Computer Science Implementation Framework

Grade 3

2026

Table of Contents

Background

Louisiana Act 541 (2022), the Computer Science Education Act, aimed to establish a comprehensive and seamless statewide computer science education program across all educational levels to meet the demands of a dynamic economy. To achieve this goal, Act 541 created the Computer Science Education Advisory Commission (CSEAC). CSEAC was responsible for advising the State Board of Elementary and Secondary Education (BESE) through the state Department of Education (LDOE) to develop and implement a state action plan for delivering education in computer science in all public schools.

The first key action within the Louisiana K-12 Computer Science Education Plan calls for establishing content standards for computer science. From May to August 2024, the K-12 Computer Standards Writing Committee drafted student-centered and developmentally appropriate computer science standards within defined grade bands: elementary (K-5), middle school (6-8), and high school (9-12). These standards were carefully designed to reflect the increasing complexity and depth of understanding expected as students progress through each educational stage, with a focus on analytical and critical thinking skills, digital literacy, digital citizenship, and technology skills. In October 2024, BESE approved the K-12 Louisiana Student Standards for Computer Science. These standards now serve as a guiding framework to support high-quality computer science instruction in Louisiana.

To support standards implementation, the Louisiana K-12 Computer Science Education Plan also calls for the development of computer science frameworks for standards implementation, which show the overlap with existing content standards.

Purpose of the Frameworks

The frameworks support Key Action 1 of the Louisiana K-12 Computer Science Education Plan by providing guidance for integrating the K-12 Louisiana Student Standards for Computer Science into existing instruction, emphasizing the connections between computer science, math, and science. By utilizing the frameworks, teachers will equip students with essential analytical thinking, digital literacy, and technology skills needed for success in society and future career opportunities.

Connections between Computer Science, Mathematics, and Science

A key feature of the Computer Science Implementation Frameworks highlights the connections between the K-12 Louisiana Student Standards for Computer Science and mathematics and science instruction. Many computer science concepts, such as sequencing, patterns, and logic, align with mathematical reasoning and scientific inquiry. Computational thinking strategies, such as decomposition and algorithmic thinking, support problem-solving approaches in math and science. Integrating computer science within mathematics and science helps students see connections between the content areas, encouraging students to view technology as a tool for solving real-world problems in a variety of professional fields.

Finding Connections to Core Content Areas

In elementary education, computer science implementation emphasizes integrating computer science across content areas. Computer science standards provide the foundation for teachers to identify and incorporate cross-disciplinary computer science content opportunities within lessons and content skills.

Unpacking the Standards

Computer science standards are essential for effective computer science implementation. When considering which computer science standards best fit a lesson, summarize the standard's ultimate purpose or goals by asking, “What should students accomplish by the end of grade 3 to master this standard?” Identifying the underlying skill or content required to master the standard helps in determining its best fit within a specific content area.

Finding Connections

Connecting standards to the real world makes standards more meaningful and relevant to students. Making connections between computer science standards and content or skills in other content standards demonstrates the real-world applications of computer science. Connections may be found between computer science core practices and practices of another content area. Computer science skills also reinforce skills emphasized in other content areas. For a full explanation of the core concepts and core practices of the computer science standards, see the K-12 Louisiana Student Standards for Computer Science.

Computational Thinking Skills

Computational thinking involves breaking down complex problems into smaller, manageable parts (decomposition), recognizing patterns (pattern recognition), creating step-by-step solutions (algorithms), and generalizing solutions to other problems (abstraction). These skills are not exclusive to computer science; they are applicable in other content areas, including math and science. By understanding computational thinking, teachers can identify opportunities to apply these problem-solving strategies in multiple contexts.

Computational Thinking Practice	In Mathematics	In Science
Abstraction	Represent concepts using visualizations such as graphs, charts, or diagrams.	Create simplified simulations of complex scientific phenomena.
Algorithmic Thinking	Create step-by-step instructions for solving problems or geometric constructions.	Create models of scientific processes like the water cycle or food chain.
Decomposition	Break down multi-step word problems into smaller, manageable parts.	Break down complex investigations into smaller, testable hypotheses.
Pattern Recognition	Identify number patterns, geometric patterns, or patterns in data sets.	Identify patterns in data, such as weather, animal behavior, or plant growth.

Components of the Computer Science Implementation Framework

Complexity Statements

Complexity statements clarify the increasing depth and rigor of learning expectations within a content standard as students progress through each grade level. These statements break down each standard into a vertical progression of learning within the grade band and ensure an age-appropriate level of knowledge for each standard. Standards and complexity statements allow flexibility and guidance to implement instruction at each grade band and each grade level, depending on a particular school's or system's implementation plan.

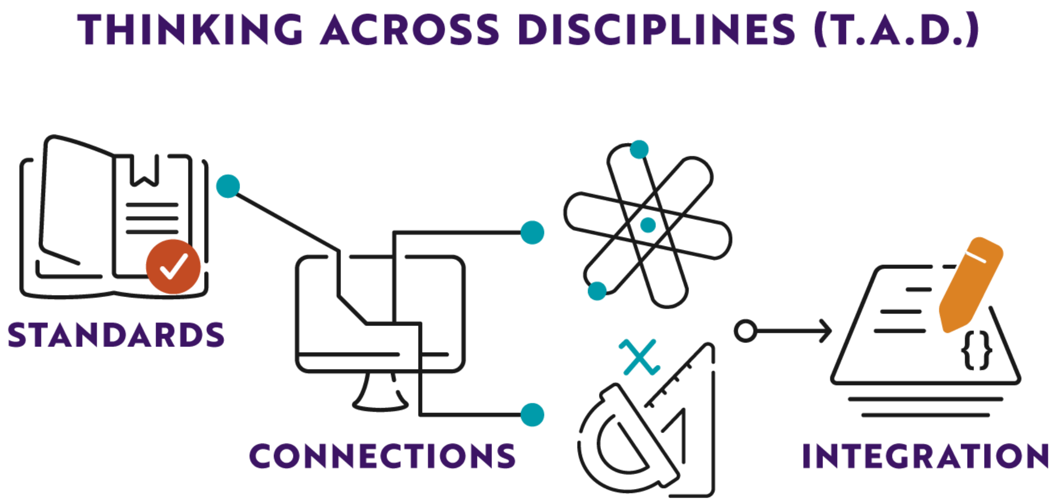
Complexity statements are foundational for effectively integrating computer science by providing a clear, grade-level-specific model for instruction. By outlining the increasing depth and rigor of learning expectations across grade levels, these statements ensure computer science concepts are introduced and reinforced in a developmentally appropriate manner, preventing both over- and under-challenging students.This clarity in learning goals empowers teachers to create integrated lessons that are both effective and appropriately challenging, ultimately fostering a deeper, more interconnected understanding across all disciplines.

Example

E.NI.1A. Explain how networks connect computers to other computing systems and the internet.	
Grade	**Complexity Statement**
K	Define what a network is and how it is used to connect to other people and places around the world.
1	Compare and contrast wireless versus plugged-in networks.
2	Model and describe how computers break down data to share on networks.
3	**Explain how data is transmitted in packets.**
4	Create a model demonstrating how information is shared within a network.
5	Diagram and summarize how data can be shared between multiple networks and users via the Internet.

Thinking Across Disciplines

Thinking Across Disciplines (T.A.D.) is a systemic approach guiding the integration of computer science concepts and practices into other content areas.



The T.A.D. Process

Initially, teachers gain a thorough understanding of the computer science standards for their grade band and the overarching core practices. Next, teachers identify meaningful connections between the standards and practices of computer science and those of other content areas, such as math and science. Finally, teachers create or modify existing lessons and activities that blend computer science concepts with existing instruction.

Benefits of T.A.D.

This interdisciplinary approach makes learning more relevant and engaging for students by demonstrating the practical applications of computer science in real-world contexts. It also reinforces critical thinking, problem-solving, and computational thinking skills, which transfer across multiple disciplines. The computer science implementation frameworks contain examples of the T.A.D. process as a starting point for teachers to discover their own connections between computer science and other content areas.

Core Concept 1: Computing Systems (CS)

Overview

Students interact with a wide variety of computers each day. Computers are devices that can collect, store, analyze, and act upon information in ways that can positively and negatively affect human capabilities. Humans (users) operate, program, and maintain the physical components (or hardware) and instructions (or software) that make up a computer. The hardware, software, and users are collectively called computing systems. Users leverage their understanding of hardware and software to resolve problems within computing systems in a process called troubleshooting.

Elementary Standards for Core Concept 1: Computing Systems

Subconcepts and Core Practices	Elementary (K-5)
Subconcept 1: Hardware and Software *Core Practices: Collaborating Around Computing; Recognizing and Defining Computational Problems; Communicating about Computing*	E.CS.1A. Identify and select the appropriate hardware to complete computing tasks.
	E.CS.1B. Identify and select the appropriate software to complete computing tasks.
	E.CS.1C. Evaluate hardware and software types to meet users' needs in completing various computing tasks.
Subconcept 2: Troubleshooting *Core Practices: Collaborating Around Computing; Recognizing and Defining Computational Problems; Testing and Refining Computation Artifacts; Communicating about Computing*	E.CS.2A. Propose potential ways to address computing problems using appropriate hardware or software.

Complexity Statements

Core Concept 1: Computing Systems

Complexity statements for the standards included in **Core Concept 1: Computing Systems** break down each standard into manageable, age-appropriate components. This breakdown ensures that instruction fits the developmental needs of students at each grade level. For instance, kindergarten focuses on describing the basic uses of hardware and software, but by grade 5, students justify and apply knowledge of hardware and software to navigate complex scenarios. Grade 5 students might analyze a scenario where a computer program runs slowly and propose a plan to troubleshoot whether the issue stems from the hardware (like insufficient memory) or the software (like too many applications running).

E.CS.1A. Identify and select the appropriate hardware to complete computing tasks.	
Grade	**Complexity Statement**
K	Describe the ways people use hardware to perform computing tasks (e.g., keyboard, mouse, monitors, laptops, tablets).
1	Explain how common hardware pieces function.
2	Compare and contrast the functions and uses of various hardware.
3	**Identify the appropriate hardware to complete a given task.**
4	Analyze the capabilities and limitations of hardware for a particular use in a given scenario.
5	Justify and use the proper hardware to complete a given task.

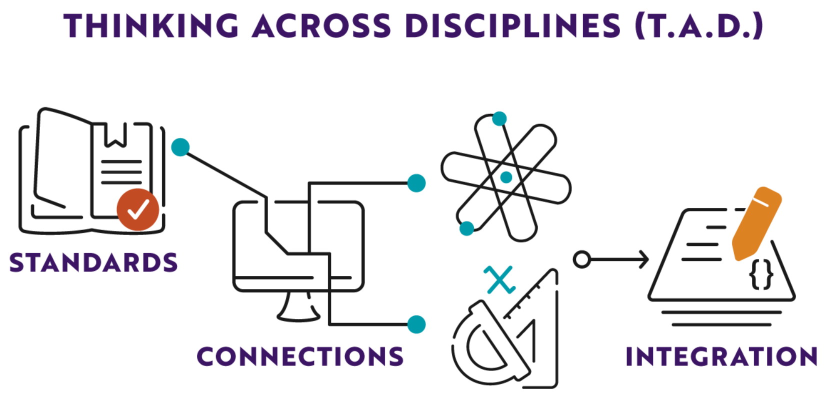
E.CS.1B. Identify and select the appropriate software to complete computing tasks.	
Grade	**Complexity Statement**
K	Describe the ways people use software to perform computing tasks.
1	Explain the function of common software components.
2	Compare and contrast the functions and uses of various software.
3	**Identify the appropriate software to complete a given task.**
4	Analyze the capabilities and limitations of software for a particular use in a given scenario.
5	Justify and use the proper software to complete a given task.

E.CS.1C. Evaluate hardware and software types to meet users' needs in completing various computing tasks.	
Grade	**Complexity Statement**
K	Identify ways that computing devices help users accomplish simple tasks.
1	Identify the basic hardware and software used to meet users' needs for common tasks.
2	Describe the function of the hardware and software in accomplishing a specific task when given a scenario.
3	**Describe the limitations of certain computing devices in given scenarios.**
4	Determine the most appropriate hardware and software to accomplish a task when given a scenario.
5	Evaluate a user's unique needs and recommend the most appropriate hardware and software in a given scenario.

E.CS.2A. Propose potential ways to address computing problems using appropriate hardware or software.	
Grade	**Complexity Statement**
K	Identify a basic hardware problem.
1	Identify a basic software problem.
2	Differentiate between a hardware problem and a software problem using observations.
3	**Describe and hypothesize causes for a basic hardware problem and a basic software problem.**
4	Describe the appropriate user response to correct a basic hardware or software error when given an error message.
5	Propose and test a plan to address basic computing problems using hardware or software when given various error messages.

Thinking Across Disciplines in Grade 3

Core Concept 1: Computing Systems



The standards in **Core Concept 1: Computing Systems** build student understanding of the hardware and software that make up computers and guide the intentional and effective use of these tools.

- **Problem-solving** is a skill emphasized across many content areas. Students need to be able to understand and address problems of various kinds to strengthen analytical abilities. Troubleshooting issues that emerge when using computers provides a new context for students to practice problem-solving skills.
- Evaluating hardware and software applications and selecting the best tools to meet specific user needs requires **critical thinking**. Strong critical thinking skills allow students to draw informed conclusions, make reliable recommendations, and justify their decisions.
- Understanding the purpose and function of hardware and software supports effective technology integration from early stages of education. Developing **competence and familiarity with technology** equips students for success in both the workforce and higher education. A solid foundation in hardware, software, and troubleshooting deepens students' knowledge of technology tools and their applications.

T.A.D. Examples

Standard	**E.CS.1A. Identify and select the appropriate hardware to complete computing tasks.** **Complexity Statement:** Identify the appropriate hardware to complete a given task. **Core Practices:** Collaborating Around Computing; Recognizing and Defining Computation Problems; Communicating about Computing
Connections	**Mathematics** 3.MD.A.1 Understand time to the nearest minute. Tell and write time to the nearest minute and measure time intervals in minutes, within 60 minutes, on an analog and digital clock.
Integration	Identifying the appropriate tool to solve a problem is a fundamental skill in both mathematics and computer science. In mathematics, as students learn to solve time-related problems, they must determine the best tool to use. Options include an analog clock, a digital clock, a number line, or another time model. For drawing these models, students can use a stylus, mouse, or finger on a tablet or iPad.

Standard	**E.CS.1B. Identify and select the appropriate software to complete computing tasks.** **Complexity Statement:** Identify the appropriate software to complete a given task. **Core Practices:** Collaborating Around Computing; Recognizing and Defining Computation Problems; Communicating about Computing
Connections	**Mathematics** 3.MD.B.3 Draw a scaled picture graph and a scaled bar graph to represent a data set with several categories. Solve one and two-step "how many more" and "how many less" problems using information presented in scaled bar graphs. *For example, draw a bar graph in which each square in the bar graph might represent 5 pets..* **Science** 3-ESS2-1 Represent data in tables and graphical displays to describe typical weather conditions expected during a particular season.

Integration	Just as tools in mathematics and science have capabilities and limitations, so does computing software. In math, students can collect data from a word problem to organize into a graph. In science, students can collect data on daily weather and layout into a table to show the changes throughout the week. Students in both math and science can then choose the best software to use to compile graphs from throughout the week. For example, students may choose to insert pictures of each graph into a word document, create a slideshow, or create digital graphs using software.

Standard	**E.CS.1C. Evaluate hardware and software types to meet users' needs in completing various computing tasks.** **Complexity Statement:** Describe the limitations of certain computing devices in given scenarios. **Core Practices:** Collaborating Around Computing; Recognizing and Defining Computation Problems; Communicating about Computing
Connections	**Mathematics** 3.MD.A.2 Measure and estimate liquid volumes and masses of objects using standard units of grams (g), kilograms (kg), and liters (l).[5] Add, subtract, multiply, or divide to solve one-step word problems involving masses or volumes that are given in the same units, e.g., by using drawings (such as a beaker with a measurement scale) to represent the problem.[6] **Science** 3-PS2-1 Plan and conduct an investigation to provide evidence of the effects of balanced and unbalanced forces on the motion of an object.
Integration	Considering the available tools and deciding which best fits the task is important when solving a mathematical problem, collecting data, or conducting research. In science, a student researching information on the ways to change motion of an object may choose to communicate their findings using presentation software instead of a project board due to the ease of access and availability. In math, students would have to choose the best tool to get measurements. Some limitations that should be considered include units, accuracy of measurement, availability, and ease of use.

Standard	**E.CS.2A. Propose potential ways to address computing problems using appropriate hardware or software.** **Complexity Statement:** Describe and hypothesize causes for a basic hardware problem and a basic software problem. **Core Practices:** Collaborating Around Computing; Recognizing and Defining Computational Problems; Communicating about Computing
Connections	**Mathematics** 3.MP.1 Make sense of problems and persevere in solving them. 3.MP.3 Construct viable arguments and critique the reasoning of others.
Integration	Supporting students as problem solvers who consider all parts of a problem to determine the most appropriate solution is already a common practice across all curricula. In computer systems, both software and hardware present problems that can be addressed in various ways. Similar to how math problems often require multiple steps and strategies, hardware and software issues can have multiple root causes. As students encounter problems with hardware and/or software in the classroom, give support by using prompts to help the students to problem solve. For instance, if a student's mouse isn't functioning, they could collaborate to brainstorm potential causes and then discuss solutions.

Core Concept 2: Networks and the Internet (NI)

Overview

Computing systems typically do not operate in isolation. Networks connect computers to share information and resources and are integral to computer and data science. Networks enable critical communication for the computing systems that drive our economy and career sectors. The increased level of connectivity brought about by the internet provides fast and secure communication that facilitates innovation.

Elementary Standards for Core Concept 2: Networks and the Internet

Subconcepts and Core Practices	Elementary (K-5)
Subconcept 1: Hardware and Network Communication *Core Practices: Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing*	E.NI.1A. Explain how networks connect computers to other computing systems and the internet.
Subconcept 2: Cybersecurity *Core Practices: Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing*	E.NI.2A. Describe personally identifiable information (PII) and identify practices for when and where sharing PII is appropriate.
	E.NI.2B. Identify ways to maintain data security when using networks.

Complexity Statements

Core Concept 2: Networks and the Internet

Complexity statements for the standards included in **Core Concept 2: Networks and the Internet** break down each standard into manageable, age-appropriate components. This breakdown ensures that instruction fits the developmental needs of students at each grade level. For instance, kindergarten focuses on defining basic network concepts and distinguishing between public and private information. By grade 5, students diagram network interactions and analyze the complexities of data sharing across networks, including the importance of responsible online behavior and security practices.

E.NI.1A. Explain how networks connect computers to other computing systems and the internet.	
Grade	**Complexity Statement**
K	Define what a network is and how it is used to connect to other people and places around the world.
1	Compare and contrast wireless versus plugged-in networks.
2	Model and describe how computers break down data to share on networks.
3	**Explain how data is transmitted in packets.**
4	Create a model depicting how information is shared within a network.
5	Diagram and summarize how data can be shared between multiple networks and users via the Internet.

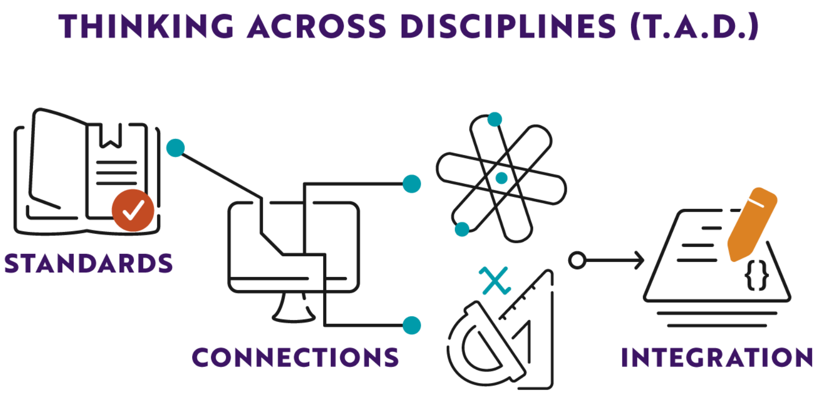
E.NI.2A. Describe personally identifiable information (PII) and identify practices for when and where sharing PII is appropriate.	
Grade	**Complexity Statement**
K	Distinguish between public and private information.
1	Define personally identifiable information (PII).
2	Evaluate the risks and benefits of sharing PII.
3	**Evaluate mechanisms of password generation and best practices for password management.**
4	Describe the role of authentication and authorization.

E.NI.2A. Describe personally identifiable information (PII) and identify practices for when and where sharing PII is appropriate.	
Grade	**Complexity Statement**
5	Identify and determine the appropriate resources and individuals to notify if PII is inappropriately shared.

E.NI.2B. Identify ways to maintain data security when using networks.	
Grade	**Complexity Statement**
K	Describe how usernames and passwords protect information and network access.
1	Distinguish between the six top-level domains and describe what information they share.
2	Identify and classify a network as secure or insecure in a given scenario.
3	**Identify and assess when an email is a phishing attempt within the examples.**
4	Explain what antivirus software is and how it protects a user, network, and data sharing.
5	Explain the ways a student can protect their home and school networks.

Thinking Across Disciplines in Grade 3

Core Concept 2: Networks and the Internet



The standards in **Core Concept 2: Networks and the Internet** develop an understanding of how networks connect computers to the internet and other computing systems, along with safe practices for sharing personally identifiable information. These skills are tied to concepts teachers already address in their classrooms.

- **Obtaining, evaluating, and communicating information** is an interdisciplinary skill. Understanding how information is transmitted through a network safely strengthens the ability to obtain, evaluate, and communicate information electronically.
- Explaining safe practices for sharing personally identifiable information on a network requires **critical thinking** to evaluate what information is appropriate to share, when to share it, and how to do so responsibly. This decision-making process supports the development of responsible technology use.
- Maintaining data security when using networks involves recognizing common risks and applying strategies to protect sensitive information. Practicing data protection builds **digital responsibility** and supports safer online interactions.

T.A.D. Examples

Standard	**E.NI.1A. Explain how networks connect computers to other computing systems and the internet.** **Complexity Statement:** Explain how data is transmitted in packets. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Math** 3.NF.A.1 Understand a fraction 1/b, with denominators 2,3,4,6, and 8, as the quantity formed by 1 part when a whole is partitioned into b equal parts; understand a fraction a/b as the quantity formed by parts of a size 1/b. **Science** 3-PS2-4 Define a simple design problem that can be solved by applying scientific ideas about magnets.
Integration	In mathematics, students learn that fractions are created when a whole is partitioned into smaller, equal parts, much like data being sent over the internet is split into packets. Students can represent the fractions or packets using a number line diagram. In science, in order for a design to function, all pieces and parts have to be connected adequately. With a failed connection, the computing system may not connect to the network. For example, magnets attract with opposite poles, or repel with like poles. In a network, computers are "connected" when they have the same network protocol or are compatible, while incompatible protocols or network misconfiguration could be compared to magnets repelling. Students can explain why magnets sometimes attract and sometimes repel, stating the importance of attracting and opposing poles.

Standard	**E.NI.2A. Describe personally identifiable information (PII) and identify practices for when and where sharing PII is appropriate.** **Complexity Statement:** Evaluate mechanisms of password generation and best practices for password management. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing

Connections	**Mathematics** 3.OA.A.3 Use multiplication and division within 100 to solve word problems in situations involving equal groups, arrays, and measurement quantities, e.g., by using drawings and equations with a symbol for the unknown number to represent the problem.
Integration	Students can generate arrays in different ways based on variables that fit the total. When creating passwords, there are often requirements that need to be met in order to have an adequate password. While there are explicit directions necessary for creating passwords, there could be a connection to the trial-and-error creation of arrays.

Standard	**E.NI.2B. Identify ways to maintain data security when using networks.** **Complexity Statement:** Identify and assess when an email is a phishing attempt within the examples. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** 3-LS4-3 Construct and support an argument with evidence that in a particular habitat some organisms can survive well, some survive less well, and some cannot survive at all.
Integration	Just as animals must recognize which habitats are safe for them, people must recognize which digital spaces and messages are safe online. In science, students can brainstorm which animals can live in certain ecosystems and why the environment is safe or unsafe. Teachers can point out that there are certain clues in emails that can help identify if it is safe or unsafe. Students can look at emails (one safe and one unsafe) about animal ecosystems and look for suspicious content such as misspelled words, links to free prizes, or asks for personal information.

Core Concept 3: Data and Analysis (DA)

Overview

Computing systems function by processing and storing data. Data is abundant due to the growing number of connected devices worldwide. As the volume of data expands, so does the demand for accurate and efficient data analysis methods. Data science is the cross-disciplinary use of data to inform decision-making, test hypotheses, predict trends, and develop precise models that drive innovation across industries.

Elementary Standards for Core Concept 3: Data and Analysis

Subconcepts and Core Practices	Elementary (K-5)
Subconcept 1: Data Representation *Core Practices: Collaborating around computing; Recognizing and defining computational problems; Creating computational artifacts*	E.DA.1A. Organize and present data visually to highlight relationships and support claims.
	E.DA.1B. Classify types of data and describe the attributes used to sort data.
Subconcept 2: Data Collection *Core Practices: Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing*	E.DA.2A. Select the appropriate data collection tool and technique to gather data to support a claim or communicate information.
	E.DA.2B. Describe and collect data utilizing the appropriate units of measure and discuss how data format impacts a computing system.
Subconcept 3: Data Storage *Core Practices: Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing*	E.DA.3A. Compare and contrast ways to store data using technology.
	E.DA.3B. Explain how to save and name data, search for data, retrieve data, modify data, and delete data using a computing device.
Subconcept 4: Visualizations and Transformations *Core Practices: Creating computational artifacts; Communicating about computing*	E.DA.4A. Organize and present data visually in at least three ways to highlight relationships and evaluate a claim.

	E.DA.4B. Evaluate data quality and clean data when indicated using the criteria of validity, accuracy, completeness, consistency, and uniformity.
Subconcept 5: Inference and Models *Core Practices: Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing*	E.DA.5A. Utilize data to create models, answer investigative questions, and make predictions.
	E.DA.5B. Analyze data for patterns and relationships.

Complexity Statements

Core Concept 3: Data and Analysis

Complexity statements for the standards included in **Core Concept 3: Data and Analysis** break down each standard into manageable, age-appropriate components. This breakdown ensures that instruction fits the developmental needs of students at each grade level. For instance, in kindergarten, the focus is on classifying and counting data into simple categories, while by grade 5, students analyze multiple types of data to support claims, compare different datasets for accuracy, and generate appropriate computational visualizations to explain their findings.

E.DA.1A. Organize and present data visually to highlight relationships and support claims.

Grade	Complexity Statement
K	Sort objects into given categories based on attributes. Determine the total count for each category and enter counts into a data table on the computer with teacher support.
1	Organize, represent, and interpret data with up to three categories based on common attributes using computational tools.
2	Create bar graphs (with single-unit scale) to represent a data set with up to four categories using computational tools.
3	**Create scaled bar graphs representing a data set with more than four categories using computational tools.**
4	Compare computationally generated graphs (e.g., bar and pie) to their corresponding data tables to validate their accuracy.
5	Generate line graphs from data that is collected and entered into data tables to support claims using computational tools.

E.DA.1B. Classify types of data and describe the attributes used to sort data.

Grade	Complexity Statement
K	Explain what the term data means and give examples.
1	Select the appropriate type of data to support or disprove a claim.
2	Define and describe digital, non-digital, numerical, text-based, audio, visual, and video data types.

E.DA.1B. Classify types of data and describe the attributes used to sort data.	
Grade	**Complexity Statement**
3	**Evaluate the tradeoffs of using one data type over another to support or refute a given claim.**
4	Analyze two or more types of data to support or disprove a claim.
5	Identify specific types of data, such as metadata, and explain what the information means, including its application.

E.DA.2A. Select the appropriate data collection tool and technique to gather data to support a claim or communicate information.	
Grade	**Complexity Statement**
K	Collect and organize data computationally in a table or chart with support.
1	Identify and use the appropriate digital tool (e.g., thermometer, scale, probe) to collect various data computationally in tables or charts with support.
2	Collect and compile data computationally to evaluate a claim's validity.
3	**Create and implement practices for collecting valid and accurate data using a data-gathering tool.**
4	Compare and contrast various data sets collected for accuracy and describe potential sources of instrumentation error.
5	Compare and contrast data sets collected from two or more instruments and use the data to support a claim.

E.DA.2B. Describe and collect data utilizing the appropriate units of measure and discuss how data format impacts a computing system.	
Grade	**Complexity Statement**
K	Define and describe various measurable attributes of objects and practice comparing two or more objects with a common attribute.

E.DA.2B. Describe and collect data utilizing the appropriate units of measure and discuss how data format impacts a computing system.	
Grade	**Complexity Statement**
1	Generate measurement data by measuring the lengths of various objects, then record the data computationally with units of support.
2	Sort data within tables based on their units of measure from various sources and tools.
3	**Describe how user errors in data entry can change or alter the computing system outputs with examples of impacts on real-world business problems.**
4	Practice entering data in fraction and decimal format using computational tools. Compare and contrast fraction and decimal notations in computing for accuracy.
5	Convert among different-sized standard measurement units within a given measurement system, and explain why the data entry format can impact how a computing program can access the use of data.

E.DA.3A. Compare and contrast ways to store data using technology.	
Grade	**Complexity Statement**
K	Apply the process to save a data file in a computing system.
1	Apply the process of saving and naming a computational data file on an external storage device.
2	Describe the cybersecurity risks of using external data storage devices with a computing system.
3	**Explain what cloud data storage is and compare it to external storage devices.**
4	Explain and apply the appropriate data units (e.g., bits, bytes, kilobytes, megabytes, gigabytes, terabytes) to numerical quantities of data.
5	Compare and contrast the storage size capacity for computational storage devices and select the appropriate format for given tasks.

E.DA.3B. Explain how to save and name data, search for data, retrieve data, modify data, and delete data using a computing device.	
Grade	**Complexity Statement**
K	Locate and open a data file on a computing system.
1	Locate the proper computing application and create a new file.
2	Explain how to delete a file and return or recover the file if accidentally deleted.
3	**Design and apply a file naming structure to organize computational data for sharing with others so they can use and locate the files.**
4	Modify the names of files and describe why file naming is connected to function in school or workplace settings.
5	Compare and contrast the uses and data stored on the file types: JPEG, GIF, PDG, DOC, or DOCX, XLS, or DLSX, PPT or PPTX, MP3, and WAV.

E.DA.4A. Organize and present data visually in at least three ways to highlight relationships and evaluate a claim.	
Grade	**Complexity Statement**
K	Describe patterns of similarity to organize data into countable amounts and record them in a computational data tool.
1	Generate numerical predictions (based on prior experiences) and compare them with data collected from a tool (observed data) in a computational table and on bar graphs.
2	Collect data and record it in a computational table. Use the table to generate a bar graph and a pie graph. Utilize the 3 data visualizations to evaluate a claim.
3	**Create a computational data table using data visualizations (e.g., graphs, or pie charts) from multiple data collection events on the same phenomenon. Analyze newly created data tables for patterns and hypothesize what they mean for the phenomenon.**
4	Collect and use data to generate the appropriate computational visualizations (e.g., data table, bar, pie, line graph) to explain a hypothesis.
5	Collect at least five data sets and use computational tools to create visualizations. Compare and contrast visualizations, findings, and conclusions with those of peers.

E.DA.4B. Evaluate data quality and clean data when indicated using the criteria of validity, accuracy, completeness, consistency, and uniformity.	
Grade	**Complexity Statement**
K	Recognize data used to determine valid groupings.
1	Recognize the need for consistency and completeness when evaluating data.
2	Propose and design a protocol for collecting data, ensuring that it is valid, consistently measured, and complete.
3	**Measure and estimate liquid volumes and masses of objects using standard units of grams, kilograms, and liters, and record the data computationally.**
4	Sort and analyze a given data set containing various units and decimals for accuracy using computational tools. Explain and justify data conversions or strategies.
5	Evaluate a data set, within the constraints of a scenario, for the criteria of validity, accuracy, completeness, consistency, and uniformity. Explain any issues with the data set and propose corrections.

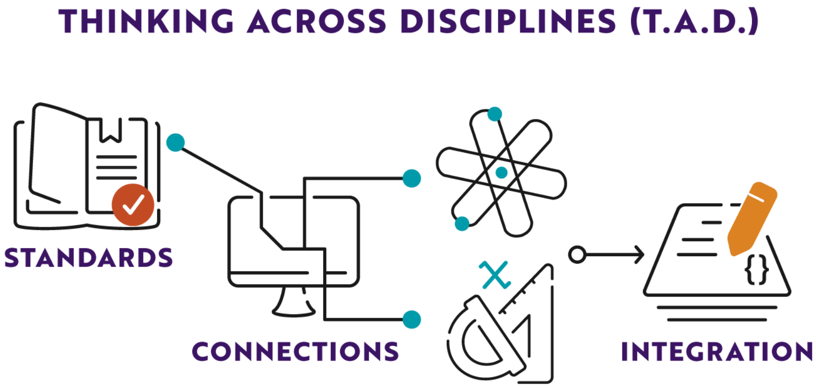
E.DA.5A. Utilize data to create models, answer investigative questions, and make predictions.	
Grade	**Complexity Statement**
K	Use computational tools to create and describe shapes.
1	Use computational tools to create partitions of circles and rectangles into two and four equal shares. Understand that decomposing into more equal shares creates smaller ones.
2	Plan and create a computational artifact to illustrate thoughts, ideas, and problems in a sequential (step-by-step) manner.
3	**Use a computational data artifact to make predictions and judge if there is enough data to support a claim.**
4	Recreate sample shapes computationally, and classify two-dimensional shapes based on the presence or absence of parallel or perpendicular lines or the presence or absence of angles of a specified size. Draw a line and test if symmetry is possible for each figure.

E.DA.5A. Utilize data to create models, answer investigative questions, and make predictions.	
Grade	**Complexity Statement**
5	Represent real-world and mathematical problems by creating a data table and graphing points in the first quadrant of the coordinate plane using computational tools. Interpret coordinate values of points in the context of the situation or examining a statement for validity.

E.DA.5B. Analyze data for patterns and relationships.	
Grade	**Complexity Statement**
K	Classify and count objects into given observable categories, then use a computational data table to record data.
1	Organize, represent, and interpret data with up to three categories, then ask and answer questions about the data.
2	Create a square or rectangular geometric shape using a repeat function using block-based code. Explain the relationship between adding more or less than four repeats and the picture's outcome.
3	**Create computational addition or multiplication tables and identify arithmetic patterns. Apply the patterns identified to predict the next three outcomes.**
4	Generate a number pattern that follows a given code rule using block-based coding.
5	Generate numerical patterns based on rules using block-based code. Identify relationships between terms and use computational tools to create a table and graph of the resulting ordered pairs.

Thinking Across Disciplines in Grade 3

Core Concept 3: Data and Analysis



The standards in **Core Concept 3: Data and Analysis** focus on collecting and representing digital data, understanding how data is stored, and creating and analyzing computational data for pattern recognition. These skills are closely tied to concepts teachers already address in their classrooms.

- Organizing and representing data visually in different formats strengthens the ability to **highlight relationships and support conclusions**. This skill enhances communication and interpretation of complex information.
- Drawing conclusions based on evidence depends on understanding various data types and using data to make and support claims. Strengthening this skill supports **evidence-based reasoning**.
- Classifying data and choosing appropriate collection methods support accurate analysis and deepen understanding of content. These practices promote **thoughtful data use** in scientific investigations, social research, math applications, and more.
- Analyzing digital data for patterns and relationships enhances **informational literacy** by reinforcing the ability to locate, use, and evaluate data effectively. This skill develops **critical thinking** and reasoning abilities necessary for making predictions, constructing explanations, and creating models, supporting inquiry and problem-solving processes.

T.A.D. Examples

Standard	**E.DA.1A. Organize and present data visually to highlight relationships and support claims.** **Complexity Statement:** Create scaled bar graphs representing a data set with more than four categories using computational tools. **Core Practices:** Collaborating around computing; Recognizing and defining computational problems; Creating computational artifacts
Connections	**Mathematics** 3.MD.B.3 Draw a scaled picture graph and a scaled bar graph to represent a data set with several categories. Solve one and two-step "how many more" and "how many less" problems using information presented in scaled bar graphs. *For example, draw a bar graph in which each square in the bar graph might represent 5 pets.* **Science** 3-ESS2-1 Represent data in tables and graphical displays to describe typical weather conditions expected during a particular season.
Integration	Comparing data tables to graphs for validity is an important scientific skill. For example, a student analyzing data to describe typical weather conditions expected during a particular season can use computational data and graphs for comparison. In math, data can be used to organize information to show patterns or trends, or to support a claim in a word problem.

Standard	**E.DA.1B. Classify types of data and describe the attributes used to sort data.** **Complexity Statement:** Evaluate the tradeoffs of using one data type over another to support or refute a given claim. **Core Practices:** Collaborating around computing; Recognizing and defining computational problems; Creating computational artifacts
Connections	**Science** 3-LS3-1 Analyze and interpret data to provide evidence that plants and animals have traits inherited from their parents and that variation of these traits exists in a group of similar organisms.
Integration	Using multiple types of data to support or disprove a claim adds to the validity of the claim. For example, students may choose to use numerical data on the inherited traits of animals as well as text-based data on the specificity of the traits. Students can then explain why one type of data over another was used.

Standard	**E.DA.2A. Select the appropriate data collection tool and technique to gather data to support a claim or communicate information.** **Complexity Statement:** Create and implement practices for collecting valid and accurate data using a data-gathering tool. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.MP.5 Use appropriate tools strategically. **Science** Science & Engineering Practices 4: Analyzing and interpreting data
Integration	In math, students use certain tools to collect various types of data. Having students create criteria for collecting accurate data could be a way to help students to understand the importance of accuracy in data collection. In science, students can put the criteria for data collection to use and evaluate and analyze its effectiveness. This could be a great opportunity for students to edit their criteria based on the results of their data collection.

Standard	**E.DA.2B. Describe and collect data utilizing the appropriate units of measure and discuss how data format impacts a computing system.** **Complexity Statement:** Describe how user errors in data entry can change or alter the computing system outputs with examples of impacts on real-world business problems. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.MD.A.2 Measure and estimate liquid volumes and masses of objects using standard units of grams (g), kilograms (kg), and liters (l).[5] Add, subtract, multiply, or divide to solve one-step word problems involving masses or volumes that are given in the same units, e.g., by using drawings (such as a beaker with a measurement scale) to represent the problem.[6]

Integration	Accurate measurements in mathematics is just as important as accurate data entry. Even in word problems, if students copy down the wrong information, the error can alter the response. In Math, students can collect measurement data, but need to ensure they use the correct tool and units to accurately enter the data. If students need to combine the data, computation errors can alter the results. For example, students measuring the liquid volume of water can be asked, "If the beaker has 300mL and we pour 200mL more, what would the total volume be?" Students can be shown that entering 3,000 instead of 300 results in an error. A connection can then be made of making an error when baking a cake and not adding up the correct volume of liquid ingredients.

Standard	**E.DA.3A. Compare and contrast ways to store data using technology.** **Complexity Statement:** Explain what cloud data storage is and compare it to external storage devices. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.MP.5 Use appropriate tools strategically
Integration	Students can store data on their computers, in the cloud, or on an external drive. The way the data will be used in the future will determine the best way to store the data. As students are creating computational artifacts, have a discussion about which storage location is best and why. For example, if students need to access the artifact at home, the cloud would be a better choice over their computer at school. If the artifact needs to be shared with others, an external storage device may not work best.

Standard	**E.DA.3B. Explain how to save and name data, search for data, retrieve data, modify data, and delete data using a computing device.** **Complexity Statement:** Design and apply a file naming structure to organize computational data for sharing with others so they can use and locate the files. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing

Connections	**Mathematics** 3.OA.B.6 Understand division as an unknown-factor problem. *For example, find 32 8 by finding the number that makes 32 when multiplied by 8.* **Science** 3-LS1-1 Develop models to describe that organisms have unique and diverse life cycles, but all have in common birth, growth, reproduction and death.
Integration	When students are looking for a document on a computer, knowing how to save the file under a name that is relevant to the function will help to find documents later. If the file is not named correctly, students can change the file name to reflect the assignment. In Math, it is important for students to correctly identify the number of groups as well as the number of items in a group when learning multiplication and division. The importance of the label or name of data is important for later retrieving, modifying, or deleting data. In science, students can discuss the similarities in the life cycles of animals and create a model to show the cycle. Students could extend this activity by creating a file name to save their model into a folder the teacher creates that would specifically identify their own work from their peers.

Standard	**E.DA.4A. Organize and present data visually in at least three ways to highlight relationships and evaluate a claim.** **Complexity Statement:** Create a computational data table using data visualizations (e.g., graphs, or pie charts) from multiple data collection events on the same phenomenon. Analyze newly created data tables for patterns and hypothesize what they mean for the phenomenon. **Core Practices:** Creating computational artifacts; Communicating about computing
Connections	**Mathematics** 3.MD.B Represent and interpret data. **Science** 3-LS3-1 Analyze and interpret data to provide evidence that plants and animals have traits inherited from their parents and that variation of these traits exists in a group of similar organisms.

Integration	Collecting and using data allows for interpretation and communication. Students can choose the best visualization using various computational platforms such as Word, Excel, or an online graphing website. In math, students construct picture or bar graphs and line plots. Showing students how to create these electronically could enhance their presentation of data. In science, students can create a model of inherited traits and predict the attributes of their offspring based on the patterns recognized in the traits.

Standard	**E.DA.4B. Evaluate data quality and clean data when indicated using the criteria of validity, accuracy, completeness, consistency, and uniformity.** **Complexity Statement:** Measure and estimate liquid volumes and masses of objects using standard units of grams, kilograms, and liters, and record the data computationally. **Core Practices:** Creating computational artifacts; Communicating about computing
Connections	**Mathematics** 3.MD.A.2 Measure and estimate liquid volumes and masses of objects using standard units of grams (g), kilograms (kg), and liters (l).[5] Add, subtract, multiply, or divide to solve one-step word problems involving masses or volumes that are given in the same units, e.g., by using drawings (such as a beaker with a measurement scale) to represent the problem.[6]
Integration	Understanding the purpose of data collection and its significance to the problem will help students to collect the most valid and accurate data. For example, students should decide how to collect data, whether it be precise or an estimation, to solve a word problem before computing. In mathematics, students can select appropriate measurements and estimate volumes and masses of objects. They can enter their estimates computationally and then measure to compare their estimates.

Standard	**E.DA.5A. Utilize data to create models, answer investigative questions, and make predictions.** **Complexity Statement:** Use a computational data artifact to make predictions and judge if there is enough data to support a claim**.** **Core Practices:** Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing

Connections	**Mathematics** 3.OA.A.3 Use multiplication and division within 100 to solve word problems in situations involving equal groups, arrays, and measurement quantities, e.g., by using drawings and equations with a symbol for the unknown number to represent the problem. [1] **Science** ESS2-1 Represent data in tables and graphical displays to describe typical weather conditions expected during a particular season.
Integration	Computer science, mathematics, and science standards focus on using data to solve real-world problems, either by analyzing data to make predictions or applying multiplication and division to understand patterns and quantities. In math, students can create an array and predict the number of coding blocks needed to convert it to a two-dimensional figure. If there was not enough data, the student would not be able to accurately create the shape or object. In science, students can predict the weather based on weather patterns, and then judge if their predictions were adequate based on the information they collected prior to data collection. They may need to redo the assignment by collecting more accurate data prior to predicting, then collecting the data again to re-evaluate by answering investigative questions.

Standard	**E.DA.5B. Analyze data for patterns and relationships.** **Complexity Statement:** Create computational addition or multiplication tables and identify arithmetic patterns. Apply the patterns identified to predict the next three outcomes. **Core Practices:** Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.OA.D.9 Identify arithmetic patterns (including patterns in the addition table or multiplication table), and explain them using properties of operations. *For example, observe that 4 times a number is always even, and explain why 4 times a number can be decomposed into two equal addends.*
Integration	In mathematics, students can use a program such as Google Sheets or Excel to create a multiplication table for numbers 1-6. Students can then use the patterns formed in the computational table to make predictions of larger numbers.

Core Concept 4: Algorithms and Programming (AP)

Overview

An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing systems. Algorithms and programs control all computing systems, empowering people to communicate with the world in novel ways and solve compelling problems. The development process to create meaningful and efficient programs involves choosing which information to use, how to process the data, and how to store the information. The decomposition of more significant problems into simpler ones, combined with recombining existing solutions and analyzing various solutions, helps determine the most appropriate solution to a problem.

Elementary Standards for Core Concept 4: Algorithms and Programming

Subconcepts and Core Practices	Elementary (K-5)
Subconcept 1: Variables and Algorithms *Core Practices: Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing*	E.AP.1A. Create clearly named variables representing different data types and perform operations on the variables' values.
	E.AP.1B. Create, use, and apply an algorithm to complete a task. Compare the results of algorithm usage trials and refine the algorithm.
Subconcept 2: Control Structures *Core Practices: Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing*	E.AP.2A. Define what a control structure is and create programs that include sequences, conditionals, events, and loops.
Subconcept 3: Modularity *Core Practices: Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing*	E.AP.3A. Define and apply decomposition to a complex problem in order to create smaller subproblems that can be solved through step-by-step instructions.
	E.AP.3B. Modify, remix, or incorporate parts of an existing problem's solution to develop something new or add more advanced features to a program.

Subconcept 4: Program Development *Core Practices: Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing*	E.AP.4A. Create a simple program to achieve a goal with expected outcomes.
	E.AP.4B. Test and debug a program or algorithm to ensure the program produces the intended outcome.
	E.AP.4C. Collaborate with a team of peers to design, implement, test, and review the stages of program development.
	E.AP.4D. Identify intellectual property rights and apply the appropriate attribution when creating or remixing programs.
	E.AP.4E. Describe and justify the steps taken and choices made during a program's development.
	E.AP.4F. Using an iterative process, test a program step-by-step and document areas of refinement.

Complexity Statements

Core Concept 4: Algorithms and Programming

Complexity statements for the standards included in **Core Concept 4: Algorithms and Programming** break down each standard into manageable, age-appropriate components. This breakdown ensures that instruction fits the developmental needs of students at each grade level. For instance, in kindergarten, the focus is on creating simple step-by-step plans and understanding basic sequencing, while by grade 5, students create efficient algorithms by structuring repeated actions, decomposing complex tasks into manageable steps, and designing multi-step programs using block-based coding to create animations or solve problems with various constraints.

E.AP.1A. Create clearly named variables representing different data types and perform operations on the variables' values.	
Grade	**Complexity Statement**
K	Count and group items under broad categorical (variable) names.
1	Understand subtraction as an unknown-addend problem.
2	Use addition and subtraction to solve word problems involving situations of adding to, taking from, putting together, taking apart and comparing, with unknowns in all positions.
3	**Solve real-world problems involving perimeters of polygons, including finding the perimeter given the side lengths and finding an unknown side length.**
4	Use the area and perimeter formulas to write math sentences for rectangles in real-world and mathematical problems with variables used to represent unknown factors.
5	Represent unknown quantities in equations with a letter. Define what a variable is and why a letter can be utilized as a placeholder or representative of the numerical value.

E.AP.1B. Create, use, and apply an algorithm to complete a task. Compare the results of algorithm usage trials and refine the algorithm.	
Grade	**Complexity Statement**
K	Create an algorithm with set step-by-step stages to complete a task.
1	Apply the properties of operations to add and subtract.
2	Identify the various ways groups are formed, draw arrays, and create math sentences.

E.AP.1B. Create, use, and apply an algorithm to complete a task. Compare the results of algorithm usage trials and refine the algorithm.	
Grade	**Complexity Statement**
3	**Solve problems involving the four operations, and identify and explain patterns in arithmetic.**
4	Develop and refine algorithms to solve multi-step word problems involving only whole numbers. Represent the unknown quantity with a letter.
5	Create efficient algorithms by structuring repeated actions using parentheses or brackets and optimizing lengthy instructions into simpler equations using multiplication, addition, and subtraction.

E.AP.2A. Define what a control structure is and create programs that include sequences, conditionals, events, and loops.	
Grade	**Complexity Statement**
K	Code a character or robot using block-based programming to follow a teacher-specified path with simple instructions. Explain the function of each command.
1	Code a character or robot using block-based programming to follow a teacher-specified sequence of multi-step instructions that include stops, starts, and turns. Define control structure and give examples.
2	Code a character or robot using block-based programming to make the figure move using loops for iteration. Define and explain the control structures.
3	**Code a character or robot using block-based programming to make the figure move based on conditionals. Define and explain conditionals.**
4	Code a character or robot using block-based programming to make the figure move using different control structures to achieve a teacher-specified, complex task. Compare and contrast the control structures.
5	Use block-based coding to create a solution to a complex task. Test and compare the solution and use of control structures with peers for efficiency.

E.AP.3A. Define and apply decomposition to a complex problem in order to create smaller subproblems that can be solved through step-by-step instructions.	
Grade	**Complexity Statement**
K	Divide a task into smaller parts with teacher assistance. Identify the steps to solve the smaller parts.
1	Define decomposition and apply it to fact, families (e.g., given three numbers, students explore the four ways to write them to achieve correct addition and subtraction questions) Explain thinking.
2	Apply the decomposing and composing subtraction strategy to work with three-digit numbers.
3	**Diagram and determine the appropriate steps to achieve a mathematical solution to a multi-step word problem.**
4	Compare color bar models, addends, unit fraction addition formulas, and number bond charts to decompose fractions and explain the reasoning for each model.
5	Decompose and create steps for solving complex problems that address various constraints such as cost, efficiency, length of time, number of people needed, etc.

E.AP.3B. Modify, remix, or incorporate parts of an existing problem's solution to develop something new or add more advanced features to a program.	
Grade	**Complexity Statement**
K	Practice analytical and sequential reasoning by reconstructing a process from out-of-order images. After sequencing, justify the arrangement to peers. Reconstruct the correct sequence of the images illustrating a process, then justify the arrangement to peers.
1	Code a character or robot using block-based programming to make the figure move using stops, starts, and turns. Reorganize the steps to make the figure return to its starting point.
2	Diagram and code multi-step word problems using block-based programming to move a figure or robot.
3	**Collaborate to design and implement a step-by-step plan for completing a task. Exchange plans, test, and provide feedback to peers to refine the plan.**
4	Design a step-by-step plan (algorithm) to draw a polygon. Collaborate with peers to combine and improve these plans, then modify the best plan to create different polygons.

E.AP.3B. Modify, remix, or incorporate parts of an existing problem's solution to develop something new or add more advanced features to a program.	
Grade	**Complexity Statement**
5	Modify pre-existing code by decomposing and identifying reusable parts to develop something new or add more advanced features.

E.AP.4A. Create a simple program to achieve a goal with expected outcomes.	
Grade	**Complexity Statement**
K	Create a step-by-step plan using pictures.
1	Sort and organize the appropriate written steps to solve the problem when given a scenario.
2	Create written steps to solve the problem in a given scenario with minimal prompting from the teacher.
3	**Collaborate to propose solutions for each stage of a multi-stage problem or scenario.**
4	Collaborate to code solutions for each stage of a multi-stage problem or scenario.
5	Create and propose a program to solve a problem with expected outcomes.

E.AP.4B. Test and debug a program or algorithm to ensure the program produces the intended outcome.	
Grade	**Complexity Statement**
K	Identify the errors and propose fixes to a process modeled by pictures with guidance from the teacher.
1	Test multistep, block-based code intended to make a figure or robot move in a specified sequence, but contains flaws. Debug the code and work towards achieving the intended outcome.
2	Test multi-step, block-based code intended to make a figure or robot move, but contains flawed loops. Work with a partner to debug, test, and explain how to correct the loops.

E.AP.4B. Test and debug a program or algorithm to ensure the program produces the intended outcome.	
Grade	**Complexity Statement**
3	**Test a multi-step, block-based code intended to make a figure or robot move but contains flawed conditionals. Work with a partner to debug, test, and explain how to correct the conditionals.**
4	Test block-based code that includes multi-step instructions and flawed control structures to make a figure move. With a partner, debug the code to explain how to properly correct this control structure, then test the program.
5	Create and test a solution to a complex task using block-based coding. Then modify the code to include up to 4 (documented) errors. Then trade programs with a peer. Test and debug the code. Identify and explain any challenges.

E.AP.4C. Collaborate with a team of peers to design, implement, test, and review the stages of program development.	
Grade	**Complexity Statement**
K	Define what a programmer is and apply what they do by modeling the creation of a simple step-by-step program to accomplish a task.
1	Collaborate to complete an outcome-driven coding task following the steps of planning, designing, and testing to a simple code sequence using block-based coding or written steps.
2	Decompose a problem and propose a solution by creating a flowchart. Use the flowchart as a model to code the solution.
3	**Collaborate with peers to test and debug. Once complete, write a story that provides an overview of the code's function and how to resolve errors.**
4	Summarize the functions of an existing program's code.
5	Document the stages of the program development process while creating a coded solution to a scenario.

E.AP.4D. Identify intellectual property rights and apply the appropriate attribution when creating or remixing programs.	
Grade	**Complexity Statement**
K	Model and discuss when students should ask permission to use an item belonging to someone else to establish the meaning of ownership and appropriate sharing.
1	Define digital ownership with teacher guidance. Identify common intellectual property symbols and text for copyright, patent, and trademark.
2	Define and identify examples of attribution, plagiarism, and piracy in computing and digital media visuals.
3	**Analyze examples of plagiarism, copyright infringement, piracy, trademark infringement, and patent violation. Determine if inappropriate use has occurred and why.**
4	Define and describe the public domain. Research and explain the consequences for violating attribution laws.
5	Define and model appropriate attribution for using another programmer's code by sharing code with peers and following appropriate attribution guidelines.

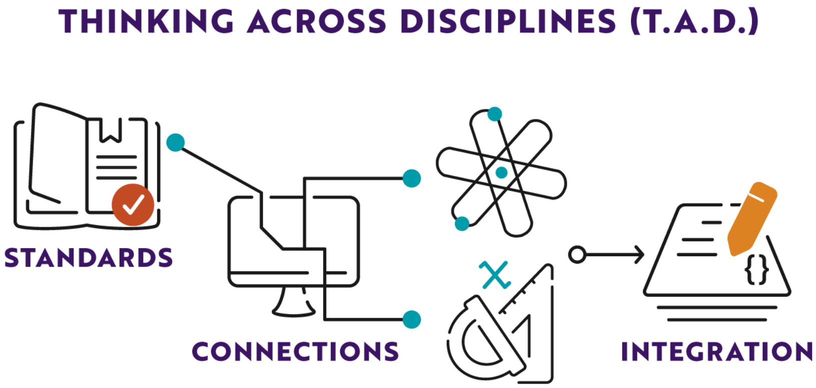
E.AP.4E. Describe and justify the steps taken and choices made during a program's development.	
Grade	**Complexity Statement**
K	Sequence steps to a familiar process using images and collaboratively discuss the reasoning behind the chosen order.
1	Draw and create the steps of a familiar process. Exchange work with a peer and add additional step drawings on sticky notes to the author's work. Justify the changes made to the original sequence.
2	Compare and contrast the steps of solving a math problem or doing a science lab with the steps of coding. Explain the choices made in each situation and how they can impact the task.
3	**Create a group presentation on how to modify erroneous code to debug a program.**
4	Define commenting on code with examples. Using a program that has multiple errors (printed on paper), annotate changes or suggestions to debug the code on the margins of the paper.

E.AP.4E. Describe and justify the steps taken and choices made during a program's development.	
Grade	**Complexity Statement**
5	Define clarity in written comments for other programmers. Given several examples and non-examples of annotated code, test each and provide written feedback on the effectiveness of the annotations.

E.AP.4F. Using an iterative process, test a program step-by-step and document areas of refinement.	
Grade	**Complexity Statement**
K	Examine a process that is lengthy and capable of being shortened with teacher guidance. Identify which parts can be removed to shorten the process and still attain the objective.
1	Define and apply the term iterative process. Through multiple rounds of testing and refinement, collectively develop an optimized solution to a problem.
2	Using the iterative process, refine an error-filled code in groups. Document the changes and compare the work with another set of peers. As a class, identify and describe similarities and differences in the strategies tried and why.
3	**Partner to test and debug multi-step, block-based code containing flawed conditionals, loops, and sequences intended for a figure or robot movement, and complete a specified task. Explain modifications.**
4	Use common testing strategies to test and refine previously created student code. Target a specific area of improvement and justify modifications.
5	Collaborate to iteratively improve a complex, multi-step program. Document all modifications and provide justification.

Thinking Across Disciplines in Grade 3

Core Concept 4: Algorithms and Programming



The standards included in **Core Concept 4: Algorithms and Programming** focus on the foundational programming skills, including creating and manipulating variables, designing and refining algorithms to solve tasks, and understanding control structures. These standards emphasize problem-solving through decomposition, writing, testing, debugging, and collaborating on programs while promoting ethical practices through proper attribution. These skills are closely tied to concepts teachers already address in their classrooms.

- Through decomposition, students break complex problems into smaller, manageable pieces solvable with step-by-step instructions, providing a practical context for practicing **problem-solving skills**. Additionally, using an iterative approach to test, refine, and improve solutions cultivates persistence and adaptability in problem-solving.
- When working with a team of peers, students build **collaboration skills** that support academic growth as well as social and emotional development. The collaborative development of solutions enhances collective creativity. Using the work of others and respecting intellectual property supports academic integrity in creating and modifying work.
- **Computational thinking** is a way of expressing solutions to problems as steps that can be carried out. Just as computers need step-by-step instructions to process information, humans use similar processes to complete tasks.

T.A.D. Examples

Standard	**E.AP.1A. Create clearly named variables representing different data types and perform operations on the variables' values.** **Complexity Statement:** Solve real-world problems involving perimeters of polygons, including finding the perimeter given the side lengths and finding an unknown side length. **Core Practices:** Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.MD.D.8 Solve real-world and mathematical problems involving perimeters of polygons, including finding the perimeter given the side lengths, finding an unknown side length, and exhibiting rectangles with the same perimeter and different areas or with the same area and different perimeters.
Integration	Computer science and mathematics both involve problem-solving and pattern recognition. Calculating perimeters and exploring shape relationships builds the logical thinking used in coding and algorithm design. In math, students can measure the perimeter of various polygons inside of the classroom (desk, chair seat, door, etc.). They can work with a partner by swapping the measurements of the side lengths to find the perimeter of the object they measured. Then they can switch with another partner and provide one side length and the perimeter to have them find the missing side length. Finally, they can compare their results with students who measured similar-sized perimeters.

Standard	**E.AP.1B. Create, use, and apply an algorithm to complete a task. Compare the results of algorithm usage trials and refine the algorithm.** **Complexity Statement:** Solve problems involving the four operations, and identify and explain patterns in arithmetic. **Core Practices:** Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing

Connections	**Mathematics** 3.OA.D.8 Solve two-step word problems using the four operations. Represent these problems using equations with a letter standing for the unknown quantity. Assess the reasonableness of answers using mental computation and estimation strategies, including rounding.[3]
Integration	Solving multi-step problems involves a step-by-step process. Students could write the process in steps, then work together to review and refine the process. Developing and refining these steps (algorithms) is a skill used in computer science.

Standard	**E.AP.2A. Define what a control structure is and create programs that include sequences, conditionals, events, and loops.** **Complexity Statement:** Code a character or robot using block-based programming to make the figure move based on conditionals. Define and explain conditionals. **Core Practices:** Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.G.A.2 Partition shapes into parts with equal areas. Express the area of each part as a unit fraction of the whole. *For example, partition a shape into 4 parts with equal area, and describe the area of each part as 1/4 of the area of the shape.*
Integration	As students are writing an algorithm for drawing a quadrilateral, they can divide the shape into equal parts with equal area by creating smaller parts that stand alongside each other to create the larger quadrilateral, creating the fractional pieces that equal the whole. In a block-based program, students can code a multi-step set of instructions to have the robot create the shape they created on paper. While creating the code, the students will need to use if-then structures to ensure the turns are made in the right direction to accurately create the figure.

Standard	**E.AP.3A. Define and apply decomposition to a complex problem in order to create smaller subproblems that can be solved through step-by-step instructions.** **Complexity Statement:** Diagram and determine the appropriate steps to achieve a mathematical solution to a multi-step word problem. **Core Practices:** Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.NF.A Develop an understanding of fractions as numbers.
Integration	In math, students can combine fractional pieces to create a whole, then explain the steps they took to combine the parts. This can also be transferred to a block-based program where students create parts of an object, then create a code to combine the parts as a whole.

Standard	**E.AP.3B. Modify, remix, or incorporate parts of an existing problem's solution to develop something new or add more advanced features to a program.** **Complexity Statement:** Collaborate to design and implement a step-by-step plan for completing a task. Exchange plans, test, and provide feedback to peers to refine the plan. **Core Practices:** Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.G.A.1 Understand that shapes in different categories (e.g., rhombuses, rectangles, and others) may share attributes (e.g., having four sides), and that the shared attributes can define a larger category (e.g., quadrilaterals). Recognize rhombuses, rectangles, and squares as examples of quadrilaterals, and draw examples of quadrilaterals that do not belong to any of these subcategories.
Integration	Creating a step-by-step plan to draw a quadrilateral allows teachers and students to evaluate learning and to target misconceptions. For example, students can write a step-by-step plan (algorithm) to draw a quadrilateral and then collaborate with peers to assess, improve, and modify the algorithm to create different quadrilaterals.

Standard	**E.AP.4A. Create a simple program to achieve a goal with expected outcomes.** **Complexity Statement:** Collaborate to propose solutions for each stage of a multi-stage problem or scenario. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.OA.A.3 Use multiplication and division within 100 to solve word problems in situations involving equal groups, arrays, and measurement quantities, e.g., by using drawings and equations with a symbol for the unknown number to represent the problem. [1]
Integration	Collaborating to propose solutions is an important skill used in each core content. In math, students can work in groups to create projects in teams. Each team needs a certain number of objects to represent equal rows and columns. Students can propose solutions for leftover pieces, such as creating a different array or getting a different number of objects.

Standard	**E.AP.4B. Test and debug a program or algorithm to ensure the program produces the intended outcome.** **Complexity Statement:** Test a multi-step, block-based code intended to make a figure or robot move but contains flawed conditionals. Work with a partner to debug, test, and explain how to correct the conditionals. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.OA.B.5 Apply properties of operations as strategies to multiply and divide. *Examples: If 6 x 4 = 24 is known, then 4 x 6 = 24 is also known. (Commutative property of multiplication.) 3 x 5 x 2 can be found by 3 x 5 = 15, then 15 x 2 = 30, or by 5 x 2 = 10, then 3 x 10 = 30. (Associative property of multiplication.) Knowing that 8 x 5 = 40 and 8 x 2 = 16, one can find 8 x 7 as 8 x (5 + 2) = (8 + 5) + (8 x 2) = 40 + 16 = 56. (Distributive property.)*

Integration	Students can test a flawed code that makes a figure move forward based on a larger group size or backward based on a larger number of objects in a group. Students identify the errors in the code that prevent the figure from moving correctly and explain how to correct the algorithm, then correct the code to test if that fixed the problem. In mathematics, students can be given a solved problem with an error. Students would use the criteria to solve the problem to find the error. This process can then be transferred to a block-based coding program with an intentional error that students work to identify.

Standard	**E.AP.4C. Collaborate with a team of peers to design, implement, test, and review the stages of program development.** **Complexity Statement:** Collaborate with peers to test and debug (flawed code provided by the teacher). Once complete, write a story that provides an overview of the code's function and how to resolve errors. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** 3-PS2-1 Plan and conduct an investigation to provide evidence of the effects of balanced and unbalanced forces on the motion of an object.
Integration	When students have identified the error in a problem or program, they can document the mistake that was made. This could be set up in stations around the classroom. After students have visited all groups, they can compare the explanations of identified errors and how they explained the solution.

Standard	**E.AP.4D. Identify intellectual property rights and apply the appropriate attribution when creating or remixing programs.** **Complexity Statement:** Analyze examples of plagiarism, copyright infringement, piracy, trademark infringement, and patent violation. Determine if inappropriate use has occurred and why. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing

Connections	**Science** Science & Engineering Practices 8: Obtaining, evaluating, and communicating information.
Integration	When obtaining information for scientific research, it is important to practice ethical use of information and ideas. In science, students can research inventors of simple machines and document the sources of their findings. Teachers can introduce the idea of plagiarism by using examples such as copying a friend's drawing versus being inspired by it and making your own. Students can then invent their own simple machine, being sure to cite where their ideas came from.

Standard	**E.AP.4E. Describe and justify the steps taken and choices made during a program's development.** **Complexity Statement:** Create a group presentation on how to modify erroneous code to debug a program. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.OA.D.8 Solve two-step word problems using the four operations. Represent these problems using equations with a letter standing for the unknown quantity. Assess the reasonableness of answers using mental computation and estimation strategies, including rounding. 3.MP.3 Construct viable arguments and critique the reasoning of others.
Integration	Once students have identified an error in an incorrectly solved word problem, they should describe the original expression or equation. Students should identify any variables and errors, and also suggest modifications that would correct the error. They can create a presentation for peers to explain and justify the modifications.

Standard	**E.AP.4F. Using an iterative process, test a program step-by-step and document areas of refinement.** **Complexity Statement:** Partner to test and debug multi-step, block-based code containing flawed conditionals, loops, and sequences intended for a figure or robot movement, and complete a specified task. Explain modifications. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 3.G.A.2 Partition shapes into parts with equal areas. Express the area of each part as a unit fraction of the whole. *For example, partition a shape into 4 parts with equal area, and describe the area of each part as 1/4 of the area of the shape.*
Integration	Designing a program is a cyclical process that involves designing, testing, feedback, and refinement. This process should be documented as refinements are made. Once students have identified an error in a problem, created a solution, and shared it with peers, they can swap problems to see if they come up with the same error and solution. Groups can then collaborate around their solutions to create a final decision on the most efficient way to fix the error and generate a new code to solve the error.

Core Concept 5: Impacts of Computing (IC)

Overview

The impacts of computing can be positive and negative, enabling innovation, communication, and access to information, while also raising concerns around ethics, privacy, and fairness. Individuals and communities not only shape computing systems through interactions, behaviors, industry practices, and laws, but are also shaped by interactions with computing systems. Computer and data science create new means of communication by accelerating information exchange and establishing cyberspace as a dynamic workspace for individuals.

Elementary Standards for Core Concept 5: Impacts of Computing

Subconcepts and Core Practices	Elementary (K-5)
Subconcept 1: Intellectual Achievements *Core Practices: Fostering responsible cyber citizenship; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing*	E.IC.1A. Describe how computing has changed the ways people live and work.
Subconcept 2: Social Interaction *Core Practices: Fostering responsible cyber citizenship; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing*	E.IC.2A. Identify and describe examples of appropriate versus inappropriate computer communications.
	E.IC.2B. Identify examples of cyberbullying with age-appropriate responses.
Subconcept 2: Laws, Safety, and Industry Practices *Core Practices: Fostering responsible cyber citizenship; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing*	E.IC.3A. Explain how online actions have real-world consequences and that laws and rules may also apply online.
	E.IC.3B. Describe the safe versus unsafe uses of computing systems at age-appropriate levels.
	E.IC.3C. Explain how the school and school system's computing rules and policies keep students safe.

Complexity Statements

Core Concept 5: Impacts of Computing

Complexity statements for the standards under **Core Concept 5: Impacts of Computing** break down each standard into manageable, age-appropriate components. This breakdown ensures that instruction fits the developmental needs of students at each grade level. For instance, in kindergarten, the focus is on understanding basic computer usage rules and recognizing how computing systems are used daily. By grade 5, students describe modifications to increase usability and accessibility, explain age restrictions on digital communication, and model appropriate online behaviors and responses to issues like cyberbullying.

E.IC.1A. Describe how computing has changed the ways people live and work.	
Grade	**Complexity Statement**
K	List and describe ways that computing systems are used by students every day. With teacher guidance, identify how selected examples may decrease student workload.
1	Sketch or list all of the computing systems that are in the classroom and one other school location. In groups, collaboratively discuss how these systems and students interact.
2	Compare and contrast doing a task with and without computing systems (e.g., reading a story from a book versus on a digital screen) and discuss the pros and cons of the task completion each way.
3	**Explain how an invention has changed through the addition of a computing system (e.g., electric power, a car, the telephone, or an airplane).**
4	Identify and describe a career that relies on technology.
5	Describe the ways that computing systems can be modified to increase usability and accessibility.

E.IC.2A. Identify and describe examples of appropriate versus inappropriate computer communications.	
Grade	**Complexity Statement**
K	Discuss and model working respectfully and responsibly.
1	Explain when to and when not to share login information.
2	Create a class list of feedback comments that offer constructive feedback and practice using them in class.

E.IC.2A. Identify and describe examples of appropriate versus inappropriate computer communications.	
Grade	**Complexity Statement**
3	**Collaborate as a group to design a digital presentation and have students point out, critique, and suggest feedback to peers via comments.**
4	Examine computational communication examples and identify where appropriate and inappropriate comments are made.
5	Explain why certain digital communication applications are age-restricted.

E.IC.2B. Identify examples of cyberbullying with age-appropriate responses.	
Grade	**Complexity Statement**
K	Describe what cyberbullying is and who to tell if it happens.
1	Identify ways to respond to inappropriate cyber-communication.
2	Explain who bystanders, allies, and upstanders are with example scenarios.
3	**Classify examples of online behaviors as either being cyberbullying or a normal part of the computing conversation.**
4	Explain what responsible digital citizenship is.
5	Create presentations on what to do and who to contact if you are cyberbullied.

E.IC.3A. Explain how online actions have real-world consequences and that laws and rules may also apply online.	
Grade	**Complexity Statement**
K	Explain that there are rules for using computers in comparison to real-world classroom rules and examples.
1	Describe how making threats or using abusive language can lead to legal consequences.
2	Explain why it is not acceptable to purchase items online without the proper permissions.

E.IC.3A. Explain how online actions have real-world consequences and that laws and rules may also apply online.	
Grade	**Complexity Statement**
3	**Define copyright and explain why permission is needed before using protected intellectual property such as images and digital media.**
4	Describe what unauthorized access is and how participating in it can lead to legal repercussions (e.g., hacking, jailbreaking phones, illegal sharing of copyrighted materials).
5	Research relevant computing laws and create awareness posters for peers.

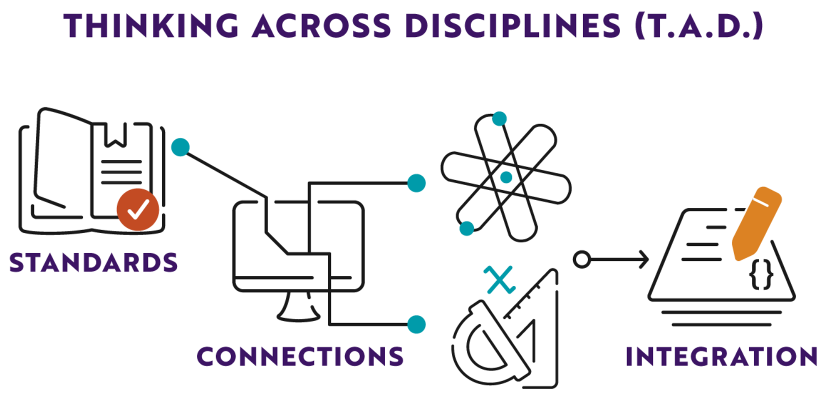
E.IC.3B. Describe the safe versus unsafe uses of computing systems at age-appropriate levels.	
Grade	**Complexity Statement**
K	Review and list responsible and safe online behaviors each time computing systems are used.
1	Explain why clicking on pop-up ads is unsafe.
2	List and categorize safe versus unsafe uses of computing systems.
3	**Propose and apply collaborative computing norms for group projects.**
4	Describe what computer viruses are and how to protect against them.
5	Explain the risks of responding to or chatting with unknown individuals online.

E.IC.3C. Explain how the school and school system's computing rules and policies keep students safe.	
Grade	**Complexity Statement**
K	Discuss and model the rules for computer use at school.
1	Explain why the use of another student's login is not allowed.
2	Identify and discuss the appropriate use of the Internet under the school's computer usage rules.

E.IC.3C. Explain how the school and school system's computing rules and policies keep students safe.	
Grade	**Complexity Statement**
3	**Classify computer usage examples as appropriate or inappropriate under the school's computer usage plan.**
4	Explain why downloading or uploading programs puts all users of the school's computer network at risk.
5	Describe and model the appropriate behaviors and practices for participating in virtual calls, collaborating with peers, and making public presentations.

Thinking Across Disciplines in Grade 3

Core Concept 5: Impacts of Computing



The standards in **Core Concept 5: Impacts of Computing** promote digital citizenship, online safety, and responsible computing practices. These standards emphasize understanding the impact of computing on daily life and work, recognizing appropriate and inappropriate online behavior, and addressing issues like cyberbullying with effective responses. Additionally, they highlight the importance of understanding online consequences, adhering to laws and rules, and following school policies to ensure safe and ethical use of technology. These skills are tied to concepts teachers already address in their classrooms.

- Interacting responsibly online in a safe environment builds upon student **communication skills**. Understanding safe and appropriate online behavior and the impacts of unethical behavior support effective communication strategies.
- **Responsibility and accountability** represent essential life skills that apply in both physical and digital contexts. Understanding responsible computing practices and the consequences of unethical behavior reinforces ethical decision-making.
- Distinguishing safe versus unsafe uses of computing systems at developmentally appropriate levels enables **critical judgment and decision-making skills** that protect personal information, maintain security, and promote responsible participation in digital environments

T.A.D. Examples

Standard	**E.IC.1A. Describe how computing has changed the ways people live and work.** **Complexity Statement:** Explain how an invention has changed through the addition of a computing system (e.g., electric power, a car, the telephone, or an airplane). **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** Science & Engineering Practices 8: Obtaining, evaluating, and communicating information.
Integration	Computing has expanded the possibilities in how people work and live. Students can research how computing systems have influenced innovation in everyday technologies (e.g, cars, telephones, and airplanes). After all presentations are complete, students should compare their research with their peers' findings. This comparison allows students to identify and incorporate any additional ways computing systems have improved their chosen invention that they may have initially overlooked, thus refining their own presentation.

Standard	**E.IC.2A. Identify and describe examples of appropriate versus inappropriate computer communications.** **Complexity Statement:** Collaborate as a group to design a digital presentation and have students point out, critique, and suggest feedback to peers via comments. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** Science & Engineering Practices 8: Obtaining, evaluating, and communicating information.
Integration	As students utilize computers to communicate, they must be taught how to interact responsibly. As students offer feedback to peers in the classroom, foster conversations about appropriate vs inappropriate comments orally. Use concrete examples and non-examples to illustrate the impact of feedback before students begin giving it to one another. Once students are prepared, encourage them to focus their feedback on how to improve the work rather than simply pointing out errors.

Standard	**E.IC.2B. Identify examples of cyberbullying with age-appropriate responses.** **Complexity Statement:** Classify examples of online behaviors as either being cyberbullying or a normal part of the computing conversation. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** Science & Engineering Practices 8: Obtaining, evaluating, and communicating information.
Integration	Responsible digital citizenship is using the internet in a responsible way that is legal, safe, and respectful. As students begin to engage online, it is important that they can explain what responsible digital citizenship is. For example, when students are providing feedback to peers, discuss the importance of creating a respectful working environment that is fair and incorporates the ideas of all members of the group.

Standard	**E.IC.3A. Explain how online actions have real-world consequences and that laws and rules may also apply online.** **Complexity Statement:** Define copyright and explain why permission is needed before using protected intellectual property such as images and digital media**.** **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** Science & Engineering Practices 8: Obtaining, evaluating, and communicating information.
Integration	To understand responsible digital citizenship, students must learn about the rules and laws governing online behavior. Students should receive explicit instruction on copyright and the significance of crediting sources for information or data used in their work. In science, teachers can model proper citation of sources in research findings and demonstrate methods for verifying a source's validity.

Standard	**E.IC.3B. Describe the safe versus unsafe uses of computing systems at age-appropriate levels.** **Complexity Statement:** Propose and apply collaborative computing norms for group projects. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** Science & Engineering Practices 1: Asking questions and defining problems
Integration	Students can create a set of class norms at the beginning of the school year for group collaboration. This process can involve discussing and defining appropriate online behaviors, such as never logging into a peer's account, always citing sources when using others' data, providing constructive and encouraging feedback in digital messages, and only sharing original work. The teacher can facilitate this by developing a set of guiding questions based on these practices, culminating in a finalized list of agreed-upon class norms.

Standard	**E.IC.3C. Explain how the school and school system's computing rules and policies keep students safe.** **Complexity Statement:** Classify computer usage examples as appropriate or inappropriate under the school's computer usage plan. **Core Practices:** Fostering responsible cyber citizenship; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** Science & Engineering Practices 8: Obtaining, evaluating, and communicating information.
Integration	One way schools create safe environments for students is by creating and implementing computing rules and policies. For example, teachers can share school policies before allowing students to use computing devices and ask students to share ways that they are kept safe by this policy. Teachers can share scenarios with students and ask them to classify the examples as appropriate or inappropriate. Having students sign to acknowledge that they understand the rules and policies, and slide it into their top computer cases, is one way to remind students of what they agreed to in being safe with their device.