Computer Science Implementation Framework

Grade 5

2026

LOUISIANA DEPARTMENT OF EDUCATION

Table of Contents

Background.......... 3

Finding Connections to Core Content Areas.......... 4

Components of the Computer Science Implementation Framework..... 5

Core Concept 1: Computing Systems (CS).......... 7

Complexity Statements.......... 8

Thinking Across Disciplines in Grade 5.......... 10

Core Concept 2: Networks and the Internet (NI).......... 14

Complexity Statements.......... 15

Thinking Across Disciplines in Grade 5.......... 17

Core Concept 3: Data and Analysis (DA).......... 20

Complexity Statements.......... 22

Thinking Across Disciplines in Grade 5.......... 28

Core Concept 4: Algorithms and Programming (AP).......... 36

Complexity Statements.......... 38

Thinking Across Disciplines in Grade 5.......... 45

Core Concept 5: Impacts of Computing (IC).......... 55

Complexity Statements.......... 56

Thinking Across Disciplines in Grade 5.......... 60

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 Louisiana 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 responsibility, 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 5 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.

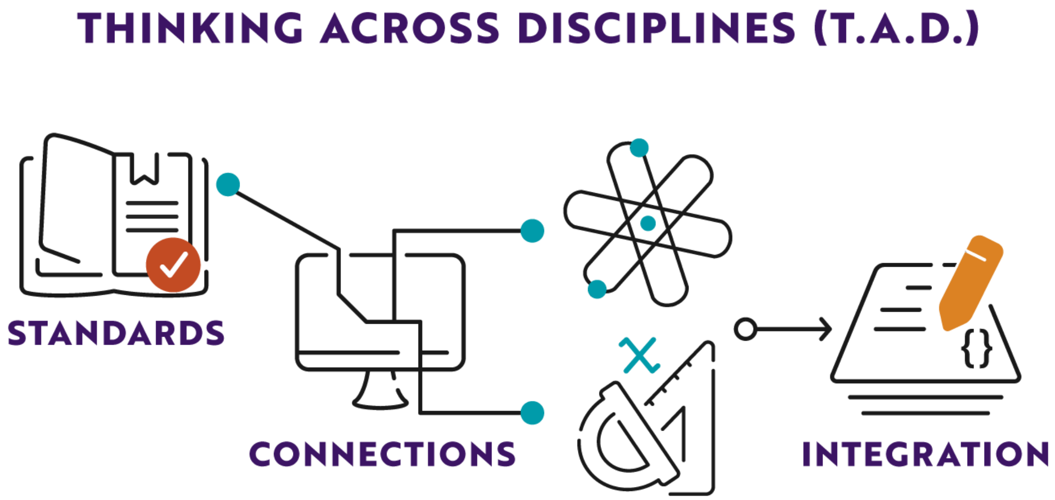
Complexity statements are foundational for effectively integrating computer science by providing a clear, grade-level-specific model for instruction. By outlining the increasing depth and rigor of learning expectations across grade levels, these statements ensure computer science concepts are introduced 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 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.**

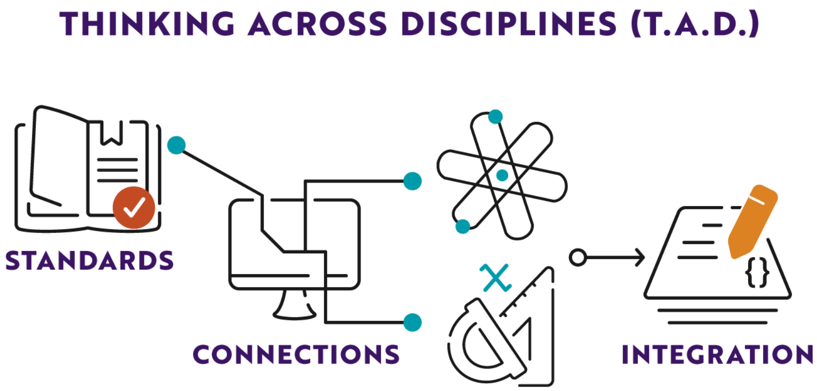
E.CS.1B. Identify and select the appropriate software to complete computing tasks.	
Grade	**Complexity Statement**
K	Describe the ways people use software to perform computing tasks.
1	Explain the function of common software components.
2	Compare and contrast the functions and uses of various software.
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 5

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 the 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:** Justify and use the proper hardware to complete a given task. **Core Practices:** Collaborating Around Computing; Recognizing and Defining Computation Problems; Communicating about Computing
Connections	**Mathematics** 5.MD.A.1 Convert among different-sized standard measurement units within a given measurement system and use these conversions in solving multi-step, real-world problems (e.g., convert 5 cm to 0.05 m; 9 ft to 108 in). **Science** 5-ESS1-2 Represent data in graphical displays to reveal patterns of daily changes in length and direction of shadows, day and night, and the seasonal appearance of some stars in the night sky.
Integration	Just as tools in mathematics and science have capabilities and limitations, so does computing hardware. With computer science, students may choose the appropriate hardware device, such as a monitor, keyboard, or printer, to complete tasks. In mathematics, students can participate in a virtual field trip in which they compare the sizes of two national parks given in different units (e.g., acres and square kilometers). Students apply their understanding of unit conversion to convert both measurements into square miles. To enter information on the computer, students likely use a keyboard for accurate typing. To move around the map and compare distances, students may use a mouse or a trackpad. Students' choice of these input and calculation tools affects how clearly and effectively they present their comparison of the park sizes. In science, students may choose the appropriate tools to report their findings on shadow patterns observed throughout the day. Some students may create and display their data on a smart board to view their content clearly, while other students may use a tablet to annotate a picture or video taken with a mobile device.

Standard	**E.CS.1B. Identify and select the appropriate software to complete computing tasks.** **Complexity Statement:** Justify and use the proper software to complete a given task. **Core Practices:** Collaborating Around Computing; Recognizing and Defining Computation Problems; Communicating about Computing
Connections	**Mathematics** 5.MD.B.2 Make a line plot to display a data set of measurements in fractions of a unit (1/2, 1/4, 1/8). Use operations on fractions for this grade to solve problems involving information presented in line plots. For example, given different measurements of liquid in identical beakers, find the amount of liquid each beaker would contain if the total amount in all the beakers were redistributed equally. **Science** 5-ESS1-2 Represent data in graphical displays to reveal patterns of daily changes in length and direction of shadows, day and night, and the seasonal appearance of some stars in the night sky.
Integration	Like mathematical and scientific tools, computing software has strengths and limitations. In a mathematics class, students can choose appropriate software, such as a drawing application, to organize data and create a line plot rather than drawing one on paper. In science, students can use software programs such as a camera application to take pictures or record videos to collect data on shadow patterns. Students may also use a spreadsheet program, such as Excel, to collect and record data that compares daily sunrise and sunset times, as well as track which constellations may appear in the night sky in their area during different times of the school year. Students can select software that helps organize and display data for the patterns they observe over the course of the school year.

Standard	**E.CS.1C. Evaluate hardware and software types to meet users' needs in completing various computing tasks.** **Complexity Statement:** Evaluate a user's unique needs and recommend the most appropriate hardware and software in a given scenario. **Core Practices:** Collaborating Around Computing; Recognizing and Defining Computation Problems; Communicating about Computing

Connections	**Mathematics** 5.NBT.B.7 Add, subtract, multiply, and divide decimals to hundredths, using concrete models or drawings and strategies based on place value, properties of operations, and/or the relationship between addition and subtraction; justify the reasoning used with a written explanation. **Science** 5-ESS2-1 Develop a model using an example to describe ways the geosphere, biosphere, hydrosphere, and/or atmosphere interact.
Integration	Considering the available tools and deciding which best fits the task is important when solving a mathematical problem, collecting data, or conducting research. For example, in mathematics class, students may choose whether to create a model or a drawing using tools like Google Drawings on a tablet to support their reasoning. Students may also use a software program such as Google Slides or PowerPoint to create a presentation that explains the reasoning behind their research. In science, students can decide how to connect the relationships between all four of Earth's major systems by discussing with the group the desired result and choosing the correct hardware and software to create a video presentation, or by creating a presentation on a laptop or desktop using software of their choice.

Standard	**E.CS.2A. Propose potential ways to address computing problems using appropriate hardware or software.** **Complexity Statement:** Propose and test a plan to address basic computing problems using hardware or software when given various error messages. **Core Practices:** Collaborating Around Computing; Recognizing and Defining Computational Problems; Communicating about Computing
Connections	**Science** 5-ESS3-1 Generate and compare multiple solutions about ways individual communities can use science to protect the Earth's resources and environment.
Integration	In science, students can research current solutions online that communities use to protect Earth's resources and environment. Students can select a computational method to generate their own solutions and share the method with the class to compare solutions. As hardware and software issues arise during research or while creating their solutions, encourage students to propose ways to resolve the problems by using prompts and examples.

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 cyber responsibility; 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 cyber responsibility; 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.

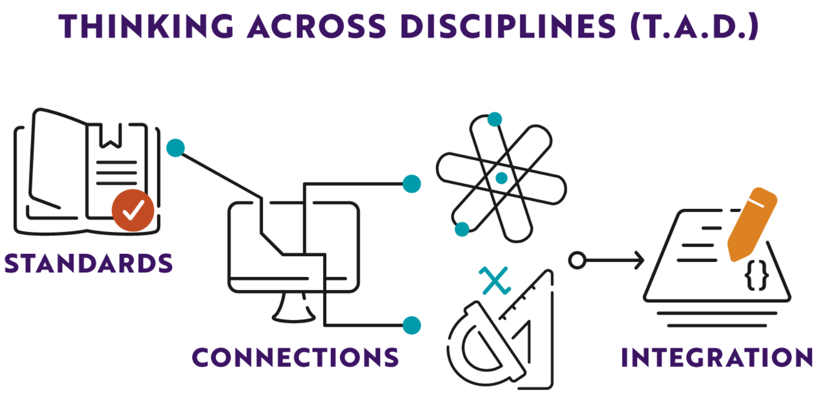
Grade	Complexity Statement
K	Distinguish between public and private information.
1	Define personally identifiable information (PII).
2	Evaluate the risks and benefits of sharing PII.
3	Evaluate mechanisms of password generation and best practices for password management.
4	Describe the role of authentication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

E.NI.2A. Describe personally identifiable information (PII) and identify practices for when and where sharing PII is appropriate.	
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 5

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:** Diagram and summarize how data can be shared between multiple networks and users via the Internet. **Core Practices:** Fostering cyber responsibility; 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** 5-LS2-1 Develop a model to describe the movement of matter among plants, animals, decomposers, and the environment.
Integration	Teachers can help students make connections between how multiple users on the same platform pass data across networks and how matter transfers between different organisms in an environment. In science, students can use a digital tool to create a simple food chain and then share it with another group or class through email, Google Classroom, or another learning management system. As students share their work, explain that matter moves through an ecosystem in a way that is similar to how information flows across networks on the internet.

Standard	**E.NI.2A. Describe personally identifiable information (PII) and identify practices for when and where sharing PII is appropriate.** **Complexity Statement:** Identify and determine the appropriate resources and individuals to notify if PII is inappropriately shared. **Core Practices:** Fostering cyber responsibility; 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** 5-ESS3-1 Generate and compare multiple solutions about ways individual communities can use science to protect the Earth's resources and environment.

Integration	When scientific data or findings are shared on one or multiple platforms, the content may sometimes include information that is incomplete, misleading, or inappropriate. Just as scientists ensure their research is accurate and shared responsibly, students need to understand appropriate ways to recognize and correct problematic content, especially when it involves data they or others have collected. This includes understanding what personally identifiable information (PII) is (such as names, addresses, and personal health details) and why it must be protected. In science, students may research and generate solutions for how communities can use science to protect Earth's resources. During this process, they may collect data from local observations, surveys, or research sources. As students prepare to share their findings, they should determine which information is relevant and safe to include and which details, especially those containing PII, should not be shared. By focusing only on data that supports their solutions while protecting privacy, students practice responsible science communication and appropriate data sharing, ensuring their work is clear, accurate, and factual.

Standard	**E.NI.2B. Identify ways to maintain data security when using networks.** **Complexity Statement:** Explain the ways a student can protect their home and school networks. **Core Practices:** Fostering cyber responsibility; 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** 5-PS3-1 Use models to describe that energy in animals' food (used for body repair, growth, motion, and to maintain body warmth) was once energy from the sun.
Integration	Students need to learn how to stay safe online by using tools and responsible habits that protect personal information while using the internet at home on a private network and in public settings. In science, students can explore how some food choices may harm the body even when intended to provide nourishment and energy. In a food chain, animals compete for food. As students conduct online research on food chains, teachers can discuss ways students protect their personal information, such as their address or passwords, in order to keep their data secure on a computer network.

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 cyber responsibility; 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 cyber responsibility; 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.
1	Generate measurement data by measuring the lengths of various objects, then record the data computationally with units of support.

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**
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 formats 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.

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**
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.**

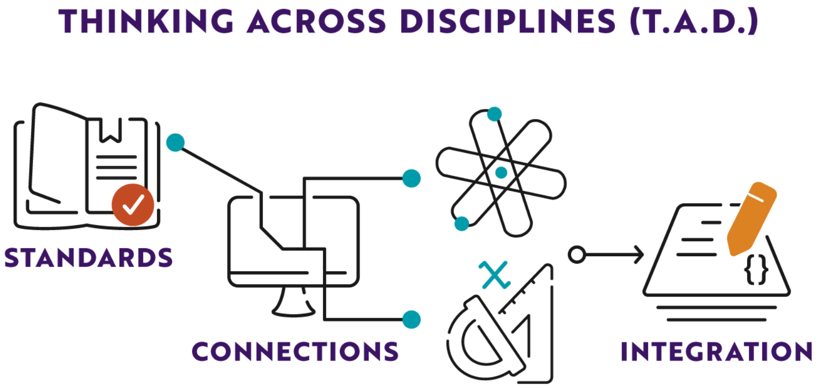
E.DA.4B. Evaluate data quality and clean data when indicated using the criteria of validity, accuracy, completeness, consistency, and uniformity.	
Grade	**Complexity Statement**
K	Recognize data used to determine valid groupings.
1	Recognize the need for consistency and completeness when evaluating data.
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.
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 5

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:** Generate line graphs from data that is collected and entered into data tables to support claims using computational tools. **Core Practices:** Collaborating around computing; Recognizing and defining computational problems; Creating computational artifacts
Connections	**Mathematics** 5.MD.B.2: Make a line plot to display a data set of measurements in fractions of a unit (1/2, 1/4, 1/8). Use operations on fractions for this grade to solve problems involving information presented in line plots. **Science** 5-LS1-1 Ask questions about how air and water affect the growth of plants.
Integration	Gathering data to create line graphs can help strengthen and support the claims being made. In science, students can design and conduct an investigation growing different types of bean plants under the same conditions (e.g., same amount of water). Height data can be collected in fractions of an inch over several weeks to track plant growth. In mathematics, students can plot their plant heights throughout the weeks. They can then use a tool (such as Google Sheets or Microsoft Excel) to build a digital table from which they can generate a line graph from their collected data.

Standard	**E.DA.1B. Classify types of data and describe the attributes used to sort data.** **Complexity Statement:** Identify specific types of data, such as metadata, and explain what the information means, including its application. **Core Practices:** Collaborating around computing; Recognizing and defining computational problems; Creating computational artifacts
Connections	**Mathematics** 5.MD.C.3 Recognize volume as an attribute of solid figures and understand concepts of volume measurement. **Science** 5-ESS2-2 Describe and graph the amounts and percentages of water and fresh water in various reservoirs to provide evidence about the distribution of water on Earth.

Integration	When collecting measurement data (i.e., length, mass, volume, time), understanding units is a basic form of metadata. The unit of measure provides context about the numerical value. When students convert units, they are essentially working with different representations of the same underlying data. In mathematics, students can be introduced to different 3D shapes to observe and collect data on their attributes, such as volume. Students can design a table for each shape attribute (i.e., name, number of faces, etc.) to demonstrate how collecting information and sorting by attributes is the same as sorting different types of data. In a science lesson, students can identify types of data by graphing different percentages of water located in various sources (i.e., oceans, lakes, rivers, groundwater, ice caps, etc.) in order to compare the amount of water in each.

Standard	**E.DA.2A. Select the appropriate data collection tool and technique to gather data to support a claim or communicate information.** **Complexity Statement:** Compare and contrast data sets collected from two or more instruments and use the data to support a claim. **Core Practices:** Fostering cyber responsibility; 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** 5-PS1-3 Make observations and measurements to identify materials based on their properties. **Mathematics** 5.NBT.B.7 Add, subtract, multiply, and divide decimals to hundredths, using concrete models or drawings and strategies based on place value, properties of operations, and/or the relationship between addition and subtraction; justify the reasoning used with a written explanation.
Integration	In science, students can be given unknown substances such as baking soda, sugar, and flour to mix with vinegar. Before mixing, students can measure the mass using a scale and the temperature using a thermometer or probe. After mixing, students can again collect mass and temperature data and compare the data sets. Using the collected data, students can compare the reactions to a properties table to make a claim about the identity of the unknown substance.

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:** 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. **Core Practices:** Fostering cyber responsibility; 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** 5.MD.A.1 Convert among different-sized standard measurement units within a given measurement system and use these conversions in solving multi-step, real world problems (e.g., convert 5 cm to 0.05 m; 9 ft to 108 in).
Integration	When collecting data, there may be moments when data needs to be converted into the same unit of measurement before being calculated. In mathematics, students can solve multi-step problems by converting measurements of classroom objects, such as calculating whether a 1.5-meter desk and an 80-centimeter chair can fit along a 3-meter wall. All measurements must be converted to the same unit before the problem can be solved. In computer science, when inputting data into a chart on the computer, students must be aware that the computer cannot accurately analyze numbers with different units (e.g., inches versus feet). Students may need to convert the measurements so all data uses the same unit of measure.

Standard	**E.DA.3A. Compare and contrast ways to store data using technology.** **Complexity Statement:** Compare and contrast the storage size capacity for computational storage devices and select the appropriate format for given tasks. **Core Practices:** Fostering cyber responsibility; 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** 5.OA.A.2 Write simple expressions that record calculations with whole numbers, fractions, and decimals, and interpret numerical expressions without evaluating them. For example, express the calculation “add 8 and 7, then multiply by 2” as 2 (8 + 7). Recognize that 3 x (18,932 + 9.21) is three times as large as 18,932 + 9.21, without having to calculate the indicated sum or product.

Integration	Students can use multiplication expressions to understand the amount of storage needed to store various-sized files on different storage devices. In mathematics, students can be given information on multiple storage device capacities (e.g., USB 16 GB, SD Card 32 GB, external hard drive 1,000 MB). Questions students could discuss and answer include, "Which device would be best for storing 100 songs if each song is 4 MB? Which device would be best for storing 50 photos if each photo is 5 MB? Which device would be best for storing 10 movies if each movie is 8 MB?"

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:** 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. **Core Practices:** Fostering cyber responsibility; 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** 5-PS1-2 Measure and graph quantities to provide evidence that regardless of the type of change that occurs when heating, cooling, or mixing substances, the total amount of matter is conserved.
Integration	As students discuss changes to matter, various file types could be used in the activities. Students can analyze photos (JPEG) from an experiment where substances were heated, cooled, or mixed. Then write descriptions of the changes in a Word document. During their analysis, the students could enter data in a shared spreadsheet that compares mass before and after the change. During discussions, students can point out the file extensions and the type of information each file type holds.

Standard	**E.DA.4A. Organize and present data visually in at least three ways to highlight relationships and evaluate a claim.** **Complexity Statement:** Collect at least five data sets and use computational tools to create visualizations. Compare and contrast visualizations, findings, and conclusions with those of peers. **Core Practices:** Creating computational artifacts; Communicating about computing

Connections	**Science** Science and Engineering Practices 4: Analyzing and Interpreting Data
Integration	The process of collecting and interpreting data to support scientific claims is aligned with science and engineering practices. In science, students can collect data and computationally create different types of charts, such as pie charts, line graphs, and bar graphs, to determine if any patterns and/or relationships are evident. Students can then compare and contrast their digital charts with peers.

Standard	**E.DA.4B. Evaluate data quality and clean data when indicated using the criteria of validity, accuracy, completeness, consistency, and uniformity.** **Complexity Statement:** 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. **Core Practices:** Creating computational artifacts; Communicating about computing
Connections	**Science** 5-LS2-1 Develop a model to describe the movement of matter among plants, animals, decomposers, and the environment.
Integration	When students evaluate data, it should be accurate, consistent, and well-organized. If students identify any problems or inconsistencies in the data, they should receive guidance to help them understand and explain the reasons for the inconsistencies. In science, students can observe the number of birds in their local ecosystem during a set amount of time each day for four days and then computationally graph their data. Students can then compare data to determine the variation in each graph and discuss the validity, accuracy, completeness, consistency, and uniformity of the data collection. For example, did some students collect data from all around them or just from one spot?

Standard	**E.DA.5A. Utilize data to create models, answer investigative questions, and make predictions.** **Complexity Statement:** 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. **Core Practices:** Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing

Connections	**Mathematics** 5.G.A.2 Represent real-world and mathematical problems by graphing points in the first quadrant of the coordinate plane, and interpret coordinate values of points in the context of the situation. **Science** Science and Engineering Practice 2: Developing and Using Models Science and Engineering Practice 3: Planning and carrying out investigations
Integration	As students create graphs in mathematics, teachers can allow students to use graphing software to interpret points and analyze patterns. Integrating the use of computational tools to input, organize, and model data strengthens the connection to computer science. In science, students can record the changes in the temperature of water being heated over time and collect the data in a table. After creating a computational graph of the data, students can then make predictions on how much time it will take the water to reach the boiling point.

Standard	**E.DA.5B. Analyze data for patterns and relationships.** **Complexity Statement:** 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. **Core Practices:** Recognizing and defining computational problems; Developing and using abstractions; Creating computational artifacts; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 5.OA.B.3 Generate two numerical patterns using two given rules. Identify apparent relationships between corresponding terms. Form ordered pairs consisting of corresponding terms from the two patterns, and graph the ordered pairs on a coordinate plane. For example, given the rule "Add 3" and the starting number 0, and given the rule "Add 6" and the starting number 0, generate terms in the resulting sequences, and observe that the terms in one sequence are twice the corresponding terms in the other sequence. Explain informally why this is so. **Science** 5-ESS1-2 Represent data in graphical displays to reveal patterns of daily changes in length and direction of shadows, day and night, and the seasonal appearance of some stars in the night sky.

Integration	In mathematics, students can use a block-based coding program to explore patterns. By designing loops or repeat blocks, they can represent iteration and generate numerical sequences based on given rules. This reinforces the concept of using a rule to create a sequence and helps students understand how each term relates to its position in the pattern. Students can then compare their digitally generated sequences to those created by hand, deepening their understanding through both methods. In science, students can determine relationships of data collected when comparing changes in the direction of shadows. Over a school day, students can observe and record the length and direction of a shadow cast by a fixed object (e.g., a meter stick stuck upright in the ground or a designated object in the classroom near a sunny window). Students can then use a block-based coding s to generate a sun clock simulation using loops and repeat blocks to show the shadow moving across the clock.

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 cyber responsibility; 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 it contains flawed loops. Work with a partner to debug, test, and explain how to correct the loops.
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.

E.AP.4B. Test and debug a program or algorithm to ensure the program produces the intended outcome.	
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 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.

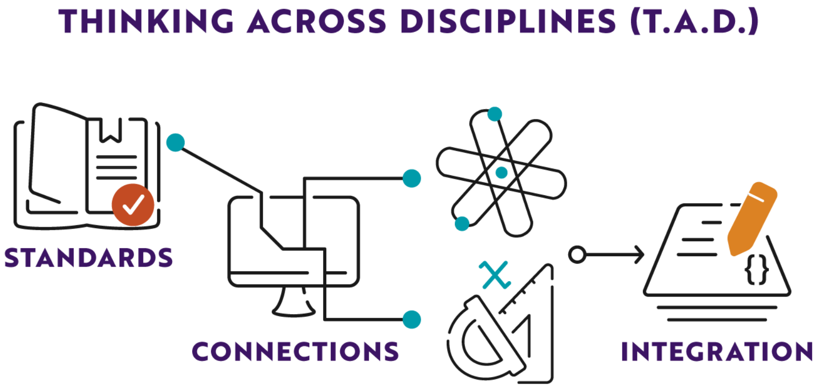
E.AP.4D. Identify intellectual property rights and apply the appropriate attribution when creating or remixing programs.	
Grade	**Complexity Statement**
2	Define and identify examples of attribution, plagiarism, and piracy in computing and digital media visuals.
3	Analyze examples of plagiarism, copyright 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.
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 5

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 personal 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:** 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. **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** 5.MD.C.5B Apply the formulas V = l x w x h and V = b x h for rectangular prisms to find volumes of right rectangular prisms with whole-number edge lengths in the context of solving real-world and mathematical problems.
Integration	Variables can represent unknown values, such as a missing side length needed to calculate the area of a rectangular prism. In mathematics, students can find the missing length of one side of a rectangular prism by solving for x (e.g., A rectangular box has a volume of 120 cubic inches. The width of the box is 4 inches, and the height is 5 inches. What is the length of the box?).

Standard	**E.AP.1B. Create efficient algorithms by structuring repeated actions using parentheses or brackets and optimizing lengthy instructions into simpler equations using multiplication, addition, and subtraction.** **Complexity Statement:** Create efficient algorithms by structuring repeated actions using parentheses or brackets and optimizing lengthy instructions into simpler equations using multiplication, addition, and subtraction. **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** 5.OA.A.1 Use parentheses or brackets in numerical expressions, and evaluate expressions with these symbols. 5.OA.A.2 Write simple expressions that record calculations with numbers, and interpret numerical expressions without evaluating them.

Integration	Finding solutions to mathematical problems can sometimes take multiple steps. To shorten the number of steps, parentheses or brackets can be used to define the order of repeated computations, shortening the mathematical algorithm. In mathematics, students are asked to calculate the tax on 4 items of clothing. Instead of calculating the cost of each item plus the tax individually, an algorithm can be used to simplify the equation. For example: Total Tax = (Item 1 + Item 2 + Item 3 + Item 4) * 0.09.

Standard	**E.AP.2A.** 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. **Complexity Statement:** When given a complex task, create a solution using block-based coding, test it, and compare the solution code with peers to refine it for efficient control structure usage. **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** 5.OA.A.2 Write simple expressions that record calculations with whole numbers, fractions, and decimals, and interpret numerical expressions without evaluating them. For example, express the calculation “add 8 and 7, then multiply by 2” as 2(8 + 7). Recognize that 3 x (18,932 + 9.21) is three times as large as 18,932 + 9.21, without having to calculate the indicated sum or product. **Science** 5-PS3-1 Use models to describe that energy in animals’ food (used for body repair, growth, motion, and to maintain body warmth) was once energy from the sun.
Integration	In mathematics, when solving constructed response problems, students can benefit from explaining and outlining the steps they used to solve complex expressions. Students can explore the consequences of choosing different operations to perform first; similarly, in computer coding, a program’s path is determined by a series of if-else decisions that lead to different outcomes. In science, modeling the flow of energy connects the role of if/then conditions in determining what occurs in food webs and food chains as the cycle continues in the biosphere. Students can create a simple interactive animation in a block-based coding environment (like Scratch) that models this energy flow. The animation could involve a sun sprite, plant sprites, and animal sprites. Students can then make adjustments to specific parts of the model and observe the impact of the change.

Standard	**E.AP.3A.** Decompose and create steps for solving complex problems that address various constraints such as cost, efficiency, length of time, number of people needed, etc. **Complexity Statement:** Decompose and create steps for solving complex problems that address various constraints such as cost, efficiency, length of time, number of people needed, etc. **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** 5.NBT.B.7 Add, subtract, multiply, and divide decimals to hundredths, using concrete models or drawings and strategies based on place value, properties of operations, and/or the relationship between addition and subtraction; justify the reasoning used with a written explanation.
Integration	Solving complex, multi-step problems is a fundamental skill that requires decomposing those problems into smaller, more manageable steps. This process helps identify what specific component needs to be addressed at each stage to ultimately find an overall solution. In mathematics, students can apply decomposition to a real-world scenario: determining the logistics of painting a mural on a school wall. This multi-step problem requires students to break down a large, daunting task into smaller, solvable components. Students can define the dimensions of the wall and what designs will be included, determine the material quantities for each color, and calculate the costs of all of the materials and supplies needed for the entirety of the project. Students are explicitly practicing the skills of decomposing a large, multi-faceted problem into smaller, sequential, sub-problems.

Standard	**E.AP.3B.** Modify pre-existing code by decomposing and identifying reusable parts to develop something new or add more advanced features. **Complexity Statement:** Using block-based coding, code (program) a multi-step set of instructions to make an animation for a story through the practices of decomposition, programming planning, and code optimization. (e.g., students decompose the story into parts, identify the actions and code needed on paper, review the code for parts that can be reused or recycled, and simplify the written code before adding it to the program. For each scene, students need a background, they need to place characters, and to program actions sequentially.) **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** 5.OA.B.3 Generate two numerical patterns using two given rules. Identify apparent relationships between corresponding terms. Form ordered pairs consisting of corresponding terms from the two patterns, and graph the ordered pairs on a coordinate plane. *For example, given the rule "Add 3" and the starting number 0, and given the rule "Add 6" and the starting number 0, generate terms in the resulting sequences, and observe that the terms in one sequence are twice the corresponding terms in the other sequence. Explain informally why this is so.* **Science** 5-ESS3-1 Generate and compare multiple solutions about ways individual communities can use science to protect the Earth's resources and environment.
Integration	By identifying apparent relationships between corresponding terms, students can determine patterns in code as well as when a part needs to be modified. The connection can be made to integrate as code is updated and new rules apply. In mathematics, students create a block-based program (e.g., in Scratch) that generates two numerical patterns based on user-defined rules and starting numbers. The program likely displays the two patterns side-by-side and might even plot the ordered pairs on a simple coordinate plane within the program. In science, connecting to new inventions/innovations connects to new ways that humans are discovering how to protect Earth's resources from depletion. Students could develop a pattern of conservation efforts for each day and make specific changes along the way.

Standard	**E.AP.4A.** Create a simple program to achieve a goal with expected outcomes. **Complexity Statement:** Create and propose a program to solve a problem with expected outcomes. **Core Practices:** Fostering cyber responsibility; 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** 5.NF.B.6 Solve real-world problems involving multiplication of fractions and mixed numbers, e.g., by using visual fraction models or equations to represent the problem.

Integration	Just as converting a fraction requires a specific sequence of operations, coding relies on step-by-step logic to solve complex problems. In mathematics, students can be presented with a real-world problem involving multiplying fractions or mixed numbers, such as: "A recipe for cookies calls for 2 1/2 cups of flour. If you want to make half of the recipe, how many cups of flour do you need?" Students then can create a simple block-based program (e.g., in Scratch) that follows a specific sequence of steps to calculate the amount of an ingredient needed when scaling a recipe.

Standard	**E.AP.4B.** 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. **Complexity Statement:** When given a complex task, create a solution using block-based coding and test it. Then, introduce up to 4 errors into the code and write it out (with errors on paper). Trade the flawed code with a peer. Test and debug the code. Reflect and write down what was challenging and why. **Core Practices:** Fostering cyber responsibility; 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** 5.NBT.B.7 Add, subtract, multiply, and divide decimals to hundredths, using concrete models or drawings and strategies based on place value, properties of operations, and/or the relationship between addition and subtraction; justify the reasoning used with a written explanation. **Science** Science and Engineering Practice 3: Planning and Carrying out Investigations Science and Engineering Practice 7: Engaging in Argument from Evidence
Integration	In mathematics, when working on multi-step problems, students can determine errors along the way and go back to their previous calculations to determine how to correct the error. This connects with debugging programs to make code function properly. Scientific investigations mirror the debugging process. When a test fails or yields unexpected results, students must isolate the flaw and retest. When students retest investigations to fix flaws, they are practicing the same troubleshooting skills needed to fix broken code. Additionally, by determining if evidence supports a claim, students learn to identify 'bugs' in their reasoning and make revisions.

Standard	**E.AP.4C.** Document the stages of the program development process while creating a coded solution to a scenario. **Complexity Statement:** Review the stages of program development. Consider a scenario with a real-world objective outcome. Follow the stages and document progress in writing appropriate code. **Core Practices:** Fostering cyber responsibility; 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** 5.NF.B.6 Solve real-world problems involving multiplication of fractions and mixed numbers, e.g., by using visual fraction models or equations to represent the problem.
Integration	In mathematics, teams of students can collaboratively design and create a simple block-based coding game that helps players practice multiplying fractions, using notes to document their process, track errors, and justify changes. This approach reinforces mathematical concepts and introduces elements of computational thinking. This process also aligns with the iterative approach of the software development lifecycle.

Standard	**E.AP.4D.** Define and model appropriate attribution for using another programmer's code by sharing code with peers and following appropriate attribution guidelines. **Complexity Statement:** Define and model appropriate attribution for using another programmer's code. Create simple programs and share the code with peers to incorporate into their own programs, and apply appropriate attribution guidelines to the merged code. **Core Practices:** Fostering cyber responsibility; 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** Standards for Mathematical Practices 3: Construct Viable Arguments and Critique the Reasoning of Others **Science** Science and Engineering Practice 8: Obtaining, Evaluating, and Communicating Information

Integration	When students collaborate on digital projects, they naturally combine their best ideas and refine their work. This collaborative process provides an opportunity to learn the essential computer science skill of providing attribution when using the work of others. Giving credit to the originators of ideas, data, or resources isn't just polite; it's a fundamental principle of digital responsibility and academic integrity. Students can create a short animation in a block-based coding program (e.g., Scratch) based on a mathematical or scientific concept. When they share their animations with classmates and merge content, they learn about acknowledging the source of all shared elements. In mathematics, students often justify their reasoning, especially when there is inaccurate information or when defending a specific approach. When creating animations that depict mathematical concepts (e.g., demonstrating fractions, geometric transformations, or number patterns), students can integrate mathematical ideas and cite the unit or specific lesson from which they have extracted the information. In science, students can create a portion of a science-themed animation (e.g., illustrating the water cycle, demonstrating plant growth, or explaining climate change) and then share their work with peers to complete the animation. As they share, they make sure to apply appropriate attribution to the original content creators.

Standard	**E.AP.4E.** 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. **Complexity Statement:** Define clarity in written comments for other programmers. Given several examples of annotated code (some correct and some incorrect), test each and provide written feedback on both the effectiveness of the annotations and the clarity. **Core Practices:** Fostering cyber responsibility; 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** Standards for Mathematical Practices 3: Construct Viable Arguments and Critique the Reasoning of Others **Science** Science and Engineering Practice 3: Planning and Carrying Out Investigations Science and Engineering Practice 8: Obtaining, Evaluating, and Communicating Information

Integration	In mathematics, students are expected to show their work when solving math problems, justifying their reasoning for each step and explaining why they chose a particular strategy. When using block-based code to create mathematical animations, ask students to write comments (annotate) the code for other programmers. When the code is shared with peers, ask peers to test the code and provide written feedback on both the effectiveness of the annotation and the clarity. In science, when planning an investigation or designing a solution, students make numerous choices about the procedure, materials, and variables. Students can describe the steps in their plan and justify why they chose that particular approach to answer their question or solve the problem.

Standard	**E.AP.4F.** Collaborate to iteratively improve a complex, multi-step program. Document all modifications and provide justification. **Complexity Statement:** Iteratively improve a given multi-step complex program in groups, documenting all modifications and the reasoning behind each change. Through a gallery walk, compare and contrast different approaches to improving the code, culminating in a discussion led by group leaders interpreting the goal of "improve the code." **Core Practices:** Fostering cyber responsibility; 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** Standards for Mathematical Practices 3: Construct Viable Arguments and Critique the Reasoning of Others **Science** Science and Engineering Practice 3: Planning and Carrying Out Investigations Science and Engineering Practice 8: Obtaining, Evaluating, and Communicating Information
Integration	In math, students can share solutions and steps to a mult-step problem. Students can review their classmates' work and provide feedback, suggestions, and support to help them improve. In science, the iterative process of testing and refining the code mirrors the pilot testing and improvement stages of scientific investigations, and improving the code to achieve a better outcome (smoother movement, bug-free navigation) aligns with the iterative nature of designing solutions.

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 cyber responsibility; 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 cyber responsibility; 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 cyber responsibility; 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 completing the task 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.
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 digital responsibility is.
5	**Create presentations on what to do and who to contact if you are cyberbullied.**

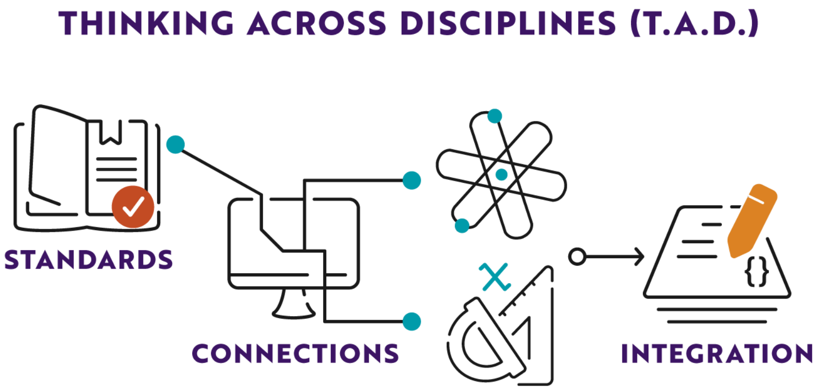
E.IC.3A. Explain how online actions have real-world consequences and that laws and rules may also apply online.	
Grade	**Complexity Statement**
K	Explain that there are rules for using computers in comparison to real-world classroom rules and examples.
1	Describe how making threats or using abusive language can lead to legal consequences.
2	Explain why it is not acceptable to purchase items online without the proper permissions.
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.
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 5

Core Concept 5: Impacts of Computing



The standards in **Core Concept 5: Impacts of Computing** promote digital responsibility, 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 supports 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:** Describe the ways that computing systems can be modified to increase usability and accessibility. **Core Practices:** Fostering cyber responsibility; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Mathematics** 5.OA.B.3 Generate two numerical patterns using two given rules. Identify apparent relationships between corresponding terms. Form ordered pairs consisting of corresponding terms from the two patterns, and graph the ordered pairs on a coordinate plane. **Science** 5-ESS3-1 Generate and compare multiple solutions about ways individual communities can use science to protect the Earth's resources and environment.
Integration	In mathematics, have students generate numerical pattern sequences and form ordered pairs to plot on a coordinate plane. Guide students in recognizing relationships within the patterns (e.g., the value of b is double the value of a). Show students how tools such as spreadsheets or coding languages allow users to modify how patterns are visualized to improve accessibility. Allow the students to adjust their work by adding specific colors, labels, and annotations. This process emphasizes how computing systems can be adjusted to present data more accessibly and improve clarity. In a science classroom, students can engage in a discussion about various ways communities impact Earth's resources and the environment (e.g., energy consumption, waste disposal, water usage, transportation). In small groups, students brainstorm multiple science-based solutions that communities could implement to address these environmental challenges. For each proposed environmental solution, groups generate and compare ways a computing system could be modified to enhance usability and accessibility. Through developing solutions to help protect the Earth's resources and the environment, students can identify and describe why technology must be adapted and how companies modify existing tools or create new products to provide new and innovative solutions that simplify everyday life.

Standard	**E.IC.2A. Identify and describe examples of appropriate versus inappropriate computer communications.** **Complexity Statement:** Explain why certain digital communication applications are age-restricted. **Core Practices:** Fostering cyber responsibility; 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 engage in online research, they need to learn the reasons some websites and applications include age restrictions. Through classroom discussion and communication with peers, students can develop a list of reasons certain sites have age restrictions and why some sites are considered appropriate for different age groups. In science, students can research the age requirements of several popular digital communication applications. Students can review app store descriptions, examine sections of the terms of service with teacher guidance, or consult reputable online resources that discuss digital safety for young users. After gathering information, students can identify reasons for the age restrictions to evaluate and communicate their findings with classmates.

Standard	**E.IC.2B. Identify examples of cyberbullying with age-appropriate responses.** **Complexity Statement:** Create presentations on what to do and who to contact if you are cyberbullied. **Core Practices:** Fostering cyber responsibility; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** Science & Engineering Practices 8: Obtaining, evaluating, and communicating information
Integration	Digital responsibility involves using the internet in responsible ways that are legal, safe, and respectful. As students engage online, they must understand and practice digital responsibility. In science, when students provide virtual feedback to peers, they plan conversations to address digital responsibility and responsible online behavior. Students can use programs such as PowerPoint or create a video to explain what actions to take if they experience cyberbullying and how to identify trusted individuals they could contact for support in those situations.

Standard	**E.IC.3A. Explain how online actions have real-world consequences and that laws and rules may also apply online.** **Complexity Statement:** Research relevant computing laws and create awareness posters for peers. **Core Practices:** Fostering cyber responsibility; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** Science & Engineering Practices 8: Obtaining, evaluating, and communicating information **Mathematics** 5.MD.B.2 Make a line plot to display a data set of measurements in fractions of a unit (1/2, 1/4, 1/8). Use operations on fractions for this grade to solve problems involving information presented in line plots.
Integration	Students should be aware of laws created for safe and responsible use of computers and the internet. In science, students can create posters that raise awareness about the consequences of not following laws and established rules when working on hardware and software platforms. In mathematics, students can apply the skills of creating and interpreting line plots by using real-world data about online behavior to develop an awareness poster. For example, students can research statistics on how often students encounter certain online rule violations (e.g., cyberbullying or misinformation) and then create a line plot to visually represent this data on the posters. The line plot serves as a powerful way to communicate the frequency of these events and helps illustrate the real-world consequences of these actions, making the abstract concept of online laws more concrete for their peers. By creating awareness posters, students use mathematics to organize and present data in a way that encourages digital responsibility.

Standard	**E.IC.3B. Describe the safe versus unsafe uses of computing systems at age-appropriate levels.** **Complexity Statement:** Explain the risks of responding to or chatting with unknown individuals online. **Core Practices:** Fostering cyber responsibility; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** Science & Engineering Practices: Asking questions and defining problems

Integration	Providing information on the risks of responding to or communicating with unknown individuals online helps students recognize safe or unsafe situations. In a science class, invite students to ask questions and define problems. Classroom discussions and student questions on what constitutes a safe, age-appropriate online interaction help build awareness both in and outside of the classroom. Through this questioning process, guide students to collectively define the problem of interacting with unknown individuals online. Possible problem statements include: "How can we identify and avoid potential harm when communicating with people online whose real identities and intentions are unclear?" or "What risks arise when people trust and share information with individuals they only know through the internet?" Discuss how these risks directly relate to the defined problem of uncertainty and potential harm.

Standard	**E.IC.3C. Explain how the school and school system's computing rules and policies keep students safe.** **Complexity Statement:** Describe and model the appropriate behaviors and practices for participating in virtual calls, collaborating with peers, and making public presentations. **Core Practices:** Fostering cyber responsibility; Collaborating around computing; Testing and refining computational artifacts; Communicating about computing
Connections	**Science** Science & Engineering Practices 8: Obtaining, evaluating, and communicating information
Integration	Allowing students to participate in creating online presentations, collaborating with peers, and joining virtual calls as a class with organizations or other classrooms provides opportunities to practice appropriate etiquette when using online platforms and to learn how to connect with others safely. In science, students can participate in a virtual field trip to a marine biology research center. During the visit, they can interact with scientists through a video call, collaborate with students from other schools to record observations, and present their findings to the class by sharing their screen on a smart board.