing systems in different types of computing devices.
- c. Describe how hardware and software work together to form a computing system.

Understanding the Standard

Computing systems are composed of different components. These components enable the user to complete different tasks using a computing system. A computing system is a group of hardware and software that work together to complete various tasks. Understanding how these parts interact helps users make better use of technology and troubleshoot problems when they arise.

[1.CSY.1a]

Hardware	Software
Hardware refers to all the physical parts of a device.	Software refers to the programs that tell the hardware what to do.
• Computer • Headphones • Keyboard • Mouse • Screen • Power Button	• Applications (Apps) • Communication Tools • Games • Media Players • Operating Systems • Web Browsers

Hardware and software need each other to make the computer function properly. The hardware does exactly what the software tells it to do. The hardware provides the structure and the tools, while the software provides the instructions and makes those tools work. Hardware is like the body of a computing device; it includes all the physical parts that can be seen and touched. But just like our bodies have internal organs working behind the scenes, a computing device also has internal hardware that keeps it running. Software, on the other hand, is like the brain, it controls how the computing device thinks and processes information, telling the hardware what to do.

[1.CSY.1b] A computing device is an electronic tool or machine designed to process data and perform tasks using instructions, often referred to as software. These devices can vary widely in size and function, from simple calculators to advanced smartphones. Computing devices may also share common components that perform similar functions but may look or work differently depending on the device.

Some common components of computing systems include:

Components	Function
Processor (CPU)	The central processing unit (CPU) is the brain of the computer. It carries out instructions from programs, performs calculations, and manages tasks, making the device function.
Memory (RAM)	Random Access Memory (RAM) is temporary storage that the CPU uses to hold data and instructions for active tasks. It allows the device to run smoothly and quickly by providing quick access to frequently used data.
Storage (Hard Drive or SSD)	Storage is where all of the data, programs, and files are kept permanently (until deleted). Hard drives (HDD) or solid-state drives (SSD) store this information, with SSDs offering faster access speeds.
Input Devices (Keyboard, Mouse, Touchscreen)	These devices allow users to interact with the computer. A keyboard and mouse are used for typing and navigation, while touchscreens enable direct interaction with the display.
Output Devices (Monitor, Speaker)	Output devices display or produce information from the computer. A monitor shows visuals, and speakers produce sound, allowing users to see and hear results from the computer's processes.
Battery	The battery provides power to the device, allowing it to function without being plugged into a power source. It stores energy and delivers it to components when needed.
Networking Components (Wi-Fi, Bluetooth)	These components enable the device to connect to networks (Wi-Fi) and communicate with other devices (Bluetooth), allowing data transfer, internet access, and wireless communication.
Graphics Processing Unit (GPU)	The GPU handles graphics-related tasks, such as rendering images, video, and animations. It enhances visual performance, especially for gaming, video editing, and graphic design.
Operating System	The operating system (OS) is the software that manages hardware resources and provides a user interface. It enables the computer to run programs and ensures everything works together smoothly.
Software Applications	Software applications are programs designed to perform specific tasks for the user, such as word processing, web browsing, gaming, or photo editing. These applications run on the operating system and make the device useful for various functions.

A computing device is any electronic tool or machine designed to process data and perform tasks using instructions, often referred to as software. These devices can vary widely in size and function, from simple calculators to advanced smartphones. Computing devices may include but are not limited to:

- Desktop Computers.
- Laptops.
- Tablets.
- Smartphones.
- Smartwatches.
- Game Consoles.
- Servers.
- Other Embedded Devices.

Embedded devices are specialized computers that are built into other machinery or objects such as in cars, appliances, or thermostats. They are designed to perform specific functions within a larger system and often operate with limited user interaction. Embedded systems typically include a processor, memory, and software optimized for real-time control and efficiency.

[1.CSY.1c] Hardware and software work together seamlessly to create a functioning computing system. Hardware refers to the physical components of a computer, such as the processor, memory, keyboard, and display. These components provide the foundation for the computer's operation. Software, on the other hand, consists of the programs and instructions that tell the hardware how to perform specific tasks. The software communicates with the hardware to process inputs, execute commands, and deliver outputs. For example, when a user types on a keyboard (hardware), the software interprets the input and displays the corresponding characters on the screen. This collaboration between hardware and software is essential for a computer to operate as a cohesive system.

Concepts and Connections

CONCEPTS

Computing systems are made up of both hardware and software components, each with a specific purpose and function. Computing devices also share common components that work together to perform similar functions. These components are designed to work together as an integrated system to perform tasks and solve problems.

CONNECTIONS

Within the grade level/course: At this grade level, students describe how computing components work together to create a computing system (1.CSY.1).

Vertical Progression: In Kindergarten, students explored the purpose, components, and functions of computing devices, identified different types of screen-based devices, recognize screen-related activities at home and school, and demonstrated proper device usage (K.CSY.1). In Grade 2, students will describe the characteristics of computing systems including hardware, software, input, and output (2.CSY.1).

ACROSS CONTENT AREAS

English

- **1.RL** The student will use textual evidence to demonstrate comprehension and build knowledge from a variety of literary texts heard or read including fantasy, humor, fable/fairy tale, realistic fiction, historical fiction, and folklore/tall tale. *(Note: only aligned if students are reading about how computing components work together to create a computing system.)*
- **1.C** The student will develop effective oral communication and collaboration skills to build a community of learners that process, understand, and interpret content together.

Opportunities for Computer Science Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students use storytelling to describe a computing system, comparing hardware and software to characters working together in a story (e.g., the keyboard as the writer, the screen as the storyteller).
- Students can create a class book defining hardware, software, and computing systems using illustrations and simple descriptions.

History and Social Science

- Students identify common components of devices used in social studies projects, such as tablets used for exploring interactive maps or computers for virtual museum tours.

Mathematics

- Students identify and define components in a math-based computing system, such as a calculator app (software) and the number pad (hardware).
- Students identify common components of devices used for math learning, such as a tablet running a counting game or a computer displaying a digital graph.

Science

- Students identify and define components of computing systems used in science, such as digital thermometers (hardware) and temperature recording programs (software).
- Students identify common components of devices used for science experiments, such as microscopes with digital screens or sensors that measure movement.
- Students explore systems by comparing how the human body and computing devices work together to process information. In science, they learn that body parts like eyes and ears send signals to the brain, which responds with actions. In computer science, they identify that hardware collects input and software gives instructions for output

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

B. FOSTERING COMPUTATIONAL THINKING PRACTICES:

2. Explore Common Features and Identify Patterns
5. Apply Computational Thinking Practices to Select, Organize, and Interpret Data

1.CSY.2 The student will use accurate terminology to describe when a computing system might not work as expected.

- a. Identify and describe a problem with a device or computing system when it does not work as expected.
- b. Propose a solution to simple hardware or software issues.

Understanding the Standard

Computing systems might not work as expected because of hardware or software problems. Accurately describing a problem is the first step toward finding a solution. During instruction, learners engage hands-on with physical components and relevant technologies to apply systematic problem-solving strategies. This experiential approach helps build foundational skills in debugging, error detection, and iterative refinement of solutions. Developing proficiency in these areas is essential for effective interaction with complex computing environments and real-world applications. (Mathematics SOL Grade 1)

[1.CSY.2a] As students identify a problem and begin finding a solution, they can use computational thinking.

- Decomposition can be used by breaking the problem into smaller, manageable tasks.
- Pattern recognition can be used to understand how the computing device typically works.
- Abstraction can be used to focus on the main issue while ignoring extraneous information.
- Algorithmic thinking can be used to create step-by-step instructions that others can follow.

[1.CSY.2b] Problems with computing systems can arise from various causes, including hardware settings, programming errors, faulty connections to other devices, or human error. To address these issues, developmentally appropriate problem-solving strategies include debugging simple programs, troubleshooting based on prior experience, and seeking assistance by clearly describing the problem for example, “The computer will not turn on,” “The pointer on the screen will not move,” or “I lost the web page.” Being aware that human factors can also affect system performance helps foster a more comprehensive approach to diagnosing and resolving technical issues.

Consider the following examples of common hardware or software issues:

Issue Type	Problem	Explanation	Possible Solution
Hardware	Computer will not turn on	The computer device is not starting up.	Check if the power button is pressed correctly. Ensure the device is plugged in, or if it is a laptop, make sure the battery is charged.
	Screen is blank	The display does not show anything, even though the computer is on.	Try adjusting the brightness, check the cables (for desktop monitors), or restart the device.
	Keyboard not working	Keys do not register when pressed.	Check if the keyboard is connected properly. If it is wireless, check the batteries or reconnect it. It can also restart the computer.
	Mouse not responding	The mouse cursor will not move or click.	Ensure the mouse is connected or the batteries are charged (if wireless). Try restarting the computer or using a different USB port.
Software	Program not opening	An application or game will not start.	Restart the computer or close other programs to free up memory. If that does not help, try reinstalling the app or checking for updates.
	Slow performance	The computer or app is running very slowly.	Close unused programs or tabs. Try restarting the device, or clear cache/history in apps or browsers.
	Frozen screen (or app)	The screen or an application becomes unresponsive, and the user can not click anything.	Try pressing Ctrl + Alt + Delete (on Windows) or Command + Option + Esc (on

			Mac) to force quit the app. Restart the device if needed.
	Cannot connect to Wi-Fi	The device is not connecting to the internet.	Check if the Wi-Fi is on and connected to the correct network. Restart the router or the device.
	Error messages	The device or program shows an error or warning message, like "File not found" or "App crashed."	Read the message carefully to understand what is wrong. The program may need to be restarted, the software updated, or an app reinstalled.

Concepts and Connections

CONCEPTS

Computing devices may not always work as intended and errors can occur during use. It is important for students to understand that these issues are common and can often be resolved through troubleshooting. By using basic troubleshooting strategies, students can identify the cause of the problem and take appropriate steps to resolve it.

CONNECTIONS

Within the grade level/course: At this grade level, students use accurate terminology to describe when a computing system might not work as expected (1.CSY.2).

Vertical Progression: In Kindergarten, students recognized when a computing device isn't working, identified the problem, and explained how to fix it (K.CSY.2). In Grade 2, students will demonstrate an understanding of how to troubleshoot simple hardware and software problems that may occur during use (2.CSY.2).

ACROSS CONTENT AREAS

English

- **1.RL** The student will use textual evidence to demonstrate comprehension and build knowledge from a variety of literary texts heard or read including fantasy, humor, fable/fairy tale, realistic fiction, historical fiction, and folklore/tall tale. *(Note: only aligned if students are using accurate terminology from the text(s) that they read or listen to. These standards must be connected to text)*
- **1.C** The student will develop effective oral communication and collaboration skills to build a community of learners that process, understand, and interpret content together.

Mathematics

- **1.CE.1** The student will recall with automaticity addition and subtraction facts within 10 and represent, solve, and justify solutions to single-step problems, including those in context, using addition and subtraction with whole numbers within 20.
g) Determine the unknown whole number that will result in a sum or difference of 10 or 20 (e.g., $14 - \underline{\quad} = 10$ or $15 + \underline{\quad} = 20$).

DIGITAL LEARNING INTEGRATION

- **K-2. CT** Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods, including those that leverage assistive technologies, to develop and test solutions.
A. Formulate problem definitions suited for technology assisted methods such as data analysis, modeling and algorithmic thinking in exploring and finding solutions.
- **K-2.ID** Students use a variety of technologies, including assistive technologies, within a design process to identify and solve problems by creating new, useful or imaginative solutions or iterations.
A. Know and use appropriate technologies in a purposeful design process for generating ideas, testing theories, creating innovative digital works or solving authentic problems.

Opportunities for Computer Science Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students create a class chart of common technology problems (e.g., “The screen is frozen.”) and brainstorm possible solutions.
- Students will engage in an role-playing activity where they act out different computing system issues and work together to troubleshoot solutions.
- Students will write or dictate a “tech help” guide using clear vocabulary to describe problems and solutions (e.g., “If the sound isn’t working, check the volume.”).

Mathematics

- Students identify and describe a problem with a device used for math activities, such as a calculator not displaying numbers correctly.
- Students propose a solution to simple issues, such as clearing the screen or checking if the device is turned on.

Science

- Students identify and describe a problem with a science-related computing system, such as a temperature sensor not displaying readings.
- Students propose a solution to simple issues, such as reconnecting the sensor or checking the battery.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

B. FOSTERING COMPUTATIONAL THINKING PRACTICES:

1. Decompose Real-World Problems
2. Explore Common Features and Identify Patterns
3. Use Abstraction to Simplify, Represent, and Problem Solve
4. Apply Algorithmic Thinking to Problem Solve and Create
5. Apply Computational Thinking Practices to Select, Organize, and Interpret Data

C. FOSTERING ITERATIVE DESIGN PRACTICES:

1. Identify, Define, and Evaluate Real-world Problems
2. Plan and Design Artifacts
3. Create, Communicate and Document Solutions
4. Test and Optimize Artifacts

[Back to Computing Systems \(CSY\)](#)

Cybersecurity (CYB)

1.CYB.1 The student will demonstrate safe and responsible use of computing technologies.

- a. Describe safe and responsible uses of computing technologies based on the school rules and acceptable use policy (AUP).
- b. Demonstrate safe and responsible behaviors when using computing technologies and online communication.
- c. Discuss the process for reporting inappropriate technology use at school or home.
- d. Classify appropriate and inappropriate uses of technology at school or at home.
- e. Explain the consequences of inappropriate uses of computing technologies.

Understanding the Standard

With teacher guidance, students will begin to understand appropriate and inappropriate uses of computing technologies within specific settings and contexts. They will identify safe practices and responsible behaviors based on their school's rules and the acceptable use policy (AUP), including guidelines on online conduct, protecting personal information, and proper device usage. Through hands-on experiences, students will have opportunities to demonstrate these practices in real-world situations, reinforcing their understanding. Additionally, they will learn how to communicate the correct procedures for reporting behaviors that violate school rules or the AUP, ensuring they know how to seek help when encountering inappropriate use of technology.

[1.CYB.1a] Computing technologies refer to a broad range of devices and tools that help us process information and perform tasks using computers and software. These technologies enable students to access information, communicate, create content, and solve problems.

- For students, this can include everything from desktop computers and laptops to tablets, smartphones, and even smartboards used in the classroom.

Computing technologies enhance learning in several keyways, making education more engaging, interactive, and personalized for young students. They can enhance learning by providing interactive experiences through educational apps, online resources, and virtual learning environments.

[1.CYB.1b] Teaching students about the role of software is essential for helping them express their ideas and complete assignments effectively. By connecting computing technologies to students' everyday experiences, educators can demonstrate their relevance and inspire responsible, purposeful use. This approach lays a foundation for developing digital literacy—an essential skill that enables students to navigate, evaluate, and create in today's technology-driven world, preparing them for future learning and responsible digital citizenship.

Digital literacy refers to the ability to use technology effectively and responsibly to access, evaluate, create, and communicate information. It goes beyond simply knowing how to use devices like computers, tablets, or smartphones. Digital literacy also includes understanding how to use these tools in appropriate ways that are safe, ethical, and meaningful in the context of learning.

[1.CYB.1c] Students will need the opportunity to demonstrate these practices in real situations, reinforcing their understanding through hands-on experience. An example could be students ensuring their personal information stays private while chatting with others in an online game.

Computer networks, including the Internet, can be used to connect people to other people, places, information, and ideas. Understanding how networks operate is fundamental to leveraging digital communication and accessing resources in today's connected world. In order to keep students safe, schools and divisions have rules on the appropriate use of technology. Students should be aware of what is allowed and not allowed when using division/school technology.

[1.CYB.1d] Educating students about responsible use is essential for creating a safe and positive environment both online and offline. Improper use of computing technologies can lead to serious consequences, such as privacy violations and exposure to harmful content. It may also pose security risks through malware infections or encourage academic dishonesty, like plagiarism. The misuse of technology can also have legal repercussions, including fines or lawsuits. Teaching students the proper use of computing technologies helps them develop respectful and responsible online behavior, reducing the likelihood of issues like cyberbullying. Cyberbullying is a form of bullying that occurs when online communications are sent that are intimidating or threatening in nature.

Concepts and Connections

CONCEPTS

Safe and responsible use of computing technologies is a key aspect of being a responsible digital citizen. Digital citizenship involves making good choices online, showing respect to others, and utilizing technology in ways that are mutually beneficial.

CONNECTIONS

Within the grade level/course: At this grade level, students describe safe and responsible uses of computing technologies based on the school rules and acceptable use policy (AUP), demonstrate safe and responsible behaviors when using computing technologies and online communication, discuss the process for reporting inappropriate technology use at school or home, classify appropriate and inappropriate uses of technology at school or at home and explain the consequences of inappropriate uses of computing technologies (1.CYB.1).

Vertical Progression: In Kindergarten, students learned and followed safe and responsible technology using an acceptable use policy (AUP), followed school rules, and discussed how to report inappropriate behavior (K.CYB.1). In Grade 2, students will explain the need for safe and responsible uses of computing technologies, create a flowchart to illustrate the process for reporting inappropriate use of technology at school or at home, and demonstrate and model safe and responsible behaviors when using computing technologies and online communication (2.CYB.1).

ACROSS CONTENT AREAS

History and Social Science

- **1.10** The student will apply the traits of a good citizen by a) focusing on fair play, exhibiting good sportsmanship, helping others, and treating others with respect; b) recognizing the purpose of rules and practicing self-control; c) working hard in school; d) taking responsibility for one's own actions; and e) valuing honesty and truthfulness in oneself and others.

DIGITAL LEARNING INTEGRATION

- **K-2.DC** Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act in ways that are safe, legal, and ethical.
B. Engage in positive, safe, legal, and ethical behavior when using technology, including social interactions online or when using networked devices.
- **K-2.CC** Students communicate clearly and express themselves creatively for a variety of purposes using appropriate technologies (including assistive technologies), styles, formats, and digital media appropriate to their goals.
D. Publish or present content that customizes the message and medium for the intended audiences.
- **K-2.GC** Students use appropriate technologies, including assistive technologies, to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.
A. Use appropriate technologies to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning.

Opportunities for Computer Science Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students describe safe and responsible uses of computing technologies, such as keeping passwords private and using kind language online. They can write, draw, or create an artifact to explain the importance of being safe online.
- Students discuss the process for reporting inappropriate technology use, such as telling a teacher if they see something unsafe online.
- Students explain the consequences of inappropriate technology use, such as losing device privileges or having limited internet access.

History and Social Science

- Students will compare digital safety to real-world safety (e.g., wearing goggles when doing an experiment vs. using strong passwords online) to understand the importance of responsible technology use.
- Students will observe and classify different materials that protect objects (e.g., cases for tablets, covers for books) and relate this to protecting personal information online.
- Students will explore cause and effect by discussing what happens when technology is used appropriately versus inappropriately (e.g., staying on approved websites vs. clicking on unknown links).

Science

- Students will conduct an experiment with clean vs. dirty hands on a touchscreen to explore how physical care of devices is part of responsible use.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

D. FOSTERING DIGITAL LITERACY PRACTICES:

1. Responsible Use Practices
2. Safeguard Well-Being of Self and Others
3. Evaluate Resources and Recognize Contributions

1.CYB.2 The student will discuss the importance of using a password to protect private information.

- a. Describe the purpose of usernames and passwords.
- b. Discuss how passwords are private information and are used to protect the privacy of information.

Understanding the Standard

Passwords are essential for protecting personal information and securing accounts from unauthorized access. Keeping passwords confidential and selecting strong, hard-to-guess passwords are critical practices for maintaining online security. Strong passwords typically include a combination of letters, numbers, and special characters, and meet length requirements that increase their complexity. Many websites enforce specific rules regarding password length and composition to enhance security. Emphasizing the importance of never sharing passwords, both in the classroom and at home, helps reinforce responsible digital habits and reduces the risk of security breaches.

[1.CYB.2a] A username is a unique name that people use to log into a device or online account. It is like a nickname that helps the computer recognize who is logging in. Usernames serve as an identifier that distinguishes one user from another within a system. For example, just like every student has their own cubby or spot in the classroom, a username is a special space just for them online. Together with a password, the username forms the basis of authentication, verifying the user's identity and granting access to personalized settings and data. Choosing a memorable but secure username is important for maintaining account security.

Passwords are used to protect devices and information from unauthorized access. A password is a secret code with a set of letters and numbers that helps keep a student's account safe. It allows people the ability to access their online space. It is something only the student should know and use to log in with their username. Strong passwords have characteristics that make them more difficult to guess. Many sites have rules as to the length and composition of passwords; these rules help create stronger passwords.

It is important to encourage the practice of keeping passwords private, both in the classroom and at home. Students are encouraged to only share passwords with trusted adults such as their teacher and guardians.

[1.CYB.2b] Passwords are used to protect personal and private information, by keeping it safe from others who might want to misuse it. When someone creates a password, it acts like a secret key that only they know. These secret keys lock up important details like names, addresses, and favorite games, making sure that only the person with the right password can access them. By using strong passwords, people can control who sees or changes their information, helping to keep their accounts and personal data safe from strangers.

Types of Information		
Personal	Private	Public
Personal information is any detail about someone that makes them unique but does not directly identify them or put their safety at risk.	Private information includes details that can identify someone or give others access to their personal life.	Public information is information that is okay to share with anyone and is typically available for everyone to see.
• First name only • Favorite color • Favorite sport • Hobby • Pet name	• Full name • Home address • Birthday • School name • passwords	• School assignments • General facts (the Earth orbits the sun) • General school rules • School spirit days

Students should explore safe practices to keep their personally identifying information private, such as using strong passwords, being cautious about what they post online, and understanding privacy settings on devices and applications.

Concepts and Connections

CONCEPTS

Using computers comes with responsibility, such as not sharing login information, keeping passwords private, and logging off devices when finished. Understanding the difference between personal, public, and private information is essential in identifying appropriate cybersecurity safeguards.

CONNECTIONS

Within the grade level/course: At this grade level, students describe the purpose of usernames and passwords, discuss how passwords are private information and are used to protect the privacy of information (1.CYB.2).

Vertical Progression: In Kindergarten, students learned the importance of protecting personal information online, what to share and not share, and how to keep it private (K.CYB.2). In Grade 2, students will identify and classify passwords as strong or weak, explain how a strong password helps protect the privacy of information, and explain the risk of sharing passwords (2.CYB.2).

ACROSS CONTENT AREAS

English

- **1.C** The student will develop effective oral communication and collaboration skills to build a community of learners that process, understand, and interpret content together.
- **1.C.1A** Participate in a range of collaborative discussions (one-on-one, in groups, and teacher-led) on grade one topics and texts. This includes: i) Listening actively and following agreed-upon rules for participating in discussions (e.g., waiting for a turn to speak without unnecessary interruptions and staying on topic); ii) Respectfully building on others' ideas and expressing their own clearly; iii) Asking questions to seek help, get information, or clarify information for further understanding; and iv) Expressing ideas and needs in complete sentences.

History and Social Science

- **1.10** The student will apply the traits of a good citizen by a) focusing on fair play, exhibiting good sportsmanship, helping others, and treating others with respect; b) recognizing the purpose of rules and practicing self-control; c) working hard in school; d) taking responsibility for one's own actions; and e) valuing honesty and truthfulness in oneself and others.

DIGITAL LEARNING INTEGRATION:

- **K-2.DC** Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act in ways that are safe, legal, and ethical.
B. Engage in positive, safe, legal, and ethical behavior when using technology, including social interactions online or when using networked devices.
D. Manage their personal data to maintain digital privacy and security and are aware of data- collection technology used to track their activity online.

Opportunities for Computer Science Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students describe the purpose of usernames and passwords, such as logging in with a unique ID.
- Students discuss how passwords are private information and should not be shared, such as keeping a login safe to protect personal work.

Mathematics

- Students describe the purpose of usernames and passwords when accessing online math tools.
- Students discuss how passwords help protect personal math progress and scores from being changed by others.

Science

- Students explore how animals and plants have natural defenses (e.g., shells, thorns) to protect themselves, comparing this to how passwords protect private information.
- Students can conduct a hands-on activity using locks and keys to illustrate how passwords work as digital “keys” to keep information safe.
- Students experiment with patterns by creating secret codes or simple ciphers, discussing how strong passwords use patterns to increase security.
- Students observe how a closed vs. open container protects objects inside, relating this to how passwords keep information secure.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

D. FOSTERING DIGITAL LITERACY PRACTICES:

1. Responsible Use Practices
2. Safeguard Well-Being of Self and Others
3. Evaluate Resources and Recognize Contributions

[Back to Cybersecurity \(CYB\)](#)

Data and Analysis (DA)

1.DA.1 The student will explore how data can be stored and retrieved from various computing devices.

- a. Identify data formats used for various purposes, including audio, images, text, and video.
- b. Explore and identify computing devices that collect, store, and/or display data.

Understanding the Standard

Data within a computing device can be stored in various formats. The format in which data is stored significantly impacts storage needs, as different formats require varying amounts of space. The choice of data format affects storage efficiency while preserving data quality and usability. Some formats are better suited for specific types of data, such as text, images, or audio, influencing how quickly and accurately the data can be accessed or processed. Understanding these trade-offs is important when managing digital storage and optimizing system performance.

[1.DA.1a] Consider the various types of data, such as audio, images, text, or video.

Data Formats	
Audio	An audio format in computer science refers to the specific way sound data is encoded and stored in a digital file. Audio formats define how sound waves, such as music or voice, are converted into digital data that computers can process, store, and play back. Each audio format balances file size, sound quality, and compatibility with different devices and software, allowing users to choose the best option for their needs.
Images	An image format in computer science defines how picture data is digitally stored, including details like colors and resolution. Different formats balance quality, file size, and compatibility. Each image format is suited to specific tasks, depending on the need for quality, transparency, or animation.
Text	A text format specifies how characters and symbols are stored as digital data, enabling computers to display, process, or edit written information. Text formats vary in complexity, from simple notes to rich, formatted documents with multimedia.
Video	A video format defines how moving images and sound are stored and compressed, determining quality, file size, and compatibility with devices. Each video format is chosen based on the need for quality, storage efficiency, or device compatibility.

[1.DA.1b] Data on a computing device, like a computer, tablet, or phone, is kept in a specific place called storage. Storage on a device is like a special place where all the important information is kept. Retrieving data from a computing device can include downloading a picture, video game, or file. The device has to go through all the pieces of information stored to find the correct data. Once it finds it, the device displays data on the screen. Retrieving data is the way a device finds something that was saved previously and brings it back to being seen or used again.

Consider these examples of computing devices that store and/or display data:

Computing Devices		
Store Data	Display Data	Both Store and Display Data
• External hard drives • USB flash drives • Cloud storage servers • Network attached storage (NAS) devices	• Monitors • Projectors • E-Readers (for reading digital books) • Digital picture frames	• Computer / laptop • Tablet • Smartphone • Game console

Devices have storage to keep information safe and ready for when it needs to be used again. This storage can come in various forms, such as hard drives, solid-state drives, memory cards, or cloud-based solutions. Each type serves different purposes, whether it's providing quick access to temporary data or securely archiving important files for long-term use.

Concepts and Connections

CONCEPTS

Computing devices can store and retrieve different types of data, such as audio, images, and text. Data is saved in various formats and can be accessed or displayed through different devices. Students explore how devices collect, store, and display data.

CONNECTIONS

Within this grade level: At this grade level, students identify data formats used for various purposes, including audio, images, text, and video, and explore and identify computing devices that collect, store, and/or display data (1.DA.1).

Vertical progression: In Kindergarten, students gathered, recorded, and recognized patterns in data, understood its importance, and identified numeric and non-numeric data (K.DA.1). In Grade 2, students will collect and record numeric and non-numeric data and describe possible patterns, create questions that can and cannot be answered by the data, and analyze data to draw conclusions and make decisions (2.DA.1).

DIGITAL LEARNING INTEGRATION

- **K-2.CT** Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods, including those that leverage assistive technologies, to develop and test solutions.
 - A. Formulate problem definitions suited for technology assisted methods such as data analysis, modeling and algorithmic thinking in exploring and finding solutions.
 - B. Collect data or identify relevant data sets, use appropriate technologies to analyze them, and represent data in various ways to facilitate problem- solving and decision-making.

Opportunities for Computer Science Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students identify data formats used for different types of text, such as digital books, typed stories, and voice recordings of their reading.
- Students explore computing devices that store and display written work, such as tablets, e-readers, and word processors.

History and Social Science

- Students identify data formats used in history, such as historical photographs, digital maps, and recorded interviews.

Science

- Students explore and identify computing devices that collect and display scientific data, such as digital microscopes, weather sensors, and video recordings of experiments.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

B. FOSTERING COMPUTATIONAL THINKING PRACTICES:

1. Decompose Real-World Problems
2. Explore Common Features and Identify Patterns
3. Use Abstraction to Simplify, Represent, and Problem Solve
4. Apply Algorithmic Thinking to Problem Solve and Create

5. Apply Computational Thinking Practices to Select, Organize, and Interpret Data

C. FOSTERING ITERATIVE DESIGN PRACTICES:

1. Identify, Define, and Evaluate Real-world Problems
2. Plan and Design Artifacts
3. Create, Communicate and Document Solutions
4. Test and Optimize Artifacts

1.DA.2 The student will create representations of data to make predictions and draw conclusions.

- Collect and organize data with or without a computing device.
- Create tables, object graphs, picture graphs, and models using abstraction.
- Identify patterns and describe trends in data visualizations of various formats.
- Use data to answer questions, draw conclusions, and make predictions.

Understanding the Standard

The collection and use of data about individuals and the world around them is a routine part of life and influences how people live. Everyday digital devices can be used to collect and display data over time. Cell phones, digital toys, and cars can contain tools (such as sensors) and computers to collect and display data from their surroundings.

Students can create tables, object graphs, picture graphs, or models to visually represent information. This visual representation is easier to understand and interpret. These skills are essential in computer science, where data visualization plays a crucial role in analyzing trends and patterns. In the classroom, students apply these concepts through hands-on activities like surveys or experiments, where they collect data from their environment.

[1.DA.2a] Analyzing data requires the ability to recognize patterns and identify trends. Patterns are repeated sequences or structures that follow a defined rule, while trends indicate directional changes in data over time, such as growth or decline. To make accurate predictions and draw conclusions, it is essential to understand both. Students must be able to identify, describe, extend, generate, and apply repeating and increasing patterns using a range of representations. These skills are foundational for effective data analysis and informed decision-making. (Mathematics 1.PFA.1).

[1.DA.2b] Abstraction is focusing only on the important information while ignoring unnecessary details. It helps simplify complex ideas by highlighting what matters most. Imagine playing a computer game. When the "Start" button is clicked to play, players do not need to know all the complicated coding behind it. The players know that clicking "Start" will begin the game. The game designers used abstraction to hide the complicated details and only show the simple, easy-to-understand "Start" button.

Students will need to represent data in various ways. Consider the examples below:

Representations of Data	
Tables	A table in computer science is a way to organize and store information so that we can easily find and understand patterns or trends. By looking at the rows and columns, we can spot similarities (patterns) or changes (trends) in the data.

Object Graphs	In computer science, we use an object graph as a way to show how different things (called objects) are connected or related to each other. Imagine it like a map or picture where objects (like people, places, or things) are shown as dots or circles. The lines that connect the dots show how these objects are related. An object graph helps us see how these objects are connected and how they work together.
Picture Graphs	A picture graph (or pictograph) is a type of graph that uses pictures or symbols to represent data or information. Each picture or symbol in the graph stands for a certain amount of something. It is one way to represent information and make it easier to understand.
Models	A model is like a mini version or example of something bigger. It helps us understand how something works by showing us a simplified version of it. Think of a model as a tool we use to study, test, or explain things without needing the real, full version. Models let us notice patterns, test ideas, and predict outcomes before we use them in real life.

Each of these representations of data can help simplify complex real-world scenarios, enabling better analysis and decision-making.

[1.DA.2c] Patterns and trends both refer to regularities, but they differ in their scope and context. Patterns are repeated sequences or structures that can be observed within a specific dataset or context. For example, a pattern might be seen in a student's weekly sleep habits, such as sleeping more on the weekends than on weekdays. Trends, on the other hand, are long-term directions or movements in data or behavior that indicate a general tendency or shift over time. Trends often reflect broader, more gradual changes, such as a rise in the use of technology in education or an increase in global temperatures over several decades.

In short, patterns tend to focus on specific, repeatable occurrences, while trends look at broader, longer-term changes or directions.

[1.DA.2d] A prediction is an educated guess about what will happen next, based on what is already known. For example, if students play a game where pressing a button makes a character jump, they might predict that pressing the button again will make the character jump again.

A conclusion is a decision made based on interpreting data. It is the answer or result that is reached after thoroughly examining or testing the information. If a student tests pressing the same button in a game many times and the character always jumps, they can conclude that the button always makes the character jump.

Making predictions can help people guess what will happen next by using data to recognize patterns. Once the predictions have been tested and analyzed, people can draw conclusions to help them understand what the information means. Abstraction helps us simplify problems and make things easier to understand. We use it to break down complex tasks into simpler parts, so we can solve them more easily.

Concepts and Connections

CONCEPTS

Visual tools, such as charts and graphs help students to organize and represent data, enabling them to make predictions and draw conclusions. These graphical representations provide a visual format that allows for analyzing patterns, trends, and outcomes.

CONNECTIONS

Within this grade level: At this grade level, students collect and organize data with or without a computing device, create tables, object graphs, picture graphs, and models using abstraction, identify patterns and describe trends in data visualizations of various formats, and use data to answer questions, draw conclusions, and make predictions (1.DA.2).

Vertical progression: In Kindergarten, students created and interpreted data visuals like tables and graphs to answer questions, make predictions, and draw conclusions (K.DA.2). In Grade 2, students will create charts, graphs, and models using abstraction to represent data, analyze data visualizations to draw conclusions, and propose and evaluate a solution to a problem or question based on data and/or data visualization (2.DA.2).

ACROSS CONTENT AREAS

English

- **1.R** The student will conduct research and listen to a series of conceptually related texts on selected topics to build knowledge on grade-level topics or solve problems using available resources.
- **1.R.1C** Use templates to organize the information collected (e.g., charts, graphs).
- **1.R.1D** Use drawing, writing, or dictation to record facts and information collected from research.

Mathematics

- **1.PS.1** The student will apply the data cycle (pose questions; collect or acquire data; organize and represent data; and analyze data and communicate results) with a focus on object graphs, picture graphs, and tables. e) Organize and represent a data set by sorting the collected data using various methods (e.g., tallying, T-charts); f) Represent a data set (vertically or horizontally) using object graphs, picture graphs, and tables; and g) Analyze data represented in object graphs, picture graphs, and tables and communicate results: i) ask and answer questions about the data represented in object graphs, picture graphs, and table.

Science

- **1.1** The student will demonstrate an understanding of scientific and engineering practices by c) interpreting, analyzing, and evaluating data and d) constructing and critiquing conclusions and explanations. *1.1 standard is integrated within science content and not taught in isolation. Potential science concepts to apply 1.1 include: 1.2 (force,motion,energy), 1.3(matter), 1.6 (relationship between the sun and Earth)and 1.7(weather and seasonal changes).*

DIGITAL LEARNING INTEGRATION

- **K-2.CT** Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods, including those that leverage assistive technologies, to develop and test solutions.
 - A. Formulate problem definitions suited for technology assisted methods such as data analysis, modeling and algorithmic thinking in exploring and finding solutions.
 - B. Collect data or identify relevant data sets, use appropriate technologies to analyze them, and represent data in various ways to facilitate problem- solving and decision-making.
 - C. Break problems into component parts, extract key information, and develop descriptive models, using technologies when appropriate, to understand complex systems or facilitate problem-solving.
- **K-2 ID** Students use a variety of technologies, including assistive technologies, within a design process to identify and solve problems by creating new, useful or imaginative solutions or iterations.
 - C. Use appropriate technologies to develop, test, and refine prototypes as part of a cyclical design process.

Opportunities for Computer Science Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students collect and organize data about their favorite books or the number of times they read each day.
- Students create tables or picture graphs to represent class reading preferences or personal reading growth.
- Students identify patterns in a class reading log, such as which days they read the most.
- Students use reading data to predict which books their classmates might enjoy based on trends.

History and Social Science

- Students collect and organize data about different types of community helpers or local landmarks.

Mathematics

- Students create tables, object graphs, and picture graphs to represent mathematical data, such as how many students prefer different shapes or numbers.
- Students identify patterns in number sequences and describe trends in class surveys.
- Students use collected data to answer questions, such as which number appears most often in a set and make predictions about future results.

Science

- Students collect and organize data from simple experiments, such as plant growth over time or daily sunlight hours.
- Students create tables and models to track science observations.
- Students identify patterns in weather data and describe seasonal changes.
- Students use collected data to draw conclusions about an experiment and predict what might happen next.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

B. FOSTERING COMPUTATIONAL THINKING PRACTICES:

1. Decompose Real-World Problems
2. Explore Common Features and Identify Patterns
3. Use Abstraction to Simplify, Represent, and Problem Solve
4. Apply Algorithmic Thinking to Problem Solve and Create
5. Apply Computational Thinking Practices to Select, Organize, and Interpret Data

C. FOSTERING ITERATIVE DESIGN PRACTICES:

1. Identify, Define, and Evaluate Real-world Problems
2. Plan and Design Artifacts
3. Create, Communicate and Document Solutions
4. Test and Optimize Artifacts

[Back to Data and Analysis \(DA\)](#)

Impacts of Computing (IC)

1.IC.1 The student will describe how computing technologies impact daily tasks and communication.

- Determine when tasks should be completed with or without computing devices.
- Describe how computing devices are used in communication.
- Describe healthy habits for using computing technologies.

Understanding the Standard

Computing devices are useful for tasks such as research, communication, organization, creating digital media, and participating in educational activities that benefit from technology. Deciding whether to use a computing device depends on the nature of the task, the desired outcome, and the tools available. In some cases, non-digital approaches may be more effective—particularly for activities that involve physical movement, hands-on creativity, or personal interaction. Being intentional about tool selection helps ensure the task is completed efficiently and appropriately.

Online communication facilitates positive interactions, such as sharing ideas with many people, including friends and family around the world. It also allows opportunities for scientists, mathematicians, businesspeople, and many other professionals to communicate about projects they are working on together. People with similar interests can meet through social media or email and share information; however, the public and anonymous nature of online communication also allows intimidating and inappropriate behavior in the form of cyberbullying. Cyberbullying is a form of bullying that occurs when online communications are sent that are intimidating or threatening in nature.

[1.IC.1a] Table identifying daily routines and activities that can be completed with a computing device or without a computing device.

Routine/activity	With a computing device	Without a computing device
Waking up	Set an alarm on a smartphone or smart speaker.	Rise when the sun comes through the window.
Checking the weather	Use a weather app or website.	Go outside to check the weather.
Exercise	Follow a workout app or online video.	Exercising outdoors or at the gym.
Communing	Use a navigation app for directions.	Use a map or asking for directions.
Eating	Use a food delivery app or ordering online.	Prepare and eat at home.
Reading	Read digital books, articles, or blogs.	Read physical books, newspapers, or magazines.
Socializing	Video calls, social media, messaging apps.	Face-to-face interactions.

Work/study	Use computer applications, emails, or online platforms.	Write, read, or meet in person.
Shopping	Browse online stores and making purchases.	Go to a physical store or market.
Entertainment	Stream movies, TV shows, or play video games.	Play board games or outdoor activities.

[1.IC.1b] Computing devices are essential in modern communication, offering a variety of tools and platforms that allow people to connect with others.

Here are some examples of how computing devices improve modern communication:

- Email allows people to send messages instantly across the world.
- Video calls let people see and talk to each other in real-time, no matter the distance.
- Social media can help people share updates, photos, and videos with friends and family.
- Messaging apps allow people to send text messages, voice notes, and photos instantly.
- Online meetings enable businesses and schools to hold virtual meetings and classes, making remote work and learning possible.

Computing devices are used to support various modes of communication, including text, audio, video, and real-time interaction. Different technologies enable different ways to share information, collaborate, and connect with others. For example, email, video conferencing, instant messaging, and shared digital workspaces all rely on computing devices to function. Understanding how these devices facilitate communication across different formats is key to using them effectively.

Device	How It Is Used for Communication
Desktop Computers / Laptops	• Email • Video Calls • Social Media • Chatting
Tablets	• Email & Messages • Video Calls • Social Media
Smartphones	• Call • Text • Video Calls

	• Social Media & Notifications
Smartwatches	• Messages & Calls • Notifications

[1.IC.1c] Understanding when tasks should be completed with or without computing devices is important every day. It entails describing how computing devices are used for communication, such as sending messages, emails, or making video calls. Additionally, covering healthy habits for using computing technologies is important in daily life skills.

Healthy Habits for using Computing Technologies	
Set Time Limits	Encourage children to take breaks and limit screen time to prevent fatigue and promote balance between digital and physical activities.
Monitor Content	Ensure that children are accessing age-appropriate content using educational apps that support learning and creativity.
Encourage Social Interactions	Promote activities that involve communication with family and friends, either in person or through safe, monitored online platforms.
Promote Digital Citizenship	Teach children about respectful online behavior, privacy, and the importance of protecting personal information.

Concepts and Connections

CONCEPTS

The use of computing technologies requires responsibility and accountability. Students will explore the principles of digital citizenship, learning how to use technology safely, respect others online, and make responsible choices when communicating or accessing information. By building these habits early, students develop the skills they need to navigate online spaces with care and confidence.

CONNECTIONS

Within this grade level: At this grade level, students determine when tasks should be completed with or without computing devices, describe how computing devices are used in communication, and describe healthy habits for using computing technologies (1.IC.1).

Vertical progression: In Kindergarten, students explored how computing technologies are used in daily life, identified common devices, and discussed healthy usage habits (K.IC.1). In Grade 2, students will identify current uses of computing/emerging technologies and discuss how they impact society, compare and contrast appropriate and inappropriate online behaviors that apply in the physical environment and the online environment, and model healthy habits for using computing technologies (2.IC.1).

ACROSS CONTENT AREAS

English

- **1.C** The student will develop effective oral communication and collaboration skills to build a community of learners that process, understand, and interpret content together.
- **1.C.1A** Participate in a range of collaborative discussions (one on-one, in groups, and teacher-led) on grade one topics and texts. This includes: i) Listening actively and following agreed-upon rules for participating in discussions (e.g., waiting for a turn to speak without unnecessary interruptions and staying on topic); ii) Respectfully building on others' ideas and expressing their own clearly; iii) Asking questions to seek help, get information, or clarify information for further understanding; and iv) Expressing ideas and needs in complete sentences.

History and Social Science

- **1.1** The student will apply history and social science skills to the content by c. demonstrating curiosity and critical thinking through questioning; e. identifying similarities and differences to clarify and explain content; and f. recognizing cause-and-effect relationships.

DIGITAL LEARNING INTEGRATION

- **K-2.KC** Students critically curate a variety of digital resources using appropriate technologies, including assistive technologies, to construct knowledge, produce creative digital works, and make meaningful learning experiences for themselves and others.
D. Actively explore real-world issues and problems, develop ideas and theories, and pursue answers and solutions.
- **K-2.DC** Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act in ways that are safe, legal, and ethical.
B. Engage in positive, safe, legal, and ethical behavior when using technology, including social interactions online or when using networked devices.

Opportunities for Computer Science Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students determine when writing should be done by hand or with a computing device.
- Students describe how computing devices help people communicate, such as sending emails, video chatting, or listening to audiobooks.
- Students describe healthy habits for technology use, such as taking breaks when reading on a screen.

History and Social Science

- Students compare how people communicated in the past (e.g., letters, telegrams) versus today (e.g., emails, video calls) and create a timeline showing changes over time.
- Students explore how different cultures and communities around the world use computing technologies to communicate and complete daily tasks.
- Students can interview family members or school staff to learn how technology has impacted their daily lives and present their findings.
- Students will discuss the importance of balancing technology use with face-to-face interactions and compare this to historical periods when technology was not available.

Mathematics

- Students determine when to solve problems with or without a computing device, such as using mental math versus a calculator.

Science

- Students describe how computing devices help scientists communicate, such as sharing discoveries through videos or digital presentations.
- Students describe healthy habits for using technology during science, such as balancing screen time with hands-on experiments.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

D. FOSTERING DIGITAL LITERACY PRACTICES:

1. Responsible Use Practices
2. Safeguard Well-Being of Self and Others
3. Evaluate Resources and Recognize Contributions

1.IC.2 The student will describe tasks and activities that use screens and categorize them based on their use.

- a. Identify daily routines and activities that can be completed with or without screens.
- b. Classify the different uses of screen time as learning, entertainment, or communication.

Understanding the Standard

Screens are a common component of computing devices (not all), common examples like computers, smart watches, portable game consoles. They are used for a variety of tasks and activities that enhance learning and daily life. Screens serve as the primary way users view and interact with digital content. The size, resolution, and touch capabilities of a screen can impact how effectively a task is completed.

Examples of screen use include but are not limited to:

- At school, students may use screens for interactive learning through educational software and online resources. They may be used for research projects and access to communication tools that allow instant feedback and collaboration.
- At home, families may use screens for entertainment, or to manage household tasks, like online banking or shopping.
- At home, children may interact with educational apps that support their learning outside of school hours.
- At home, screens facilitate communication with friends and family through video calls or messaging apps.

By recognizing these various uses of screens, students can better understand their role in modern life and how to use them effectively and responsibly.

[1.IC.2a] A task is a specific action or set of actions that a computing device performs to achieve a user's goal. Tasks can be simple, like opening a file, or more complex, such as performing calculations or processing data. Computing devices execute tasks by following algorithms—structured sets of instructions. Automating tasks enables technology to help users work more efficiently and accurately.

An activity, on the other hand, refers to a collection of related tasks performed by a person, often for learning or enjoyment, such as reading a book or drawing a picture. Activities may involve multiple tasks carried out in sequence or combination. A routine is a series of activities or tasks performed regularly and often in a specific order, like a morning routine.

Task	Activity	Routine
A task is something the computer is asked to do. It is like giving the computer a job or a set of instructions.	An activity is a fun or interesting thing that can be done on a computer to help people learn more about how computers work. Activities often involve solving problems or playing games that make people think like a computer scientist.	In computer science, a routine is like a special task or a set of instructions that a computer follows to do something over and over again. It is a way of organizing work so that the computer knows exactly what to do without having to be told every single time.
Tell the computer to add 5 and 3 together. The task is to make the computer do the math and present the answer (8).	Using a software with block-based programming to make an animated story. People create the characters and tell the computer what to do, like "move this character to the right" or "make the character jump". This is an activity where the user practices giving instructions to the computer.	An example of a routine in computer science is a login function in a website. Each time a user enters their credentials, the computer follows a set of instructions (the routine) to check the username and password, grant access, and direct the user to the correct page, all without needing to be manually told each time.

[1.IC.2b] There are many routines and activities students participate in daily. Some of these can be completed with or without screens. Different uses of screen time are classified below.

Learning	Entertainment	Communication
Screen time for learning involves using digital devices to engage with educational content to supports academic development and skill-building.	Screen time for entertainment focuses on enjoyment and relaxation, providing entertainment through various digital platforms.	Screen time for community helps maintain relationships and build a sense of connection, even when physically distant.
• Online courses • Educational apps • Research information • Interactive simulations	• Movies • Video games • Music / videos • Live stream	• Video conferences • Message or chat • Email • Online forums or discussion boards

These are only some examples of screen time and how it can be used for learning, entertainment, or communication. Screen time often overlaps multiple categories, as many activities combine elements of education, social interaction, and recreation. Understanding the different ways screens are used can help promote balanced and purposeful technology use..

Concepts and Connections

CONCEPTS

Computing devices often have screens. Due to the common implementation of screens in many computing devices, it is important to recognize when and how they are used. Daily routines and activities can be completed with or without screens. Students should be able to identify and classify different uses of screens and make informed choices about when activities can be done with or without screens.

CONNECTIONS

Within this grade level: At this grade level, students identify daily routines and activities that can be completed with or without screens and classify the different uses of screen time as learning, entertainment, or communication (1.IC.2).

Vertical progression: In Kindergarten, students explored how different careers use computing technologies, their benefits, and local jobs that rely on them (K.IC.2). In Grade 2, students will discuss appropriate times and places for screen use, list and describe alternatives to screen time (2.IC.2).

ACROSS CONTENT AREAS

English

- **1 1.C** The student will develop effective oral communication and collaboration skills to build a community of learners that process, understand, and interpret content together.
- **1.C.1A** Participate in a range of collaborative discussions (one on-one, in groups, and teacher-led) on grade one topics and texts. This includes: i) Listening actively and following agreed-upon rules for participating in discussions (e.g., waiting for a turn to speak without unnecessary interruptions and staying on topic); ii) Respectfully building on others' ideas and expressing their own clearly; iii) Asking questions to seek help, get information, or clarify information for further understanding; and iv) Expressing ideas and needs in complete sentences.

History and Social Science

- **1.1** The student will apply history and social science skills to the content by c) demonstrating curiosity and critical thinking through questioning; e) identifying similarities and differences to clarify and explain content; and f) recognizing cause-and-effect relationships.

Opportunities for Computer Science Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students identify daily reading and writing activities that can be completed with or without screens, such as reading a printed book versus an eBook.
- Students classify screen-based literacy activities, such as watching an educational video (learning), typing a story (communication), or playing a word game (entertainment).

History and Social Science

- Students compare how people completed tasks before and after the invention of screens, such as reading physical maps versus using GPS or writing letters versus sending emails.
- Students will explore how different communities and cultures use screens for learning, entertainment, and communication, discussing similarities and differences.
- Students will create a timeline showing how screen-based technology has changed over time, from early televisions and computers to modern smartphones and tablets.
- Students will interview older family members or community members about their childhood screen use and compare it to their own experiences.

Science

- Students identify when to observe nature directly versus using a digital microscope or virtual simulation.
- Students classify screen-based science activities, such as watching an experiment (learning), recording data on a tablet (communication), or exploring a science game (entertainment).

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

D. FOSTERING DIGITAL LITERACY PRACTICES:

1. Responsible Use Practices
2. Safeguard Well-Being of Self and Others
3. Evaluate Resources and Recognize Contributions

1.IC.3 The student will compare and contrast ways people complete tasks with and without computing technologies.

- a. Identify tasks that can be completed with and without computing technologies.
- b. Discuss advantages and disadvantages of using and not using computing technologies.
- c. Describe how the appropriate use of computing technologies can improve efficiency.
- d. List computing technologies used in various careers.

Understanding the Standards

Computing technologies refer to a broad range of devices and tools designed to process, store, and transmit information. These tools assist users process information and perform tasks. These technologies enable students to access information, communicate, create content, and solve problems.

[1.IC.3a] In the classroom, computing technologies can enhance learning by providing interactive experiences through educational apps, online resources, and virtual learning environments. For example, students can use computing devices to conduct research, practice math skills with interactive games, or collaborate on projects using shared documents.

When a computing device is used appropriately (as intended) then it can make work faster and easier to complete. Appropriate use of computing technologies can improve efficiency by streamlining workflows, automating repetitive tasks, and enhancing communication. These tools enable faster data processing, organized information management, and seamless collaboration, allowing individuals and organizations to accomplish tasks more accurately and in less time. Computing devices can make certain tasks easier and faster, but not all tasks need to be automated using computing technologies.

Consider the following tasks that can be complete with or without computing technologies.

Task	With Computing Technology	Without Computing Technology
Communicating with someone far away	People can communicate instantly if they have a device that connects to the internet and allows texts, phone or video calls.	People can write letters to someone who lives far away, but communication will not be instantaneous.
Buying an item	People can pay with their device or through other payment applications.	People can pay with cash or check.
Reading	People can use e-books or read online with a device.	People can read from a physical book, newspaper, or other printed material.

Using computing technologies has both advantages and disadvantages, a few examples are listed below.

Advantages	Description
Doing tasks more efficiently	Efficiency of computing devices refers to how quickly and effectively they can perform tasks, process information, and solve problems. These devices can greatly enhance productivity by automating complex tasks, saving time and providing quick access to vast amounts of information.
Communication	Computers and phones make it easy for people to talk to each other, even if they are far away. They provide ways to share information, connect with others globally, and provide opportunities for collaboration.
Finding information	The internet provides quick access to a wealth of information. It helps people learn new things and can enable them to delve deeper into a specific topic.
Helping with routine tasks	Computers can handle simple tasks, such as sending reminders, organizing schedules, and managing daily activities, improving overall organization and time management.

Disadvantages	Description
Over-reliance on technology	Excessive use of computing devices can lead to dependency, making people feel unable to complete tasks or make decisions without the assistance of technology.
Cybersecurity	Devices connected to the internet are at risk of hacking, malware, and data breaches, which can compromise personal and sensitive information.
Automation and AI	As automation and artificial intelligence (AI) increase, there is a potential for job displacement, as machines replace human workers in various industries.

Computing technologies are the tools and devices that use computers to help people do tasks or solve problems. These include things like computers, tablets, smartphones, and even robots. They work by following instructions called programs or software, which tell them what to do. Computing technologies can help with many activities, such as learning, playing games, sending messages, creating art, or even exploring space. By using these tools, people can make their work more efficient.

[1.IC.3d] Computing technologies are found in many professions around the world. Consider the following examples:

Jobs	How Computing Technologies are Used
Doctors	Use computing devices to view medical images and to keep track of patient health records. Doctors can use diagnostic tools and robot-assisted surgery.
Teachers	Use computing devices to create lessons, show educational videos, and enhance classroom learning through interactive activities.
Artists	Use computing devices to design digital artwork, create animations, and produce graphics.

Builders	Use special software to design building plans and create 3D models, helping visualize structures before construction begins.
Scientists	Use computing devices to collect data from their experiments and to understand things like weather patterns or how plants grow.
Game designers	Use computing devices to design, build and program video games.
Farmers	Use technology, such as drones to help them know when to plant seeds or how much water to give their crops. This helps them grow food more easily.

Regardless of the profession students choose to pursue, they will use computing technologies in some capacity.

Concepts and Connections

CONCEPTS

Computing technologies are the tools and devices that enable the processing, storage, and transmission of information, helping people and businesses to solve problems, and connect in a digital world. The use of computing technologies has grown and is increasingly evident in many careers. These technologies support automation, data analysis, and digital communication and are widely used for decision making and to solve real-world problems.

CONNECTIONS

Within this grade level: At this grade level, students identify tasks that can be completed with and without computing technologies, discuss advantages and disadvantages of using and not using computing technologies, describe how the appropriate use of computing technologies can improve efficiency, and list computing technologies used in various careers (1.IC.3).

Vertical progression: In Grade 2, students will explain how computing technology is used in various careers, identify skills needed for careers that use computing technologies, and discuss how computing technologies have changed the workplace (2.IC.3).

ACROSS CONTENT AREAS

English

- **1.C** The student will develop effective oral communication and collaboration skills to build a community of learners that process, understand, and interpret content together.

Opportunities for Computer Science Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students identify ways to write a story with and without computing technologies, such as handwriting in a journal versus typing on a device.
- Students discuss the advantages and disadvantages of using a device for writing, such as faster editing versus the need for electricity.
- Students describe how spell-check and voice-to-text tools can improve writing efficiency.
- Students list computing technologies used by authors, journalists, and editors.

History and Social Science

- Students discuss the advantages and disadvantages of digital maps compared to paper maps.
- Students list computing technologies used by historians, archaeologists, and museum curators.

Mathematics

- Students identify ways to solve math problems with and without computing technologies, such as using mental math versus a calculator.
- Students discuss the advantages and disadvantages of using digital tools for math, such as quick calculations versus needing internet access.
- Students describe how spreadsheets and graphing tools can improve efficiency in organizing data.
- Students list computing technologies used by engineers, accountants, and architects.

Science

- Students identify ways to record observations with and without computing technologies, such as using a paper notebook versus a digital data tracker.
- Students discuss the advantages and disadvantages of using digital tools in science, such as virtual simulations versus hands-on experiments.
- Students describe how data collection software and digital microscopes improve efficiency in scientific research.
- Students list computing technologies used by meteorologists, biologists, and chemists.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

A. FOSTERING COMPUTING PRACTICES:

1. Building Relationships and Norms
2. Include Multiple Perspectives
3. Use Collaboration Tools

D. FOSTERING DIGITAL LITERACY PRACTICES:

4. Responsible Use Practices
5. Safeguard Well-Being of Self and Others
6. Evaluate Resources and Recognize Contributions

[Back to Impacts of Computing \(IC\)](#)

Networks and the Internet (NI)

1.NI.1 The student will explain that computing devices and the use of the Internet allow people the ability to gather information and connect with others.

- a. Describe how the Internet can be used to gather information.
- b. Explain ways people communicate using computing devices and the Internet.

Understanding the Standard

Computing devices and the internet work together and depend on each other to function effectively. Common computing devices, such as smartphones, laptops, or tablets, rely on the internet to access information, communicate, and use cloud-based applications. In return, the internet relies on computing devices to allow users to access, share, and interact with content, data, and services.

For example, a smartphone relies on the internet to retrieve emails, stream videos, or use online navigation services. The internet, in turn, depends on the smartphone to send requests for information and relay it back to the user in real-time. This interdependence enables seamless access to services and information from anywhere.

Computing devices are electronic tools that help us do a task using technology. These devices can receive, store and process data and help us connect with others.

[1.NI.1a] The Internet is a global network of interconnected computers and servers that communicate with each other to share information and resources. It enables users to access websites, send emails, participate in online communication, stream media, and use various services and applications from virtually anywhere in the world.

Many computing devices, such as tablets, smartphones, and computers, need to be connected to the Internet to access a wide variety of resources.

These resources describe how the internet can be used to gather information.

Resource	Description
Search engines	Users can enter keywords to find a wide range of websites and resources on a topic.
Websites	A collection of pages on the internet that people can visit using a web browser.
Databases	Using online databases (e.g., academic, scientific) to find structured data and studies.
Digital libraries	Accessing collections of digital books, journals, and other scholarly materials.

Educational platforms	Using online courses, tutorials, or webinars to learn about specific subjects.
-----------------------	--

[1.NI.1b] Online communication facilitates positive interactions, such as sharing ideas with many people, including friends and family around the world.

- Scientists, mathematicians, businesspeople, and many other professionals communicate and collaborate with one another on projects.
- Video conferencing allows for live digital conversations with family, friends, or colleagues, making long-distance communication feel more personal and engaging.

Here are some different methods of communication using computing devices and the Internet.

Method	Description
Email	Sending and receiving written messages, sometimes with attachments, for communication.
Instant messaging	Real-time, text-based conversations for quick exchanges.
Video conferencing	Virtual face-to-face meetings with audio and video capabilities.
Voice communication	Making calls or sending voice messages for audible communication.
Content sharing	Creating and distributing written or visual content to inform or engage others.
Collaborative tools	Real-time team collaboration on projects or documents.

The Internet can be used to gather information and help people communicate using computing devices. The Internet is a helpful tool for finding answers to questions, learning new things, and exploring different topics. People can use the Internet to gain access to search information on websites, watch videos, read articles, and much more. Additionally, computing devices like computers, tablets, and smartphones allow people to communicate in many ways, such as through emails, messages, video calls, or social media. These devices and the Internet help people stay connected and share ideas easily, whether for school, work, or fun.

Concepts and Connections

CONCEPTS

The Internet is a global network that connects devices and allows people to access and share information. One major part of the Internet is the World Wide Web, which people use to visit websites, watch videos, read articles, and communicate. The web helps users gather larger amounts of information quickly and interact globally.

CONNECTIONS

Within the grade level/course: At this grade level, students describe how the Internet can be used to gather information and explain ways people communicate using computing devices and the Internet (1.NI.1).

Vertical Progression: In Kindergarten, students learned how to communicate using connected devices and the Internet and understand its benefits (K.NI.1). In Grade 2, students will explore ways information is organized and share on the Internet, gather information from the Internet, and summarize collected information using their own words (2.NI.1).

ACROSS CONTENT AREAS

English

- **1.C** The student will develop effective oral communication and collaboration skills to build a community of learners that process, understand, and interpret content together.
- **1.C.1A** Participate in a range of collaborative discussions (one-on-one, in groups, and teacher-led) on grade one topics and texts. This includes: i) Listening actively and following agreed-upon rules for participating in discussions (e.g., waiting for a turn to speak without unnecessary interruptions and staying on topic); ii) Respectfully building on others' ideas and expressing their own clearly. iii) Asking questions to seek help, get information, or clarify information for further understanding; and iv) Expressing ideas and needs in complete sentences.

DIGITAL LEARNING INTEGRATION:

- **K-2.KC** Students critically curate a variety of digital resources using appropriate technologies, including assistive technologies, to construct knowledge, produce creative digital works, and make meaningful learning experiences for themselves and others.
B. Evaluate the accuracy, perspective, credibility, and relevance of information, media, data, and other digital sources.
- **K-2.CC** Students communicate clearly and express themselves creatively for a variety of purposes using appropriate technologies (including assistive technologies), styles, formats, and digital media appropriate to their goals.
A. Choose the appropriate technologies and resources for meeting the desired objectives of their creation or communication.

Opportunities for Computer Science Integration

Curriculum integration strengthens conceptual understanding and skill application. This can be done through multidisciplinary, interdisciplinary, and transdisciplinary approaches to integration. The examples below illustrate multiple ways to integrate computer science.

English

- Students describe how the Internet can be used to find books, listen to audiobooks, or watch educational videos.
- Students explain how people communicate using computing devices, such as video chatting or texting.

History and Social Science

- Students describe how the Internet can be used to explore historical events, view digital maps, or take virtual museum tours.
- Students explain how people use computing devices to communicate about history, such as watching interviews with historical figures.

Science

- Students explain how the Internet serves as a valuable tool for scientists by providing access to real-time data, enabling the study of weather patterns through satellite imagery, and offering live-streamed animal cams for observing wildlife behavior in its natural habitat.
- Students explain how scientists communicate using computing devices, such as sharing discoveries in online articles or presenting data in virtual meetings.

Skills in Practice

Students should engage in the following practices to deepen their conceptual understanding and enhance the application of skills aligned with the Computer Science *Standards of Learning*. These practices are explained in more detail in [Appendix A](#).

B. FOSTERING COMPUTATIONAL THINKING PRACTICES:

1. Decompose Real-World Problems
2. Explore Common Features and Identify Patterns
3. Use Abstraction to Simplify, Represent, and Problem Solve
4. Apply Algorithmic Thinking to Problem Solve and Create
5. Apply Computational Thinking Practices to Select, Organize, and Interpret Data

[Back to Networks and the Internet \(NI\)](#)

Appendix A

K-5 Computer Science Skills and Practices Continuum

Students develop essential practices: collaboration, computational thinking, iterative design, and digital literacy. Students use these practices to engage with core computer science concepts, create artifacts, and problem-solve across disciplines. Artifacts can include but are not limited to prototypes, programs, planning documents, animations, or abstractions (e.g. visualizations, storyboards, flowcharts, decision trees, models, computer simulations).

A. Fostering Collaboration in Computing Practices

1. Building Relationships and Norms:

- **K-2:** Students work collaboratively with others. Students take turns in different roles on the project.
- **3-5:** Students work collaboratively with others. Students practice assigning roles within their teams and recognize group member strengths.

2. Include Multiple Perspectives:

- **K-2:** Students differentiate their technology preferences from the technology preferences of others. Students will be presented with perspectives from people with different backgrounds, ability levels, and points of view.
- **3-5:** Students discuss design choices, compare preferences, ask questions, and seek input from group members with diverse abilities, experiences, and perspectives.

3. Create and Accept Feedback:

- **K-2:** With teacher scaffolding, students seek help and share ideas to achieve a particular purpose. Students ask questions of others and listen to their opinions.
- **3-5:** Students provide and receive feedback related to computing in constructive ways. For example, pair programming is a collaborative process that promotes giving and receiving feedback.

4. Use Collaboration Tools:

- **K-2:** Students collaboratively brainstorm by writing on a whiteboard or paper.
- **3-5:** Students use collaboration tools to manage teamwork and utilize online project spaces. They also begin to make decisions about which tools would be best to use and when to use them.

Instructional Considerations for Collaboration Practices

Possible **instructional approaches** to foster collaboration practices:

1. Design instruction around authentic problems that require collaboration. Assign roles, provide clarifying and probing question stems, and model strategies students can use to identify and advocate for their needs.
2. Provide resources to support exploring different viewpoints and end users. Model curiosity, perspective-taking, and empathy.
3. Model sentence stems for constructive feedback, establish routines for self and group reflection, and practice incorporating diverse viewpoints. Implement pair programming with opportunities to practice giving and receiving feedback.
4. Model tool selection and project management structures. Provide opportunities to practice various methods and reflect.

Instructional activities may include but are not limited to:

- **Classroom Discussion:** Organize discussions that engage students in hearing differing perspectives.
- **Timeline Creation:** Have students create or evaluate and modify timelines that illustrate the steps needed to complete a task as a group.
- **Simulated Shark Tank Innovation Challenge:** Create an innovation design challenge where students collaboratively apply computer science content to solve a problem or launch a new idea.
- **Case Studies:** Provide case studies of design decisions that real computer scientists face and have students analyze and present their recommended choices based on computer science content knowledge.

B. Fostering Computational Thinking Practices

1. Decompose Real-World Problems:

- **K-2:** Students break problems, information, and processes into parts. Identify relationships and connections among parts. Reflect on how decomposition aids problem-solving across contexts.
- **3-5:** Students further break problems into subproblems, apply systems thinking to explore interdisciplinary connections and integrate existing solutions or procedures (i.e. classroom processes, math procedures, school routines) Apply algorithms to break a problem into subtasks that can be solved and combined to solve the main problem.

2. Explore Common Features and Identify Patterns:

- **K-2:** Students will be able to identify and describe repeated sequences in data or code through analogy to visual patterns or physical sequences of objects. Students will identify patterns, such as recognizing repeated patterns of code that could be more efficiently implemented as a loop.
- **3-5:** Students analyze patterns to develop generalizations and models, test their limits, and validate inputs. Use patterns to analyze trends, justify design decisions, and create artifacts.

3. Use Abstraction to Simplify, Represent, and Problem Solve:

- **K-2:** Students use and/or create abstractions (e.g. storyboards, flowcharts, decision trees, models) to simplify problems, represent information, organize thinking, communicate, and create artifacts. Artifacts can include but are not limited to prototypes, programs, planning documents, and animations.
- **3-5:** Students use and/or create abstractions (e.g. visualizations and computer simulations) to simplify problems, represent information, organize thinking, communicate, and create artifacts. Students intentionally use abstractions to support the problem-solving process to aid in understanding, planning, and predictions.

4. Apply Algorithmic Thinking to Problem Solve and Create:

- **K-2:** Students use algorithmic thinking to develop a sequence of steps to plan, create, test, and refine artifacts with and without technology.
- **3-5:** Students use pseudocode and generalizations to organize, create and seek and incorporate feedback on more complex designs.

5. Apply Computational Thinking Practices to Select, Organize, and Interpret Data:

- **K-2:** Students use computational thinking to organize data and make predictions. Explore parts and relationships within data sets.
- **3-5:** Students visualize data. Use patterns and algorithmic thinking to organize data, identify trends, and make predictions. Use decomposition to explore parts and relationships within data sets. Ask questions about available data sources, and compare and analyze test results to inform decisions, plan, and refine designs.

Instructional Considerations for Computational Thinking Practices

Possible **instructional approaches** to foster computational thinking practices:

1. Model strategies for breaking complex information into smaller parts. Provide opportunities to analyze and discuss the relationship among parts.
2. Support students with recognizing patterns. Model how to analyze, interpret, and display patterns to make predictions and draw conclusions.
3. Model the use of abstraction (e.g. visualizations, storyboards, flowcharts, decision trees, models, computer simulations) to simplify problems; represent information (e.g. data, patterns, processes, phenomena, systems); organize thinking; and support sense-making. Support students with creating and evaluating abstractions and their limitations.
4. Plan opportunities for students to use sequencing in problem solving, incorporate user feedback, and check for bias, accessibility, and other design criteria. Model ways to systematically test, validate, evaluate, refine, and optimize algorithmic solutions. Provide opportunities to reflect on how algorithms are used in solutions.
5. Model abstraction, pattern analysis, and decomposition. Use models to develop and test predictions. Identify limitations and benefits of models.

Instructional activities may include but are not limited to:

- **Create an Artifact:** Students could create an app, program, animation, simulations, etc. to solve a community problem or creatively express an idea.
- **Identify Patterns to Make Predictions:** Students notice repetition in sequences of numbers or parts of a process to make predictions about future events or missing components.
- **Create Abstractions:** Students choose the best tool to use for problem solving using abstractions. Discuss which tools worked best for the team and the problem. Tools may include models, visualizations, storyboards, flowcharts, decision trees, generalizations, simulations.
- **Create Models:** Develop models to represent information such as patterns, relationships, inputs/outputs. Create models of systems (e.g. model networks, cybersecurity, emerging technologies) to understand how parts connect to perform a function. Students can create models to engage in systems thinking and modularization.
- **Evaluate existing models and programs:** Evaluate outputs for bias, accessibility, reliability or other established design criteria. Students can identify applicable parts or modules of existing programs and reuse to solve different problems.
- **Reflection and Transfer:** Have students reflect on how each computational practice facilitates problem-solving and identify opportunities to apply the practice to other situations. Support students identify key points in feedback.

C. Fostering Iterative Design Practices

1. Identify, Define, and Evaluate Real-world Problems:

- **K-2:** With guidance from an educator, identify, define, and explore existing problems and potential solutions. Ask questions to view problems from different perspectives.
- **3-5:** Students identify, define, and explore existing problems and potential solutions. Ask questions to understand problems from different perspectives. Clarify success criteria, identify constraints, and uncover missing information. Explore patterns and develop generalizations about the types of problems that benefit from computational solutions.

2. Plan and Design Artifacts:

- **K-2:** With guidance from an educator, students will generate ideas for new solutions, incorporate peer feedback, and reflect on impact of diverse perspectives. Use tools like class or group discussions, outlines, flowcharts, and storyboards to plan prototypes.
- **3-5:** Students will generate ideas for new solutions, incorporate peer feedback, and reflect on the impact of diverse perspectives. Use tools like outlines, flowcharts, and storyboards to plan prototypes. Predict the performance and impacts of prototypes, including potential errors, user needs, and accessibility.

3. Create, Communicate and Document Solutions:

- **K-2:** Students create artifacts with or without technology, such as algorithms and programs using plans and outlines. Describe design choices and make connections to the design challenge, criteria, and constraints. Engage in giving and receiving feedback enhances communication skills.
- **3-5:** Students create artifacts, such as algorithms and programs using plans and outlines. Describe design choices and make connections to the design challenge, criteria, and constraints. Engage in giving and receiving feedback to refine solutions and enhance communication skills.

4. Test and Optimize Artifacts:

- **K-2:** Students test artifacts to ensure they meet criteria and constraints, comparing results to intended outcomes. Use computational thinking and other problem-solving strategies like trial and error to fix simple errors, debug, revise, and evaluate artifacts against design criteria.
- **3-5:** Students test artifacts to ensure they meet criteria and constraints, comparing results to intended outcomes. Use computational thinking and other problem-solving strategies like trial and error to fix simple errors, debug, revise, and evaluate artifacts against design criteria. Reflect on how the iterative design and computational thinking practices facilitate program development.

Instructional Considerations for Iterative Design Practices

Possible **instructional approaches** to foster iterative design practices:

1. Design learning experiences where students identify real-world problems and evaluate the appropriateness of using computational tools to develop solutions.
2. Provide instructional time and model strategies to support students with using an iterative process to plan the development of an artifact while considering key features, time and resource constraints, and user expectations. Design instructions to provide students with multiple paths to solve problems.
3. Provide instructional time for students to prototype, justify, and document computational processes and solutions using iterative processes. Model how to listen to differing ideas and consider various approaches and solutions.
4. Provide instructional time and model strategies for evaluating artifacts using systematic testing and iterative refinement to enhance performance, reliability, usability, and accessibility as outlined in the design criteria.

Instructional activities may include but are not limited to:

- **Class Discussions:** Discuss the pros and cons of using computing technologies to solve real-world problems. Consider examples like drones monitoring the environment; AI-generated art; or personalized learning applications. Progressive examples include, using machine learning in self-driving cars to interpret road conditions and make decisions, and robots assisting in surgeries for precision and reduced recovery times.
- **Prototype and Improve:** Create simple animated stories, solve pre-existing problems, and utilize coding platforms to simulate solutions. Incorporate available technology to develop physical models. Use peer feedback to refine designs, and document changes while justifying improvements at each step.
- **Debug and Enhance:** Work with a pre-built program containing intentional errors and limited features to debug to optimize the program for performance and enhance it with new capabilities.
- **Accessibility Upgrade:** Emphasize empathy and inclusion in design by analyzing an existing program or interface (e.g., a basic website). Evaluate it for usability and accessibility. Propose iterative changes to improve the design, such as adding features like text-to-speech, adjustable font sizes, or simplified navigation and implementing when available and appropriate.

D. Fostering Digital Literacy Practices

1. Responsible Use Practices:

- **K-2:** Students use technology in ways that are safe, legal, and ethical. Implement strategies to protect their digital identity, personal data, and the data of others.
- **3-5:** Explore and ask questions about how computer science and emerging technologies work, and their benefits and risks. Students explore data privacy rights, data protections, terms of service and privacy policies. Weigh tradeoffs and risks with actions and decisions involving computer science.

2. Safeguard Well-Being of Self and Others:

- **K-2:** Students reflect on their emotional response to the use of digital technology. Consider how the use of technology can impact others and make choices that benefit others and avoid harm. Identify the roles and responsibilities of humans in designing and using technologies. Practice empathy and engage in positive online practices as an upstander.
- **3-5:** Students reflect on their emotional response to the use of digital technology and identify how to use technology in ways that support personal well-being. Consider how the use of technology can impact others and make choices that benefit others and avoid harm. Identify the roles and responsibilities of humans in designing and using technologies. Practice empathy and engage in positive online practices as an upstander.

3. Evaluate Resources and Recognize Contributions:

- **K-2:** Students apply strategies for evaluating the accuracy, validity, accessibility, reliability, appropriateness, credibility, and relevance of digital sources.
- **3-5:** Students apply strategies for evaluating the accuracy, validity, accessibility, reliability, appropriateness, credibility, and relevance of digital sources. Keep track of sources of information and give credit the creators of information. Students evaluate the bias and relevance of sources. Identify false or misleading information.

Instructional Considerations for Digital Literacy Practices

Possible **instructional approaches** to foster digital literacy practices:

1. Model how to use technology in ways that are safe, legal, and ethical. Model how to make decisions about data privacy and information sharing that protect individual and peer identify and digital footprint. Incorporate learning activities like discussions of digital dilemmas that help students explore different perspectives, benefits, risks, and tradeoffs.
2. Incorporate opportunities for students to reflect on possible positive and negative impacts of how they use computing technologies. Choose instructional technology that aligns with learning goals and use data on students learning to reflect on and assess the extent to which the technology is supporting learning outcomes. Provide opportunities to identify the role of humans in developing and using technology.

3. Model strategies for how to investigate the credibility of information sources and give appropriate attributions for content created by others.

Instructional activities may include but are not limited to:

- **Source Evaluation:** Assign students articles. Have students distinguish between fact and opinion within article and evaluate the reliability of the sources.
- **Comparative Analyses:** Encourage students to explore ethical dilemmas, compare different approaches to data privacy and possible impacts across different time periods using evidence to support arguments.
- **Class Discussions:** Organize discussions where students take roles representing different perspectives and defend their positions.
- **Digital Dilemmas:** Discuss case studies of complex topics that do not have one right answer such as the CommonSense Education digital dilemmas.

Appendix B

Grade 1 Computer Science Vocabulary

Term	Definition
Abstraction	A filtering process used to create a simplified representation of relevant data to identify essential details, excluding less important details.
Acceptable Use Policy (AUP)	Rules and guidelines that define safe practices and responsible use of technology.
Algorithm	Finite and specified set of step-by-step instructions designed to solve a problem or perform a task.
Algorithmic Thinking	Process of developing algorithms in a logical, systematic, and procedural way to solve problems or complete tasks.
Attribute	Characteristic or quality that helps us describe and differentiate objects or data (i.e. color, size, shape, weight, position, number, or texture).
Author	The creator of a book, image, song, or object.
Basic Functionality of Classroom Devices	May include turning them on and off, navigating simple menus, adjusting volume, and troubleshooting common minor issues.
Block-Based Programming	A visual drag and drop programming tool that users can use to create programs using command blocks.
Character	A person or animal in a book, story, movie, or project. A letter, number, or symbol used in a password.
Code	Any set of instructions expressed in a programming language.
Computational Artifacts	Any creation made by a human using a computing device. It can include but are not limited to prototypes, programs, planning documents, animations, or abstractions (e.g. visualizations, storyboards, flowcharts, decision trees, models, computer simulations).
Computational Thinking	A logical and systematic problem-solving process that uses decomposition, pattern recognition, abstraction, and algorithm thinking to foster creativity and develop solutions.
Computer	An electronic computing device that processes, stores, and retrieves data and capable of executing a wide range of tasks, from basic calculations to complex data processing.
Computer Science	The study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society.

Computing Device	An electronic device that can receive input, process data, store information, and produce output based on instructions (programs).
Computing Technologies	Broad range of devices and tools that help us process information and perform tasks using computers and software.
Data	Individual pieces of information about people, things, or events that can be processed, stored, and analyzed by computing devices.
Data Representation	How data is visually represented, such as in graphs or charts.
Data Visualization	The representation of data through use of common graphics, such as charts, plots, infographics and even animations to make complex data more accessible and understandable.
Debug	Process of identifying, isolating, and fixing errors (often referred to as "bugs") in a set of instructions, code, or system. This can also include hardware and software.
Decomposition	Process of breaking down a problem, process, or task into smaller, more manageable components.
Design	Creation of a plan or prototype of a proposed solution.
Design Document	A detailed plan that outlines the structure, features, and implementation strategy of a project. It serves as a blueprint, providing clear specifications, goals, and guidelines for developers, designers, and stakeholders. Design documents often include diagrams, technical requirements, workflows, and rationale to ensure a shared understanding of the project's direction and execution.
Desktop Computer	A stationary computing device.
Device Care	Includes basic maintenance, such as handling devices carefully and setting clear rules for use, while fostering respect and responsibility.
Digital Citizenship	The rights, responsibilities, and opportunities of living, learning, and working in an interconnected digital world. This promotes responsible and ethical behavior in digital environments, including understanding data privacy, security, and the impact of digital actions.
Digital Literacy	The ability to use technology effectively and responsibly to access, evaluate, create, and communicate information.
Digital Sharing	Process of sending or posting information, pictures, or videos from one person's connected device to another, or on the Internet where others can see it.
Email	Program used to communicate online
Flowchart	A diagram that shows the steps in a process using shapes and arrows. It helps the user visualize how things happen in order.

Function	Like variables, except instead of storing data they store lines of code. Help to simplify the programming process and make code more readable.
Hardware	The physical parts of a computing device.
Healthy Screen Habits	Practices that emphasize balanced use of digital devices to support physical, mental, and emotional well-being.
Information	Facts provided or learned about something or someone.
Input	Information or action you give to a computer or device to tell it what to do.
Input Devices	Hardware components that allow users to enter data into a computing device.
Internet	A global network of interconnected computing devices that allows devices to share information and resources.
Key	A distinct identifier used to differentiate data elements within a set.
Keyboard	An input device used to enter text on a screen.
Laptop Computer	A mobile computing device.
Library	A collection of books and periodicals.
List	A data structure that stores an ordered collection of elements, which can be of any type (numbers, strings, objects, etc.).
Memory	Physical storage in computing devices where data is processed and instructions for processing are stored. Memory types include RAM (Random Access Memory), ROM (Read-Only Memory), and secondary storage like hard drives, removable drives, and cloud storage.
Messaging App	It is a program on a connected device that lets people send written messages, pictures, or videos to each other.
Model	A simplified representation of an idea, object, system, or process that helps describe, test, or predict how something works often using diagrams, simulations, or code.
Mouse	An input device used to move items on the screen and navigate.
Non-numeric Data	Refers to data that involves categories, qualities, or descriptions rather than numbers such as name, address, and favorite color.
Numeric Data	Refers to data that involves numbers and can be counted, measured, or quantified such as age, weight, or height.
Object Graph	A graph that uses concrete materials to represent the categorical data that is collected.
Output	Results of a computing device after it has followed given instructions.

Output Devices	Hardware components that display processed data or information.
Pattern Recognition	Ability to identify commonalities, similarities, or differences in recurring elements.
Personal Information	Data or information about a person that relates to their identity, characteristics, or activities.
Picture Graph	A type of graph that use pictures or symbols to represent and compare information.
Prediction	Making a guess of what will happen based on current facts or information.
Printer	An output device that displays the result on paper.
Private Information	Sensitive data or information that can identify a person or give others access to your personal life.
Problem Definition	Clearly identifying the problem or challenge that needs to be solved.
Program	The implementation of an algorithm (set of instructions) translated into a programming language that a computer can follow and execute to perform a specific task.
Programming Language	A structured system for writing instructions that a computer can understand and execute. It includes syntax, which defines the rules for how code is written, and semantics, which conveys the meaning of the instructions. Programming languages enable developers to build software, automate processes, and control computer hardware.
Public Information	Information that is okay to share with anyone and is typically available for everyone to see.
Protect	To keep safe.
Purposeful Use of Computing Devices	Includes the understanding that technology has specific purposes, such as learning new things, solving problems, and communicating.
Reboot	To turn off the device and turn it back on.
Selection	Using conditions to manage the sequence of a program's execution.
Sequence	The specific order in which instructions or steps are executed in an algorithm or program.
Simulation	Replicating the behavior of a real-world process or system over a period of time.
Social Media	Applications that allow people to socialize, communicate, and share content with each other.
Software	A set of instructions that tells the computer how to act and respond but cannot be seen or touched.
Sort	Comparing a set of objects in order to find similarities and differences, so that they may be arranged and organized.
Storage	Location where data, programs, and files are kept permanently (until deleted).
Table	A structured format to organize and record information in rows and columns.

Texting/text Message	A short, written message sent from one phone to another over the Internet or a mobile network.
Trends	Are long-term directions or movements in data or behavior that indicate a general tendency or shift over time.
Troubleshoot	Processes used to diagnose why a system or process is not working as expected and systematically testing solutions to resolve the issue.
Username	A unique name that people use to log into a device or online account. It is like a nickname that helps the computer recognize who is logging in.
Video Call	A live conversation where people can see and hear another person while talking through a connected device, like a tablet or computer.
Visualization	Refers to graphical representations of data or information that help users understand patterns, trends, and relationships more effectively. Visualizations make complex data more accessible, interpretable, and actionable.
Website	A location on the Internet that is accessed using a web address (URL) and contains one or more connected pages of information.
Wi-Fi	The device that allows computing devices to access the Internet without being connected to physical cables within a specific area using radio waves to send and receive data.
World Wide Web (WWW)	A system of interconnected web pages that users can access through the internet.