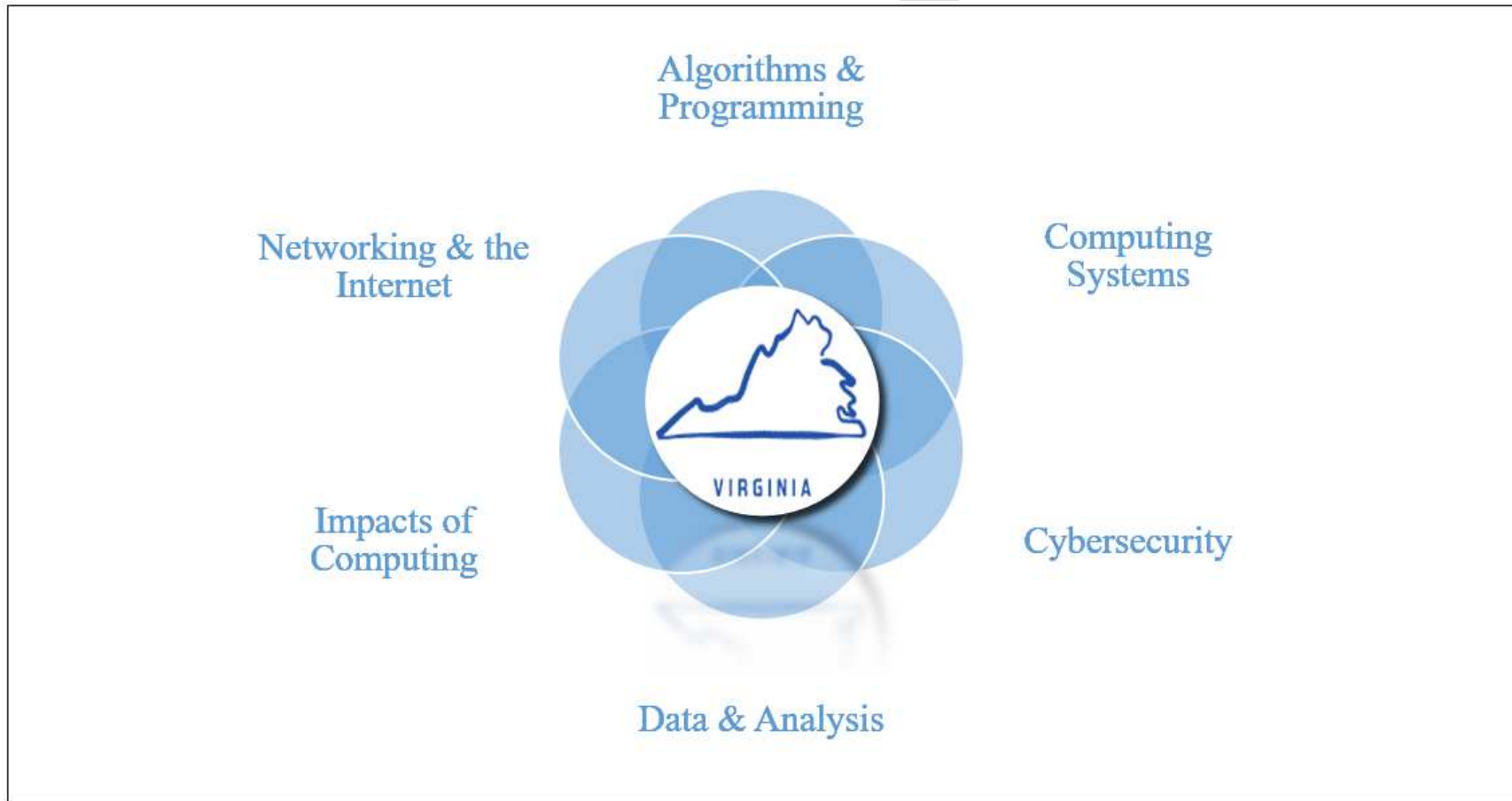


Computer Science Standards of Learning Curriculum Framework



Board of Education
Commonwealth of Virginia

Copyright © 2019
by the
Virginia Department of Education
P.O. Box 2120
Richmond, Virginia 23218-2120
<http://www.doe.virginia.gov>

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

Superintendent of Public Instruction

James F. Lane, Ed.D.

Assistant Superintendent of Learning

Gena Keller

Office of Science, Technology, Engineering, and Mathematics

Tina Manglicmot, Ed.D., Director

Anne Petersen, Ph.D., Science Coordinator

Timothy Ellis, Computer Science Specialist

Joshua Bearman, Science Specialist

NOTICE

The Virginia Department of Education does not discriminate on the basis of race, sex, color, national origin, religion, age, political affiliation, veteran status, or against otherwise qualified persons with disabilities in its programs and activities.

The 2017 *Computer Science Curriculum Framework* can be found on the Virginia Department of Education's [Web site](#).

Introduction

The *Computer Science Standards of Learning* Curriculum Framework amplifies the *Computer Science Standards of Learning for Virginia Public Schools* and defines the content knowledge, skills, and understandings that are measured by the Standards of Learning. The Computer Science Curriculum Framework provides additional guidance to school divisions and their teachers as they develop an instructional program appropriate for their students. It assists teachers as they plan their lessons by identifying essential questions and vocabulary to drive instruction and defining the essential skills students should demonstrate. This supplemental framework delineates in greater specificity the minimum content that all teachers should teach and all students should learn.

School divisions should use the *Computer Science Curriculum Framework* as a resource for developing sound curricular and instructional programs. This framework should not limit the scope of instructional programs. Additional knowledge and skills that can enrich instruction and enhance students' understanding of the content identified in the Standards of Learning should be included as part of quality learning experiences.

Each topic in the *Computer Science Standards of Learning* Curriculum Framework is developed around the Standards of Learning. The format of the Curriculum Framework facilitates teacher planning by broadening the context of the standards and identifying essential student skills that should be the focus of instruction for each standard.

Context of the Standard

The Context of the Standard provides educators an explanation of the standard, including a description and the vertical development of the concept. This context will support teachers in incorporating computer science content into discipline-specific lessons. The intention of the Computer Science standards in grades K-8 is that Computer Science principles be integrated throughout content area instruction.

Essential Skills

The Essential Skills define student performance expectations aligned to each standard. The intent of the K-8 computer science standards is that the concepts are integrated into existing disciplines and this will result in these skills being emphasized differently in each content area. The expectation is that these Essential Skills are partnered with content area performance expectations as appropriate in instruction. At the high school level, the expectations in the 2017 *Computer Science Standards of Learning Curriculum Framework* are to be used in the support of standalone computer courses; the essential skills outlined in the document are not intended to be integrated into other coursework unless a teacher chooses to use the content to support discipline practices.

Essential Questions

Each standard has identified key questions to drive classroom instruction. These questions lead teachers and students toward the big ideas of each concept and provide a more holistic viewpoint used to lead instruction relating to the context of each standard.

Essential Vocabulary

In order to effectively communicate Computer Science concepts, essential vocabulary terms are defined in grade-level appropriate terms. These definitions are found in the glossary (Appendix A).

DRAFT

Kindergarten

The kindergarten standards place emphasis on developing awareness of computing and computing devices by gathering and organizing data, by sorting or step-by-step actions. Students will use accurate terminology to identify components and describe their purposes. Students will also be introduced to communication, security, and responsible computing behaviors. The use of technology will be an integral part of successful acquisition of skills in all content areas.

Algorithms and Programming

K.1 The student will construct sets of step-by-step instructions (algorithms) either independently or collaboratively including sequencing, emphasizing the beginning, middle, and end.

Context of the Standard
At school and at home, students engage in step-by-step activities on a routine basis. These may include such activities as brushing their teeth or preparing to leave school at the end of the school day. When students document these step-by-step instructions they are creating algorithms. As students learn to construct simple algorithms that reflect a daily activity, they realize that the sequences use have a beginning, middle, and end. Just as people use algorithms to complete daily routines, they can program computers to use algorithms to complete different tasks. Algorithms are commonly implemented using a precise language that computers can interpret.

Essential Skills	Essential Questions	Essential Vocabulary
-------------------------	----------------------------	-----------------------------

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify daily activities that are completed using a sequence of steps. - Describe the steps taken to accomplish an activity. - As a class, construct a sequence of steps to accomplish an activity. - Identify the beginning, middle and end of a sequence of steps.	Students should <i>investigate</i> these concepts: - What are examples of activities that are part of a daily routine? - What are the steps of an activity that you perform frequently? - How does having a defined sequence of steps make an activity easier?	Students should be <i>introduced</i> to these concepts: Sequence

K.2 The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, either independently or collaboratively, including sequencing, emphasizing the beginning, middle, and end.

Context of the Standard
When an algorithm or a set of algorithms is tested, a program has been created. People work together to plan, create and test these programs. This process of planning, creating, and testing program is called programming and is used to create a wide variety of products such as video games, interactive art projects and digital stories. The sequencing of tasks and the testing of programs can be done with or without the use of a computer. When it is done without a computer it is considered an unplugged activity. Unplugged activities allow students to understand the sequencing and testing process in a concrete way. Unplugged activities can be found at sites such as CS Unplugged. Programs can also be developed and tested using computer applications; these are referred to as plugged activities. Students may use block based programs to sequence steps as they develop simple computer programs.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - As a class, construct a sequence of steps to accomplish an activity (unplugged). - Recognize that a sequence of steps when using a computer is called a program. - As a class or individually, use a block based programming language (e.g., Scratch Jr.) or unplugged activity to complete a simple task as a form of creative expression. - Model the steps of a program using coding cards or similar instructional strategy.	Students should <i>investigate</i> these concepts: - How would you describe a series of steps that you do every day to complete an activity (brushing your teeth, putting on a shirt, etc)? - Why is it important that some tasks have a beginning, middle, and end? - What would happen if a set of instructions was begun in the middle, rather than at the beginning?	Students should be <i>introduced</i> to these concepts: - Algorithm - Program - Sequence

K.3 The student will create a design document to illustrate thoughts, ideas, and stories in a sequential (step-by-step) manner (e.g., story map, storyboard, and sequential graphic organizer).

Context of the Standard
As students listen to stories, they realize that these stories have a beginning, middle, and end. Authors use multiple ways of creating a story; stories can be told using story maps, storyboards, or graphic organizers. Just as authors sequence events in a story, people may use computers as they work together to plan, create, and test programs. Programming is used as a tool to create products that reflect a wide range of interests, such as video games, interactive art projects, and digital stories.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Design a sequence using a visual tool such as storyboard. - Use illustrations to tell a story that has a beginning, middle, and end. - Using a graphical representation (graphic organizer, storyboard, or story map) retell a story by arranging the events in the correct sequence.	Students should <i>investigate</i> these concepts: - What tools can an author/comic book artist/movie director use to write a story? - Why would authors want different ways to organize a story?	Students should be <i>introduced</i> to these concepts: - Design document - Storyboard - Graphic organizer

K.4 The student will categorize a group of items based on one attribute or the action of each item, with or without a computing device.

Context of the Standard		
Objects and actions have attributes; these attributes allow people to group items. Attributes may be physical properties such as color, shape or form, texture, and size. Actions may be categorized as movement such as those seen in step by step sequences (algorithms). Actions may be categorized as back and forth movement, turning, and stopping. Categorizing relies on careful observation of patterns and similarities and differences. These similarities and differences can be used to categorize using both unplugged activities and computer devices.		
Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify the attributes of an object (e.g., color, size, shape, thickness, actions). - Sort and classify objects into appropriate groups (categories) based on one attribute.	Students should <i>investigate</i> these concepts: - How do people describe different cars/animals/movements/etc.? - How do people organize different objects into categories? - How would you compare and contrast two basic actions (taking a step vs turning your head, for example)?	Students should be <i>introduced</i> to these concepts: - Pattern Matching - Program

Computing Systems

K.5 The student will identify components of computing systems (e.g., keyboard, mouse, desktop computer, laptop computer, tablet, and printer).

Context of the Standard
Computing systems are composed of different components. These components enable the user to complete different tasks using a computing system. Common components among desktop and laptop computers, tablets, and mobile phones include a keyboard (either physical or screen based) for inputting information, audio for hearing information, and a screen for viewing information.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify the keyboard, mouse/trackpad, and printer. • Describe the purpose of the keyboard, mouse/trackpad, and printer. • Demonstrate proper use of a mouse/trackpad and keyboard. • Identify different computing systems such as a desktop computer, laptop computer, tablet, and mobile device.	Students should <i>investigate</i> these concepts: • How does using a mouse or keyboard make computing easier? • What are different computing devices used at home or in school?	Students should be <i>introduced</i> to these components: • Keyboard • Mouse • Trackpad • Desktop computer • Laptop computer • Tablet • Printer

K.6 The student will identify, using accurate terminology, simple hardware and software problems that may occur during use (e.g., app or program is not working as expected, no sound is coming from a device, device will not turn on).

Context of the Standard
Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution. In kindergarten, students are expected to clearly describe a problem in the computing system using developmentally appropriate language. Examples include, “The computer won’t turn on,” “The pointer on the screen won’t move,” or “I lost the web page.” <i>Students are not expected to diagnose or troubleshoot a problem with a computing system in Kindergarten.</i>

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify when a device or program is not working properly. <i>Students are not expected to differentiate between software and hardware.</i> - Communicate that a device or program is not working.	Students should <i>investigate</i> these concepts: - How can you tell if your device is not working properly? - What are ways you can describe the problem to your teacher?	Students should be <i>introduced</i> to these concepts: - Computing device - Keyboard - Mouse - Printer

Cybersecurity

K.7 The student will identify what is allowed and what is not allowed at school when using technology.

Context of the Standard
Computer networks, including the Internet, can be used to connect people to other people, places, information, and ideas. In order to keep students safe, schools and divisions have rules on the appropriate use of technology. All students should be aware of what is allowed and not allowed when using division/school technology.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Classify computer actions as allowed or not allowed based on school rules. - Communicate process for reporting inappropriate use of technology. - Demonstrate proper care for electronic devices (e.g., handling, logging off or shutting down correctly, and keeping devices away from water/food).	Students should <i>investigate</i> these concepts: - What are your school's rules about computer use? - Why does the school have rules that students and teachers need to follow? - Why are the rules at school different (maybe) than those that you have at your house? - How could improper use of a computer be harmful to someone else? - How could improper use of a computer be harmful to you?	Students should be <i>introduced</i> to these concepts: - Digital Citizenship - Computing Device - Digital Safety

K.8 The student will identify personal information (e.g., address, telephone number, and name) and the importance of protecting personal information online.

Context of the Standard
Connecting devices to a network or the Internet provides great benefit, but care must be taken to ensure that personal information is not shared with others. Personal information includes a student's name, address, and phone number. When using a computing system, personal information such as computer passwords are not to be shared with others in the classroom. Students and teachers should never share personal information with anyone on the Internet. Privacy should be considered when posting information online; such information can persist for a long time and be accessed by others, even unintended viewers.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain why they should not share personal information. • Explain the importance of passwords.	Students should <i>investigate</i> these concepts: • What are some examples of information that should not be shared with strangers? • What are some ways people can protect information on their computer?	Students should be <i>introduced</i> to these concepts: • Digital Citizenship • Digital Safety • Personal information

Data and Analysis

K.9 The student will gather and display data and organize it in a chart or graph in order to answer questions about the data, with or without a computing device.

Context of the Standard
Students use their senses to make observations and to collect data about the world around them. Data are pieces of information collected about people or things. These data can be recorded in tables and can be used to construct object graphs or picture graphs. Everyday digital devices can also be used to collect and display data over time. Cell phones, digital toys, and cars can contain tools (such as sensors) and computers to collect and display data from their surroundings.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Collect data on categories identified by the teacher and/or student. • Represent gathered data in tables (vertically or horizontally). • Represent data by arranging concrete objects into organized groups to form a simple object graph. • Be able to use data to answer questions. • Represent gathered data, using pictures to form a simple picture graph.	Students should <i>investigate</i> these concepts: • Why do we collect data? • What are examples of data? • What are ways that we can arrange data? • What questions can be answered with a set of collected data?	Students should be <i>introduced</i> to these concepts: • Data • Table • Object graph • Picture graph

Impacts of Computing

K.10 The student will identify responsible behaviors associated with using information and technology.

Context of the Standard
Using computers comes with a level of responsibility, such as not sharing login information, keeping passwords private, and logging off when finished. These behaviors apply regardless of where a student is at school or on a computer at another location. In addition to keeping information private, responsible behaviors should be exhibited when engaging in online communications. Online communication facilitates positive interactions, such as sharing ideas with many people, but the public and anonymous nature of online communication also allows intimidating and inappropriate behavior in the form of cyberbullying. <i>Students are not responsible for the term or description of cyberbullying in Kindergarten.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Interact responsibly with peers when using technology. - Describe what information should be shared and not shared. - Describe online behaviors that may be harmful to others.	Students should <i>investigate</i> these concepts: - What are behaviors used in school to keep people safe online? - How are responsible computer behaviors important to good citizenship?	Students should be <i>introduced</i> to these concepts: - Digital Citizenship - Computing Device - Digital Safety

Networking and the Internet

K.11 The student will discuss, in a whole class setting, how information can be communicated electronically (e.g., email, social media).

Context of the Standard
Communication channels have increased with the use of electronic devices. Devices such as cell phones and computers, allow people to communicate through email, texts, video calling, and social media. With many means of sharing information, computer safety and the sharing of personal information need to be reviewed and reinforced on a regular basis.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills:	Students should <i>investigate</i> these concepts:	Students should be <i>introduced</i> to these concepts:

Essential Skills	Essential Questions	Essential Vocabulary
• Understand how information can be communicated electronically. • Describe different types of electronic communication.	• What are different ways people communicate with each other? • What are different ways people can communicate with a computer?	

Grade One

The first grade standards place emphasis on developing organizational skills, such as classifying based on common attributes, completing a pattern, or explaining step-by-step processes. Students will use accurate terminology to identify components and describe their purposes. Students will also be able to describe to communication, security, and responsible computing behaviors. The use of technology will be an integral part of successful acquisition of skills in all content areas.

Algorithms and Programming

- 1.1 The student will construct sets of step-by-step instructions (algorithms) either independently or collaboratively, including
- sequencing (including ordinal numbers) and;
 - simple loops (patterns and repetition).

Context of the Standard

At school and at home, students engage in step-by-step activities on a routine basis. These may include such activities as brushing their teeth or preparing to leave school at the end of the school day. When students document these step-by-step instructions they are creating algorithms. Sometimes there are repeating steps in a task, and students can create a loop in their algorithm to indicate that repeating pattern. As an example, walking forward 10 steps and turning could be repeated 4 times to illustrate a square. A loop allows the core of the pattern to be written once instead of 4 times, shortening the overall length of the algorithm.

Algorithms can be created with or without computers. Computers follow precise sequences of instructions that automate tasks. A precise sequence of instructions that a computer uses is referred to as a program. Programs are sequential and may contain loops. Unplugged activities can be found at sites such as [CS Unplugged](#).

Essential Skills	Essential Questions	Essential Vocabulary
------------------	---------------------	----------------------

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Individually and/or as a class, construct a sequence of steps to accomplish an activity. - Identify a section of repeated actions to replace with a loop. - Given a sequence of steps that include a loop, predict the next step in the sequence.	Students should <i>investigate</i> these concepts: - When would a loop be used when describing a sequence of steps? - What is an example of a daily task that includes a loop? (Given a pattern of numbers or images): How can you predict what should come next in this series?	Students should be <i>introduced</i> to these concepts: - Loop - Pattern matching - Repeat - Sequence

- 1.2 The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, either independently or collaboratively including
- sequencing, ordinal numbers; and
 - simple loops (patterns and repetition).

Context of the Standard
When an algorithm or a set of algorithms is tested, a program has been created. People work together to plan, create and test these programs. This process of planning, creating, and testing program is called programming and is used to create a wide variety of products such as video games, interactive art projects and digital stories. In first grade, students are expected to develop and test simple algorithms that include both sequencing and simple loops to complete a task. Block based programs (i.e. Scratch Jr., Tynker) allow students to develop simple algorithms using a computer. Students can also create simple unplugged programs that don't require a computer but contain sequencing and loops through the use of coding cards, mazes, and other activities that provide students opportunities to describe tasks as a sequence of events.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • As a class, construct a sequence of steps to accomplish an activity (unplugged). • Recognize that a sequence of steps when using a computer is called a program. • Recognize a repeated sequence of steps as an opportunity to use a loop. • As a class or individually, use a block based programming language (e.g., Scratch Jr.) or unplugged activity to complete a simple task as a form of creative expression. • Model the steps of a program that contains at least one loop using coding cards or similar instructional strategy.	Students should <i>investigate</i> these concepts: • How can you write a series of steps to complete an action? • When is it useful to use a loop?	Students should be <i>introduced</i> to these concepts: • Loop • Program • Block-based programming language

1.3 The student will analyze, correct, and improve (debug) an algorithm that includes sequencing.

Context of the Standard
The practice of reviewing work should be taught early and can be applied across disciplines, including computer science. Students should check that the sequence of steps that compose an algorithm works as intended. That is the only way to determine if the algorithm appropriately reflects the steps that must occur to complete a task. This process can be conducted for both computer programs and unplugged activities. If the algorithm does not work as intended, the students should determine the changes to make in the algorithm in order to complete the task. These changes may include adding, deleting, rearranging, or changing a step in order

Context of the Standard
to obtain the intended outcome. The process of revising a program so that it works as intended is called debugging.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Understand that algorithms and/or programs may not always work correctly. - Describe how an algorithm does not work (e.g., character is not moving as intended). - As a class, analyze a simple sequence of steps that is flawed and determine possible solution(s).	Students should <i>investigate</i> these concepts: - If something is not working, how do you figure out what is broken? - What does “bug” mean when we are talking an algorithm? - How can you tell if a program or sequence has a bug? - What are the steps to take when you are trying to locate a bug in a program or sequence?	Students should be <i>introduced</i> to these concepts: - Debug - Program

- 1.4 The student will plan and create a design document to illustrate thoughts, ideas, and stories in a sequential (step-by-step) manner (e.g., story map, storyboard, sequential graphic organizer).

Context of the Standard
As students listen to stories, they realize that these stories have a beginning, middle, and end. Planning a story is similar to writing an algorithm or program in that there are steps that are followed as the author determines the beginning, middle and end of the story. Authors use multiple ways of planning and telling a story; stories can be told using story maps, storyboards, or graphic organizers. This standard is intended to develop a fundamental understanding that programs can be developed using similar strategies as stories.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Design a sequence using a visual tool such as storyboard. - Use illustrations to tell a story that has a beginning, middle, and end. - Using a graphical representation (graphic organizer, storyboard, or story map) to retell a story by arranging the events in the correct sequence.	Students should <i>investigate</i> these concepts: - Why should you plan a program before beginning to write? - What tools can a programmer use to plan?	Students should be <i>introduced</i> to these concepts:

1.5 The student will categorize a group of items based on one or two attributes or the actions of each item, with or without a computing device.

Context of the Standard
Objects and actions have attributes; these attributes allow people to group items. Attributes may be physical properties such as color, shape, form, texture, and size. Actions may be categorized as movement such as those seen in step-by-step sequences (algorithms). Actions include back and forth movement, turning, and stopping. Categorizing relies on careful observation of patterns and similarities and differences. These similarities and differences can be used to categorize using both unplugged activities and computer devices. In this standard, students are expected to group items on one or two attributes or actions. In the science and mathematics standards the term classify is used when grouping objects or organisms based on one or more attributes. In block based programming environments, commands are grouped into categories based on function. In higher level programming languages, the kind or type of data determines the classification.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Sort and group (classify) objects into appropriate groups (categories) based on one or two attributes. - Label attributes of a set of objects that has been sorted. - Name multiple ways to sort a set of objects.	Students should <i>investigate</i> these concepts: - Why is it useful to organize objects into categories? (Given a set of objects/drawings): What are ways to group these objects or actions? What attributes did you use to group these objects or actions? (Given a set of objects/drawings): What attributes do you think a person used to group these objects? What is another way you can group these objects? (Given assorted objects) How would you group these items using two different attributes?	Students should be <i>introduced</i> to these concepts: Attribute

1.6 The student will acknowledge that materials are created by others (e.g., author, illustrator).

Context of the Standard
As students start to work with different artifacts (reference materials, resources, etc.) they should understand that these sources of information were created by others. Authors, illustrators, and programmers are responsible for the creation of many sources of information that we use in our classroom and in our homes. This standard begins an exploration of the concepts of intellectual property laws and plagiarism. <i>Students are not responsible for these terms in first grade.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain that artifacts have owners. • State whether an artifact was created by the student or someone else. • Identify when to credit others work when using their resources. • Identify authors as needed in class projects (individually and as a class) either in writing or orally.	Students should <i>investigate</i> these concepts: • How can you tell who wrote a book or painted a picture? • What does it mean to give credit? • Why is important to give someone credit for their work? • What is the difference between using someone's idea and copying their idea?	Students should be <i>introduced</i> to these concepts: • Author • Artifact • Illustrator

Computing Systems

- 1.7 The student will identify components of computing systems that are common among different types of computing devices including desktop and laptop computers, tablets, and mobile phones.

Context of the Standard

Computing systems are composed of different components. These components enable the user to complete different tasks using a computing system.

Common components among desktop and laptop computers, tablets, and mobile phones include a keyboard (either physical or screen based) for inputting information, audio for hearing information, and a screen for viewing information.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify different computing systems such as a desktop computer, laptop computer, tablet, and mobile device. • Identify the keyboard, mouse/trackpad, and printer. • Describe the purpose of the keyboard, mouse/trackpad, and printer. • Compare and contrast the components used by different computing systems (desktop, laptop, tablet, mobile device). • Demonstrate proper use of a mouse/trackpad and keyboard.	Students should <i>investigate</i> these concepts: • What are the components that are shared between desktops, laptops, and tablets? • What are the components that are different between desktops, laptops, and tablets? • How do the different components (keyboard, mouse, printer, etc.) of a computing system help you accomplish a given task?	Students should be <i>introduced</i> to these concepts: • Desktop Computer • Keyboard • Laptop Computer • Mobile Phone • Mouse • Printer • Tablet • Touch Screen

- 1.8 The student will identify, using accurate terminology, simple hardware and software problems that may occur during use (e.g., app or program is not working as expected, no sound is coming from the device, the device won't turn on).

Context of the Standard
Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution. Problems with computing systems have different causes, such as hardware settings, programming errors, or faulty connections to other devices. Developmentally appropriate ways to solve problems include debugging simple programs and seeking help by clearly describing a problem (for example, “The computer won’t turn on,” “The pointer on the screen won’t move,” or “I lost the web page.”). <i>Students are not expected to diagnose or troubleshoot a problem with a computing system in first grade. Students are not expected to differentiate between software and hardware in first grade.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify when a device or program is not working properly. • Communicate that a device or program is not working.	Students should <i>investigate</i> these concepts: • How can you tell if your device is not working properly? • What are ways you can describe the problem to your teacher?	Students should be <i>introduced</i> to these concepts: • Keyboard • Mouse • Desktop computer • Laptop computer • Tablet • Printer

Cybersecurity

1.9 The student will describe what is allowed and what is not allowed at school associated with the use of technology.

Context of the Standard

Computer networks, including the Internet, can be used to connect people to other people, places, information, and ideas. In order to keep students safe, schools and divisions have rules on the appropriate use of technology. All students should be aware of what is allowed and not allowed when using division/school technology.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Classify computer actions as allowed or not allowed based on school rules. - Communicate the process for reporting inappropriate use of technology. - Demonstrate proper care for electronic devices (e.g., handling, logging off or shutting down correctly, and keeping devices away from water/food).	Students should <i>investigate</i> these concepts: - What are the school rules for technology use? - Why do you think the school has rules for technology? - If you were in charge, what rules would you make for technology use and why?	Students should be <i>introduced</i> to these concepts: Digital Safety

1.10 The student will identify and use strong passwords, explain why strong passwords should be used (e.g., protect name, address, and telephone number).

Context of the Standard

Context of the Standard
Connecting devices to a network or the Internet provides great benefit, but care must be taken to protect private information such as a student’s name, phone number, and address. Passwords are used to protect devices and information from unauthorized access. Strong passwords have characteristics that make them more difficult to guess. Many sites have rules as to the length and composition of passwords; these rules help create stronger passwords. The practice of not sharing passwords should be emphasized in the classroom and at home.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain how a password helps protect the privacy of information. • Refrain from using other students' passwords. • Explain how logging off devices can protect personal information.	Students should <i>investigate</i> these concepts: • Why do we use passwords? • What makes a password strong? • What kind of information would you want to protect with a password? • Why is it important not to share passwords?	Students should be <i>introduced</i> to these concepts: • Digital Safety • Personal information

Data and Analysis

- 1.11 The student will identify and interpret data and organize it in a chart or graph in order to make a prediction, with or without a computing device.

Context of the Standard

The collection and use of data about individuals and the world around them is a routine part of life and influences how people live. Data are pieces of information collected about people or things. These data can be recorded in tables and can be used to construct object graphs or picture graphs. Everyday digital devices can be used to collect and display data over time. Cell phones, digital toys, and cars can contain tools (such as sensors) and computers to collect and display data from their surroundings.

Once data has been collected and organized into a chart or graph, it can be analyzed to determine if a pattern exists. The pattern can be used to make predictions or answer questions.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify different types of data that can be collected. • Collect data on categories identified by the teacher and/or student. • Represent gathered data in tables (vertically or horizontally). • Represent data by arranging concrete objects into organized groups to form a simple object graph. • Represent gathered data, using pictures to form a simple picture graph.	Students should <i>investigate</i> these concepts: • What are examples of different data we can collect? • What are ways that we can arrange data? • What are the different types of graphs we can use to represent data? • (Give a set of data) What predictions can you make based on the data you have?	Students should be <i>introduced</i> to these concepts: • Data • Prediction

Essential Skills	Essential Questions	Essential Vocabulary
Analyze data and identify patterns that can be used to make predictions.		

Impacts of Computing

1.12 The student will identify and explain responsible behaviors associated with using information and technology.

Context of the Standard
Using computers comes with a level of responsibility, such as not sharing login information, keeping passwords private, and logging off a computer device when finished with a task. These behaviors apply regardless of whether a student is at school or on a computer at another location. In addition to keeping information private, responsible behaviors should be exhibited when engaging in online communications. Online communication facilitates positive interactions, such as sharing ideas with many people, but the public and anonymous nature of online communication also allows for intimidating and inappropriate behavior in the form of cyberbullying. Cyberbullying is a form of bullying that occurs when online communications are sent that are intimidating or threatening in nature. <i>Students are not responsible for the term cyberbullying in first grade.</i>

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Interact responsibly with peers when using technology. • Describe what information should be shared and not shared with strangers. • Describe online behaviors that may be harmful to others.	Students should <i>investigate</i> these concepts: • What are examples of responsible online behavior? • What information is acceptable to share online? • What are behaviors that should be avoided when interacting with others online? • What should you do if a person sends you information that is intimidating, threatening, or that makes you feel badly?	Students should be <i>introduced</i> to these concepts: • Digital Citizenship • Digital Safety • Personal information

Networking and the Internet

- 1.13 The student will, in a whole class environment, discuss how information can be communicated electronically (e.g., email, social media).

Context of the Standard
Online communication facilitates positive interactions, such as sharing ideas with many people, including friends and family around the world. It also allows opportunities for scientists, mathematicians, business people, and many other professionals to communicate about projects they are working on together. People with similar interests can meet through social media or email and share information; however, the public and anonymous nature of online communication also allows intimidating and inappropriate behavior in the form of cyberbullying. When using social media or email with strangers, do not share personal information such as phone number or address. Never share pictures of yourself with others you do not know using social media or email.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Understand that information can be communicated electronically. • Describe different types of electronic communication.	Students should <i>investigate</i> these concepts: • What are different ways people communicate with each other? • What are different ways people can communicate with a computer?	Students should be <i>introduced</i> to these concepts: • Email • Social media

Grade Two

The standards for second grade place an emphasis on creating models of physical objects or processes in order to demonstrate relationships. Second grade standards build on students' skills in constructing programs and utilizing algorithms. The accurate use of terminology as well as the responsible use of technology will continue to be built upon. The foundational understanding of computing and the use of technology will be an integral component of successful acquisition of skills across content areas.

Algorithms and Programming

- 2.1 The student will construct sets of step-by-step instructions (algorithms) both independently and collaboratively
- using sequencing;
 - using loops (a wide variety of patterns such as repeating patterns or growing patterns); and
 - identifying events.

Context of the Standard

At school and at home, students engage in step-by-step activities on a routine basis. These may include such activities as brushing their teeth or preparing to leave school at the end of the school day. When students document these step-by-step instructions they are creating algorithms. Sometimes there are repeating steps in a task, and students can create a loop in their algorithm to indicate that repeating pattern. Algorithms can be created with or without computers. In first grade the students are introduced to the use of loops, in second grade this understanding expands to include repeating patterns and growing patterns.

In second grade, the construction of loops becomes more complex as students use a wide variety of patterns to include repeating and growing patterns. In a repeating pattern the units of the pattern repeat and remain the same. In a growing pattern, an addition is added to the pattern causing the pattern to change every time it repeats. Growing patterns involve a progression from step to step which make them more difficult for students than repeating patterns.

Repeating and growing patterns are foundational in mathematics in the development of algebraic reasoning and in computer science in developing computational thinking.

Sample numeric patterns include:

- 6, 9, 12, 15, 18,....(growing pattern);
- 1, 2, 4, 7, 11, 16,....(growing pattern);
- 20, 18, 16, 14,.....(growing pattern); and

Context of the Standard
1, 3, 5, 1, 3, 5, 1, 3, 5,.....(repeating pattern). In second grade, students are also expected to identify events. In computer science, an event is an action or occurrence detected by a program. Events can be user actions, such as clicking a mouse button or pressing a key, or system occurrences, such as a timer or low battery.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe the steps taken to accomplish an activity using both sequences and loops. - Identify a section of repeated actions to replace with a loop. - Given a sequence of steps that include a loop, predict the next step in the sequence. - Given a sequence of steps to accomplish a daily task, give a possible event that triggers these steps. - Compare and contrast a repeating pattern and a growing pattern.	Students should <i>investigate</i> these concepts: - What are examples of tasks where a loop would be appropriate? - What are examples of events in a plugged or an unplugged activity? - What is the difference between a repeating and a growing pattern? - How can repeating and growing patterns be represented in a sequence?	Students should be <i>introduced</i> to these concepts: - Loop - Event

- 2.2 The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively
- using sequencing;
 - using loops (a wide variety of patterns, such as repeating patterns or growing patterns); and
 - identifying events.

Context of the Standard
When an algorithm or a set of algorithms is tested, a program has been created. People work together to plan, create and test these programs. This process of planning, creating, and testing program is called programming and is used to create a wide variety of products such as video games, interactive art projects and digital stories. In second grade, students are expected to develop simple programs that use both sequencing and simple loops to complete a task. These programs may be developed using block-based or unplugged activities. Block-based programs (e.g., Scratch Jr., Tynker) allow students to develop simple algorithms using a computer. Students are also expected to identify events; an event is an action or occurrence detected by a program.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - As a class and individually, construct a sequence of steps to accomplish an activity. - Recognize that a sequence of steps when using a computer is called a program. - Recognize a repeated sequence of steps as an opportunity to use a loop. - Model the steps of a program that	Students should <i>investigate</i> these concepts: - How would you write instructions for an action that repeats itself? - What are different ways that you can signal the start of a program: plugged and unplugged? - What does the word “event” mean in the context of programming?	Students should be <i>introduced</i> to these concepts: - Event - Loop - Program - Repeat

Essential Skills	Essential Questions	Essential Vocabulary
contains at least one loop using coding cards or similar instructional strategy. - Identify events that are used in a program. - Explain the role of an event in a program.		

- 2.3 The student will analyze, correct, and improve (debug) an algorithm that includes sequencing and simple loops, with or without a computing device.

Context of the Standard
The practice of reviewing work should be taught early and can be applied across disciplines, including computer science. Students should check that the sequence of steps that compose an algorithm works as intended. That is the only way to determine if the algorithm appropriately reflects the steps that must occur to complete a task. This process can be conducted for both computer programs and unplugged activities. If the algorithm does not work as intended, the students should determine the changes to make in the algorithm in order to complete the task. These changes may include adding, deleting, rearranging, or changing a step in order to obtain the intended outcome. The process of revising a program so that it works as intended is called debugging.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Understand that a sequence and/or program may not always work correctly.	Students should <i>investigate</i> these concepts: - How do you identify an error in a set of instructions? - Once you have found an error in your	Students should be <i>introduced</i> to these concepts: Debug

Essential Skills	Essential Questions	Essential Vocabulary
- Describe how an algorithm did not work (e.g., character is not moving as intended). - Analyze a simple sequence of steps that is flawed and determine possible solution(s). - As a class, implement a proposed adjustment to a sequence that did not work as intended.	instructions, how do you decide what adjustment needs to be made to the sequence? How can you accomplish a goal with fewer steps?	

2.4 The student will plan and create a design document to illustrate thoughts, ideas, and stories in a sequential (step-by-step) manner (e.g., story map, storyboard, sequential graphic organizer).

Context of the Standard
As students write and tell stories, they use tools such as story maps, storyboards, and sequential graphic organizers to plan and describe events. In second grade, planning stories becomes more complex as they begin to add descriptive details to their stories. Planning a story is similar to writing an algorithm or program in that there are steps that are followed as the author determines the beginning, middle and end of the story as well as adds details to accomplish a task. Just as there are multiple ways to tell a story, different algorithms can be used to describe a task. While the end results may be similar, the actual pathway to achieve the task may not be the same. This standard is intended to develop a fundamental understanding that programs can be developed using similar strategies as stories.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Plan and create a design document to tell a story. - Using a graphical representation (graphic organizer, storyboard, or story map) to retell a story by arranging the events in the correct sequence. - Participate in teacher-directed planning and writing strategies to organize ideas and information into a story and/or program.	Students should <i>investigate</i> these concepts: - How is a program like a story? - What are ways that you plan and write a story and/or program? - How can you identify the beginning, middle, and end of a story?	Students should be <i>introduced</i> to these concepts: - Storyboard - Graphic organizer

2.5 The student will compare and contrast a group of items based on the attributes or actions of each item, with or without a computing device.

Context of the Standard
Objects and actions have attributes; these attributes allow people to group items. Attributes may be physical properties, behaviors, or actions. Actions in computer science are reflected in step-by-step sequences (algorithms). Actions may be categorized as back and forth movement, turning, and stopping. Categorizing of attributes or actions relies on careful observation of patterns and similarities and differences. In this standard, students are expected to analyze groups of items and compare and contrast the attributes that led to the development of the group. In block based programming environments, commands are grouped into categories based on function. In higher level programming languages, data are often classified by the type and format of the information.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Sort and group (classify) objects into appropriate groups (categories) based on multiple attributes. - Label attributes of a set of objects that has been sorted. - Name multiple ways to sort a set of objects.	Students should <i>investigate</i> these concepts: - How are items organized with multiple attributes? - Why is it useful to sort objects into groups? - Why is sorting objects into groups helpful in our daily lives?	Students should be <i>introduced</i> to these concepts: Attribute

2.6 The student will acknowledge that materials are created by others (e.g., author, illustrator, and website).

Context of the Standard
As students start to work with different artifacts (reference materials, resources, etc.) they should understand that these sources of information were created by others. Authors, illustrators, and programmers are responsible for the creation of many sources of information that are used in the classroom and at home. This standard begins an exploration of the concepts of intellectual property laws and plagiarism. <i>Students are not responsible for the terms property laws and plagiarism in second grade.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Explain that artifacts have owners.	Students should <i>investigate</i> these concepts: How can you find the author of a	Students should be <i>introduced</i> to these concepts:

Essential Skills	Essential Questions	Essential Vocabulary
- State whether an artifact is created by the student or someone else. - Identify when to credit others work when using their resources. - Identify authors as needed in class projects (individually and as a class) either in writing or orally.	website or book? - What is the difference between creating something on your own and changing someone else's work? - How can you give credit when you are using other people's ideas/work?	- Author - Digital Artifact - Illustrator - Website

Computing Systems

2.7 The student will describe the characteristics of computing systems to include hardware, software, input, and output.

Context of the Standard
A system is defined as a regularly interacting or interdependent group of items forming a unified whole. A computing system is composed of hardware and software. Hardware consists of physical components, while software provides instructions for the system. These instructions are represented in a form that a computer can understand and allow the user to input information and, once the task is completed, obtain output in a form that can be understood. Hardware and software work together as a system to accomplish tasks, such as sending, receiving, processing, and storing units of information. Hardware include screens to display information and buttons, keys, or touch screens to enter information. Software applications are programs with specific purposes, such as a web browser or game. A person may use a mouse (hardware) to click on a button displayed in a web browser (software) to navigate to a new web page.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills:	Students should <i>investigate</i> these concepts:	Students should be <i>introduced</i> to

Essential Skills	Essential Questions	Essential Vocabulary
• Compare and contrast hardware and software. • Describe how both hardware and software are used in a computing system. • Identify the input and output associated with a given task.	• What are the differences between hardware and software? • What are examples of hardware and software? • What are examples of input and output? • If you were a computer, what would be an example of input and output? ○ What is an example of your teacher providing you with input? ○ What is an example of you providing your teacher with output?	these concepts: • Hardware • Software • Input • Output

2.8 The will identify, using accurate terminology, simple hardware and software problems that may occur during use (e.g., app or program not working as expected, no sound, device won't turn on).

Context of the Standard
Although computing systems may vary, common troubleshooting strategies can be used on them, such as checking connections and power or swapping a working part in place of a potentially defective part. Rebooting a machine is commonly effective because it resets the computer. Since computing devices are composed of an interconnected system of hardware and software, troubleshooting strategies may need to address both. Students in second grade are expected to use accurate terminology to describe simple problems with computer hardware and software.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify when a device or program is not working properly. - Communicate that a device or program is not working. - Perform simple troubleshooting tasks (e.g., rebooting the computer).	Students should <i>investigate</i> these concepts: - How can you tell a computer is not working as intended? - What is the first thing you should try if a program is not working? - Why is it important to be as specific as possible when you are describing a problem?	Students should be <i>introduced</i> to these concepts: - Reboot - Troubleshoot

Cybersecurity

2.9 The student will explain what is allowed and what is not allowed at school associated with the use of technology (e.g., class rules).

Context of the Standard
Computer networks, including the Internet, can be used to connect people to other people, places, information, and ideas. In order to keep students safe, schools and divisions have rules on the appropriate use of technology. All students should be aware of what is allowed and not allowed when using division/school technology. Appropriate use of technology as well as school and division rules when using technology should be reviewed with students on a regular basis. Consistent monitoring of students when engaged with technology should be conducted at all times.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Classify computer actions as allowed or not allowed based on school rules. - Communicate the process for reporting inappropriate use of technology. - Demonstrate proper care for electronic devices (e.g., handling, logging off or shutting down correctly, and keeping devices away from water/food).	Students should <i>investigate</i> these concepts: - What is appropriate use of technology in school? - If you see someone using technology inappropriately in school, how should you notify the proper person?	Students should be <i>introduced</i> to these concepts:

- 2.10 The student will identify and create strong passwords, explain why strong passwords should be used. (e.g., protect name, address, and telephone number).

Context of the Standard
Connecting devices to a network or the Internet provides great benefit, but care must be taken to protect private information such as a student's name, phone number, and address. Passwords are used to protect devices and information from unauthorized access. Because computer programs can be used to guess passwords, strong passwords have characteristics that make them more difficult to guess. Many sites have rules as to the length and composition of passwords; these rules help create stronger passwords. The practice of not sharing passwords should be emphasized in the classroom and at home. Suggestions for creating strong passwords include: 1. Use uppercase and lowercase letters. 2. Use numbers. 3. Use symbols.

Context of the Standard
4. Use at least 8 characters. 5. Don't use words from a dictionary. 6. Don't use the same password twice. 7. Don't use personal information. <i>Students are not expected to list these suggestions; however, these suggestions may be introduced when students are allowed to create or classify passwords.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain how a password helps protect the privacy of information. • Refrain from using other students' password. • Explain how logging off devices can protect your information. • Classify passwords as strong or weak.	Students should <i>investigate</i> these concepts: • What are the components of a strong password? • Why are strong passwords needed when using a computer?	Students should be <i>introduced</i> to these concepts: • Password

Data and Analysis

- 2.11 The student will construct and analyze data and organize it in a chart or graph in order to make a prediction, with or without a computing device.

Context of the Standard
The collection and use of data about individuals and the world around them is a routine part of life and influences how people live.

Context of the Standard
Data are pieces of information collected about people or things. These data can be recorded in tables and can be used to construct pictographs or bar graphs. Everyday digital devices can be used to collect and display data over time. Examples include cell phones, digital toys, and cars. These can contain tools (such as sensors) and computers to collect and display data from their surroundings. Once data has been collected and organized into a chart or graph, it can be analyzed to determine if a pattern exists. The pattern can be used to make predictions or answer questions.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Represent gathered data in tables (vertically or horizontally). - Represent data using pictographs or bar graphs.	Students should <i>investigate</i> these concepts: - What are examples of data that we can collect in the classroom or schoolyard? - How can we display data collected? - What are the steps involved in collecting, arranging, and displaying data?	Students should be <i>introduced</i> to these concepts: - Data - Prediction

- 2.12 The student will create a model of a physical object or process in order to show relationships with or without a computing device (e.g., water cycle, butterfly life cycle, seasonal weather patterns).

Context of the Standard
Scientists, computer scientists, mathematicians, and programmers construct and use models to better conceptualize and understand phenomena under investigation or to develop a possible solution to a proposed problem. Models include diagrams, physical replicas, mathematical representations, analogies, and computer simulations. Models are used to represent a system (or parts of a system)

Context of the Standard
under study, to aid in the development of questions and explanations, to generate data that can be used to make predictions, and to communicate ideas to others. In second grade, students are expected to create simple models. These models may be created using a computing device.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create a model of an object or process both individually and as a class. • Compare and contrast attributes of a model.	Students should <i>investigate</i> these concepts: • What are examples of models that we see and use regularly? • What are examples of things in the world that you can model? • What kinds of things do you need to know before you begin to make a model?	Students should be <i>introduced</i> to these concepts: • Model

Impacts of Computing

2.13 The student will compare and contrast examples of how computing technology has changed and improved the way people live, work, and interact.

Context of the Standard
People have always used devices to assist in computation. They can help in the collection, storage, or manipulation of data. Early computers used mechanical components to perform calculations. In the early 1800s, the first programmable computers were created. They were still limited in their capability, and relied heavily on people to do more complex computation. These people were referred to as computers due to their similar role. Many of these computers were women who were employed in commerce,

Context of the Standard
government, military, and research establishments. The development of computing technology has expanded exponentially over the past 100 years. The development and modification of computing technology is driven by people's needs and wants. Computing technologies influence, and are influenced by, cultural practices.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify different types of technologies that people use in their daily lives. - Explain tasks that are made easier because of computing technology.	Students should <i>investigate</i> these concepts: - What do you think is the earliest example of a computer? - How do computers affect your daily life? - How do computers make certain tasks easier? - What would your life be like without computing devices?	Students should be <i>introduced</i> to these concepts:

2.14 The student will identify and model responsible behaviors when using information and technology.

Context of the Standard
Responsible behavior should always be used when working with computers, such as not sharing login information, keeping passwords private, and logging off when finished. These behaviors apply regardless of whether a student is at school or on a computer at another location. In addition to keeping information private, responsible behaviors should be exhibited when engaging in online communications. Online communication facilitates positive interactions, such as sharing ideas with many people, but the public and anonymous

Context of the Standard
nature of online communication also allows for intimidating and inappropriate behavior in the form of cyberbullying. Cyberbullying is a form of bullying that occurs when online communications are sent that are intimidating or threatening in nature.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Interact responsibly with peers when using technology. - Describe what information should be shared and not shared. - Describe online behaviors that may be harmful to others. - Practice responsible behaviors at all times when using computers.	Students should <i>investigate</i> these concepts: - What are examples of responsible online behavior? - What information is acceptable to share online? - What are behaviors that should be avoided when interacting with others online? - What should you do if a person sends you information that is intimidating, threatening, or that makes you feel badly? - What should you do if a friend is being bullied or cyberbullied?	Students should be <i>introduced</i> to these concepts: Cyberbullying

Networking and the Internet

2.15 The students will discuss in partners and as a class how information can be communicated electronically (e.g., email, social media, video conferencing, blogging).

Context of the Standard

Context of the Standard

Online communication facilitates positive interactions, such as sharing ideas with many people, including friends and family around the world. It also allows opportunities for scientists, mathematicians, business people, and many other professionals to communicate about projects they are working on together. Types of electronic communication include email, video conferencing, blogs, and social media platforms. People with similar interests can meet through social media or email and share information. When using social media or email with strangers, do not share personal information such as phone number or address. Never share pictures of yourself with others you do not know using any type of electronic communication.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Understand that information can be communicated electronically. • Describe different types of electronic communication.	Students should <i>investigate</i> these concepts: • What are different ways that people communicate electronically? • Why do people want to communicate in different ways?	Students should be <i>introduced</i> to these concepts: • Blog • Email • Social media • Video conference

Grade Three

The standards for third grade place an emphasis on decomposing larger problems and utilizing the iterative design process to develop a plan to construct and execute programs. Students in third grade are introduced to using computing systems to model attributes and behaviors associated with a concept. The accurate use of terminology as well as the responsible use of technology will continue to be built upon. The foundational understanding of computing and the use of technology will be an integral component of successful acquisition of skills across content areas.

Algorithms and Programming

- 3.1 The student will construct sets of step-by-step instructions (algorithms), both independently and collaboratively
- using sequencing;
 - using loops (a wide variety of patterns such as repeating patterns or growing patterns); and
 - using events.

Context of the Standard

Algorithms are commonly used in school and at home as students engage in step by step activities that are done on a routine basis. Students can create algorithms as they describe and sequence tasks that are part of daily activities. Students can also use loops to repeat steps when a task requires a repeated action or actions.

In third grade, the construction of loops becomes more complex as students use a wide variety of patterns to include repeating and growing patterns. In a repeating pattern the units of the pattern repeat and remain the same. In a growing pattern, an addition is added to the pattern causing the pattern to change every time it repeats. Growing patterns involve a progression from step to step which make them more difficult for students than repeating patterns.

Repeating and growing patterns are foundational in mathematics in the development of algebraic reasoning and in computer science in developing computational thinking.

Sample numeric patterns include:

- 6, 9, 12, 15, 18,....(growing pattern);
- 1, 2, 4, 7, 11, 16,....(growing pattern);
- 20, 18, 16, 14,.....(growing pattern); and

Context of the Standard
1, 3, 5, 1, 3, 5, 1, 3, 5,.....(repeating pattern). Students in third grade are expected to use events when constructing algorithms. In computer science, an event is an action or occurrence detected by a program. Events can be user actions, such as clicking a mouse button or pressing a key, or system occurrences, such as a timer or low battery.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify and describe algorithms used to accomplish a variety of tasks. - Differentiate between growing and repeating patterns. - Describe how an event signals the start of an algorithm. - Construct algorithms that use loops and events. - Compare and contrast a repeating pattern and a growing pattern.	Students should <i>investigate</i> these concepts: - How do you decide when to use a loop in a sequence? - What are examples of repeating patterns and growing patterns? - What are examples of events in a plugged or an unplugged activity?	Students should <i>apply</i> these terms in context: - Algorithm - Loop - Event

- 3.2 The student will construct programs to accomplish tasks as a means of creative expression using a block or text based programming language, both independently and collaboratively
- using sequencing;
 - using loops (a wide variety of patterns such as repeating patterns or growing patterns); and
 - identifying events.

Context of the Standard

Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem. A computational artifact is anything created by a human using a computer. Examples of computation artifacts include programs, images, audio, videos, presentations, or web page files.

Computing has the potential to provide students' opportunities to extend their creative expression to solve problems, create computational artifacts, and develop new knowledge. As students create block and text based programs, they move from being mere consumers of content to engaging in the subject matter by creating computational artifacts.

In third grade, students are expected to use block based or text based programming to develop basic programs that include sequences, loops, and events. This may be the first time a student uses a text or block-based application to construct programs. Prior grades may have used unplugged activities when integrating computer science standards into instruction.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Construct programs using text or block-based applications. - Modify programs to use loops when appropriate.	Students should <i>investigate</i> these concepts: - How would you write instructions for an action that repeats itself? - What are different ways that you can signal the start of a program? - How are loops and events used when constructing programs? - What are examples of creative products	Students should <i>apply</i> these terms in context: - Algorithm - Loop - Repeat - Event

Essential Skills	Essential Questions	Essential Vocabulary
	that you can use a sequence to make?	

3.3 The student will analyze, correct, and improve (debug) an algorithm that includes sequencing, events, and loops.

Context of the Standard
In order to determine if an algorithm is an appropriate reflection of the steps that must occur in order to complete a task, the students should check that the sequence of steps and any embedded loops that compose the algorithm works as intended. If the algorithm does not work as intended, the students should determine what changes could be made to the algorithm in order to complete the task. These changes may include adding, deleting, rearranging, or changing a step in order to obtain the intended outcome. This process can be conducted for both computer programs and unplugged activities. The process of revising a program so that it works as intended is called debugging.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe how an algorithm did not work (e.g., character is not moving as intended). - Analyze a sequence of steps that is flawed and determine possible solution(s). - Implement a proposed adjustment to a sequence that did not work as intended.	Students should <i>investigate</i> these concepts: - If your algorithm is not working, how could you find the error? - Once you have found an error in your algorithm, how do you decide what adjustment needs to be made? - How can the order of your steps affect what happens?	Students should <i>apply</i> these terms in context: - Bug - Debug - Program

- 3.4 The student will create a plan as part of the iterative design process, independently and/or collaboratively using a variety of strategies (e.g., pair programming, storyboard, flowchart, pseudo-code, story map).

Context of the Standard
An iterative design process is a process in which there is repetition of steps of a process in order to generate a sequence of outcomes. Many occupations and content areas use an iterative design process, including computer science and engineering. In computer science, the development of programs uses an iterative process involving design, implementation, and review. The design stage occurs before writing code. This is a planning stage in which the programmers gather information about the problem and sketch out a solution. This design process may include the use of pseudocode - writing out the steps of a program in English to make sure the flow of control and logic make sense. During the implementation stage, the planned design is expressed in a programming language (code) that can be made to run on a computing device. During the review stage, the design and implementation are checked for adherence to program requirements, correctness, and usability. This review could lead to changes in implementation and possibly design, which demonstrates the iterative nature of the process. The student should focus on the planning portion of the iterative design process in third grade.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Design an algorithm using a planning tool. • Review and revise a plan to better fit the needs of a task. • Communicate how an iterative design process can improve an algorithm.	Students should <i>investigate</i> these concepts: • How can you use planning tools to create a program (just like you would for a story)? • Why is reviewing and revising your work important? • Why is planning out a story or program an important part of the writing process?	Students should <i>apply</i> these terms in context: • Storyboard • Graphic organizer • Pair programming

- 3.5 The student will compare and contrast a group of items based on attributes or actions classified into at least two sets and two subsets.

Context of the Standard
Objects and actions have attributes; these attributes allow people to group items into sets. Attributes may be physical properties, behaviors, or actions. Actions in computer science can be displayed as step-by-step sequences (algorithms). Categorizing of attributes or actions relies on careful observation of patterns and similarities and differences. In this standard, students are expected to analyze sets of items and compare and contrast the attributes that led to the development of the set. Students can use additional attributes to categorize sets into subsets. In block based programming environments, commands are grouped into categories based on function. In higher level programming languages, data are often classified by the type and format of the information.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Name multiple ways to sort a set of objects. - Sort and group (classify) objects into appropriate sets (categories) based on multiple attributes. - Classify objects into subsets based on a secondary attribute. - Label attributes of a set of objects that has been sorted.	Students should <i>investigate</i> these concepts: - Why is it useful to sort objects into sets and why is it helpful in our daily lives? (Given a set of objects) How many different ways can you find to organize these objects? - How are items organized with multiple attributes? - What attributes can you use take a set of things that you have sorted and then sort them into subsets? - Why is the sorting of attributes important for computer science?	Students should <i>apply</i> these terms in context: Subset

3.6 The student will break down (decompose) a larger problem into smaller subproblems, independently or collaboratively.

Context of the Standard
When approaching a task it is sometimes easier to break the problem down into manageable chunks. Programs can also be broken down into smaller parts to facilitate their design, implementation, and review. This is called program decomposition. Decomposition helps in addressing aspects of program development, such as testing, by allowing people to focus on one piece at a time. Decomposition also enables different people to work on different parts of a program at the same time. An example of decomposition at this level is creating an animation by separating a story into different scenes. For each scene, a background needs to be selected, characters placed, and actions programmed. The instructions required to program each scene may be similar to instructions in other programs. Programs can also be built by adding together these smaller components to complete a task. In third grade, students are expected to break down large problems into subproblems when designing or debugging programs.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Analyze a problem and determine sets of smaller problems. - Explain why dividing problems into subproblems will help in completing a project and in planning or testing programs.	Students should <i>investigate</i> these concepts: - If you have a big job to do, what are ways you can break it down to make it easier? - Why does breaking a problem down into smaller problems make the overall task easier?	Students should <i>apply</i> these terms in context: Decompose

3.7 The student will give credit to sources when borrowing or changing ideas (e.g., using information and pictures created by others, using music created by others, remixing programming projects).

Context of the Standard

As students start to work with different artifacts (reference materials, resources, etc.) they should understand that these sources of information were created by others. Authors, illustrators, and programmers are responsible for the creation of many sources of information that are used in the classroom and at home. As students choose to use, reference, or modify some of these sources in their own work, they are expected to recognize the original creator of the source. This practice should be reiterated throughout a student's K-12 education and beyond.

Other topics related to copyright are plagiarism, fair use, and properly citing online sources. Knowledge of specific copyright laws is not an expectation at this level. This standard supports English standards as they learn about plagiarism in writing.

Students are introduced to these concepts but are not expected specific citing practices in third grade.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Review a program they created and identify portions that are created by others. • Explain why it is important to give credit to authors. • Describe when it is acceptable to use people's work, and how to give credit to sources.	Students should <i>investigate</i> these concepts: • How can you find the creator of an artifact? • What are examples of artifacts that need to need to have their creators credited? • Why is important to give credit for using someone else's idea, even if you aren't quoting them directly?	Students should <i>apply</i> these terms in context: • Author • Illustrator • Composer

Computing Systems

3.8 The student will model how a computing system works including input and output.

Context of the Standard

A system is defined as a regularly interacting or interdependent group of items forming a unified whole. Systems always have inputs and outputs. In computer science, input and output, also referred to as I/O, is the communication between an information processing system, such as a computer, and the outside world, possibly a human or another information processing system.

Inputs are the signals or data received by the system; these include electricity, the movements and clicks of your mouse, and the keys you type on a keyboard. An output is whatever comes out of the system; for example, outputs include data and what can be seen on the computer screen.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Describe how a computing system may take in input. • Describe how a computing system may produce output. • Model a simple computing system indicating inputs and outputs.	Students should <i>investigate</i> these concepts: • What is a system? • Why are computers considered a system? • What are the different types of output that a computer can produce?	Students should <i>apply</i> these terms in context: • Computing system • Input • Output

- 3.9 The student will identify, using accurate terminology, simple hardware and software problems that may occur during use, and apply strategies for solving problems (e.g., rebooting the device, checking for power, checking network availability, closing and reopening an app).

Context of the Standard
As with any system, there are times that a computer system does not work as intended. Although computing systems may vary, common troubleshooting strategies can be used on them, such as checking connections and power, or swapping a working part in place of a potentially defective part. Rebooting a machine is commonly effective because it resets the computer. Since computing devices are composed of an interconnected system of hardware and software, troubleshooting strategies may need to address both. Common troubleshooting strategies, such as checking that power is available, checking that physical and wireless connections are working, and clearing out the working memory by restarting programs or devices, are effective for many systems. Students in third grade are expected to use accurate terminology to describe simple problems with computer hardware and software.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify when a device or program is not working properly. • Communicate that a device or program is not working. • Perform simple troubleshooting tasks (e.g., rebooting the computer).	Students should <i>investigate</i> these concepts: • How can you tell a computer is not working as intended? • How can you find out specifically why your computer is not working? • What are different troubleshooting tactics you should try if a program is not working? • Why is it important to be as specific as possible when you are describing a problem?	Students should <i>apply</i> these terms in context: • Reboot • Troubleshoot

Cybersecurity

3.10 The student will identify problems that relate to inappropriate use of computing devices and networks.

Context of the Standard

Computer networks, including the Internet, can be used to connect people to other people, places, information, and ideas. In order to keep students safe, schools and divisions have rules on the appropriate use of technology. As students increase their use of the networks and interact with others outside of the school or home environment, digital safety is an increasing concern. Students should be aware of what is allowed and not allowed when using division/school technology.

Appropriate use of technology as well as school and division rules when using technology should be reviewed with students on a regular basis. Consistent monitoring of students when engaged with technology should be conducted at all times.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify causes and effects related to inappropriate use of computing devices. - Identify real-life situations they encounter while using computing devices that could cause problems in school or at home. - Describe how a technology-related problem could be avoided or prevented.	Students should <i>investigate</i> these concepts: - What is appropriate use of technology? - If you see someone using technology inappropriately in school, how should you notify the proper person? - What are some consequences of inappropriate use of computing technology?	Students should <i>apply</i> these terms in context: Computing device

- 3.11 The student will create examples of strong passwords, explain why strong passwords should be used, and demonstrate proper use and protection of personal passwords.

Context of the Standard
Connecting devices to a network or the Internet provides great benefit, but care must be taken to protect private information such as a student's name, phone number, and address. Passwords are used to protect devices and information from unauthorized access. Because computer programs can be used to guess passwords, strong passwords have characteristics that make them more difficult to guess. Many sites have rules as to the length and composition of passwords; these rules help create stronger passwords. The practice of not sharing passwords should be emphasized in the classroom and at home. Suggestions for creating strong passwords include: 1. Use uppercase and lowercase letters. 2. Use numbers. 3. Use symbols. 4. Use at least 8 characters. 5. Don't use words from a dictionary. 6. Don't use the same password twice. 7. Don't use personal information.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain how a password helps protect the privacy of information. • Respect other students' password privacy. • Explain how logging off devices can protect your information.	Students should <i>investigate</i> these concepts: • What are the components of a strong password? • Why should you change your password periodically? • Why should you have a different password for different accounts?	Students should <i>apply</i> these terms in context: • Password

Essential Skills	Essential Questions	Essential Vocabulary
Classify passwords as strong or weak.		

Data and Analysis

3.12 The student will use a computer to observe and analyze data in order to draw conclusions and make predictions.

Context of the Standard
When answering questions about text in history or English or investigating a question in science, evidence should be used to support your answer. Data are a form of evidence that can be used when answering questions or in making predictions. One way to access data is through the computer. The computer can be used to construct tables and graphs from data collected in class; it can also be the source of existing data sets that have been compiled by others. The ability to determine what type of data is needed to answer a question and use a computer to find these data are skills needed in many career and academic fields.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Use a computer to organize data using various forms of data collection. - Analyze a data set to identify a pattern or make a prediction.	Students should <i>investigate</i> these concepts: - What can you learn from looking at your data in different ways? - How can you use the data you have collected to make a prediction or answer a question?	Students should <i>apply</i> these terms in context: - Data - Prediction

Essential Skills	Essential Questions	Essential Vocabulary
Use data to answer a question or make a prediction.	How does a computer help you to look at data in different ways?	

- 3.13 The student will create an artifact using computing systems to model the attributes and behaviors associated with a concept (e.g., day and night, animal life cycles, plant life cycles).

Context of the Standard
Scientists, computer scientists, mathematicians, and programmers construct and use models to better conceptualize and understand phenomena under investigation or to develop a possible solution to a proposed problem. Models include diagrams, physical replicas, mathematical representations, analogies, and computer simulations. Models are used to represent a system (or parts of a system) under study, to aid in the development of questions and explanations, to generate data that can be used to make predictions, and to communicate ideas to others.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create an artifact to model a concept using a computing system. • Describe how a model reflects the attributes or behaviors of a concept.	Students should <i>investigate</i> these concepts: • What are examples of models that we see and use regularly? • What are examples of things in the world that you can model? • What kinds of things do you need to know before you begin to make a model? • How does a computer model help us learn and predict things about systems?	Students should <i>apply</i> these terms in context: • Model

Impacts of Computing

3.14 The student will identify computing technologies that have changed the world and express how those technologies influence, and are influenced by, cultural practices.

Context of the Standard

The development and modification of computing technology is driven by people's needs and wants. New computing technology is created and existing technologies are modified to increase their benefits (e.g., Internet search recommendations), decrease their risks (e.g., autonomous cars), and meet societal demands (e.g., smartphone apps). Computing technologies influence, and are influenced by, cultural practices. Increased Internet access and speed have allowed people to share cultural information but have also affected the practice of traditional cultural customs. An example of this is that people collaborate and communicate in different ways than they did decades ago. Storytelling and the sharing of information that was conducted between people in a face to face environment now happens virtually much of the time.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify computing technologies that have changed the world. - Explain how the technology is influenced by culture. - Explain how the culture can affect the technology.	Students should <i>investigate</i> these concepts: - What are examples of computing technologies that changed the world? - How has technology, like mobile phones, changed society? - How does society influence the technology that we invent? - If you could design a new computing technology, what would it do, and why?	Students should <i>apply</i> these terms in context: Internet

- 3.15 The student will identify the positive and negative impacts of the pervasiveness of computers and computing in daily life (e.g., downloading videos and audio files, electronic appliances, wireless Internet, mobile computing devices, GPS systems, wearable computing).

Context of the Standard
The use of technology, including computers, has allowed for global communication and has revolutionized the everyday access of information, whether for business, scientific or personal use. Although there are many positive impacts in using technology, there are also times when computer use has impacted us in undesirable ways. As computer technology continues to advance and new generations of machines grow faster and have greater capabilities, the machines become more deeply fixed in daily life, magnifying both the benefits and the downside risks. Positive impacts include easy access to information, automated machinery, and fast and accurate data processing. Negative impacts include an increase in sedentary lifestyles, family and leisure interruption, and loss of privacy.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify how the use of computers and computing positively influences daily life. - Identify how the use of computers and computing negatively influences daily life.	Students should <i>investigate</i> these concepts: - How do computing devices make your life easier? - How have computing devices made people's lives more complicated? - What are ways to limit the negative influences of computing devices?	Students should <i>apply</i> these terms in context:

3.16 The student will identify social and ethical issues that relate to computing devices and networks.

Context of the Standard
People can work in different places and at different times to collaborate and share ideas when they use technologies that reach across the globe. These social interactions affect how local and global groups interact with each other. As with any social interaction, there are manners that people should use when interacting with others. The use of manners when collaborating or interacting with others through computing devices or networks is more complex since many times the communication is done without seeing the person on the other side of the communication. Care should be taken when sharing information so that the intent of the message is not misunderstood by the person on the other end of the communication. In addition, due to the anonymous nature of online communication, intimidating and inappropriate behavior in the form of cyberbullying may occur. Cyberbullying is a form of bullying that occurs when online communications are sent that are intimidating or threatening in nature.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify problems that arise from computer use. - Determine solutions to common computer use issues.	Students should <i>investigate</i> these concepts: - How could computing technology make it easier for people to engage in negative behavior? - What should you do if you see other people using a computer to do harm to others? - What is cyberbullying?	Students should <i>apply</i> these terms in context: Cyberbullying

Networking and the Internet

3.17 The students will discuss in partners and as a class that information can be transmitted using computing devices via a network (e.g., email, blogging, video messaging).

Context of the Standard
Just like electricity travels through a closed circuit, information in a computing system needs a physical or wireless path to travel to be sent and received. Information is broken into smaller pieces that are sent independently and reassembled at the destination. There are physical paths for communicating information, such as ethernet cables, and wireless paths, such as Wi-Fi. Often, information travels on a combination of physical and wireless paths; for example, wireless paths originate from a physical connection point. At this level, the priority is understanding that there are different ways to transfer information, rather than the details of how routers and switches work and how to compare paths. <i>Students are not expected to know the exact mechanisms for conducting the transmission in third grade.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify types of electronic communications (e.g., email, blogging, text messaging, video messaging). • Discuss the different types of information that can be communicated through different transmission media. • Explain that electronic pathways are needed to transfer information within computing systems.	Students should <i>investigate</i> these concepts: • What are the different types of electronic communication? • How do you decide which method of communication is best for a particular situation? • How does information travel from computing device to computing device?	Students should <i>apply</i> these terms in context: • Network • Email • Blog • Text message • Video conferencing

Grade Four

The fourth-grade standards place emphasis on constructing programs and utilizing algorithms to accomplish a task. Students continue to decompose larger problems into smaller tasks. In fourth grade, students begin to think about the impacts of computing and computing devices. The accurate use of terminology as well as the responsible use of technology will continue to be built upon. The foundational understanding of computing and the use of technology will be an integral component of successful acquisition of skills across content areas.

Algorithms and Programming

- 4.1 The student will construct sets of step-by-step instructions (algorithms) both independently and collaboratively
- using sequencing;
 - using loops;
 - using variables to store and process data; and
 - performing number calculations on variables (e.g., addition, subtraction, multiplication and division).

Context of the Standard

Algorithms are commonly used in school and at home as students engage in step-by-step activities that are done on a routine basis. Students can create algorithms as they describe and sequence tasks that are part of daily activities. Students can also use loops to repeat steps when a task requires a repeated action or actions. In third grade, the construction of loops becomes more complex as students use a wide variety of patterns to include repeating and growing patterns. In a repeating pattern the units of the pattern repeat and remain the same. In a growing pattern, an addition is added to the pattern causing the pattern to change every time it repeats. Growing patterns involve a progression from step to step which make them more difficult for students than repeating patterns.

In fourth grade, algorithms become more complex through the addition of variables. Variables in a computer program are analogous to "Buckets" or "Envelopes" where information can be maintained and referenced. On the outside of the bucket is a name. When referring to the bucket, we use the name of the bucket, not the data stored in the bucket. Many programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.

In fourth grade, understanding how to use variables to conduct number calculations (e.g., addition, subtraction, multiplication and

Context of the Standard
<i>division) is sufficient. Manipulation on non-numeric data types is not expected.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify and describe algorithms used to accomplish a variety of tasks. - Describe when a variable can be used. - Identify a variable in an algorithm. - Construct algorithms that use loops and variables. - Apply the use of variables in a math calculation.	Students should <i>investigate</i> these concepts: - What are the kinds of tasks that you can write an algorithm to complete? - What is a variable? - Why do we use variables in algorithms? - How can you write an algorithm to complete basic mathematical calculations?	Students should <i>apply</i> these terms in context: - Algorithm - Loop - Variable

- 4.2 The student will construct programs to accomplish a task as a means of creative expression using a block or text based programming language, both independently and collaboratively
- using sequencing;
 - using loops;
 - using variables; and
 - performing number calculations (e.g., addition, subtraction, multiplication and division) on variables.

Context of the Standard

Context of the Standard

Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem. A computational artifact is anything created by a human using a computer. Examples of computation artifacts include programs, images, audio, videos, presentations, or web page files. Computing has the potential to provide students opportunities to extend their *creative expression* to solve problems, create computational artifacts, and develop new knowledge. As students create block and text based programs, they move from being mere consumers of content to engaging in the subject matter by creating computational artifacts.

In fourth grade, students are expected to use block based or text based programming to develop basic programs that include sequences, loops, and variables.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Construct a program to accomplish an activity. • Modify algorithms to use loops when appropriate. • Declare a variable to store values when appropriate in an algorithm. • Apply the use of variables in a math calculation.	Students should <i>investigate</i> these concepts: • What is the role of a variable in a program? • How do you decide when to use a loop in your algorithm? • How do you decide when to use a variable in your algorithm?	Students should <i>apply</i> these terms in context: • Algorithm • Loop • Variable

4.3 The student will analyze, correct, and improve (debug) an algorithm that includes sequencing, events, loops and variables.

Context of the Standard

Context of the Standard

In order to determine if an algorithm is an appropriate reflection of the steps that must occur in order to complete a task, the students should review the sequence of steps and any embedded loops that compose the algorithm to determine if it works as intended.

During the review stage, the design and implementation are checked for adherence to program requirements, correctness, and usability. This review could lead to changes in implementation and possibly design, which demonstrates the iterative nature of the process. If the algorithm does not work as intended, the students should determine what changes could be made to the algorithm in order to complete the task. These changes may include adding, deleting, rearranging, or changing a step in order to obtain the intended outcome.

The process of revising a program so that it works as intended is called debugging.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe how an algorithm did not work (e.g., character is not moving as intended). - Analyze a sequence of steps that is flawed and determine possible solution(s). - Implement a proposed adjustment to a sequence that did not work as intended.	Students should <i>investigate</i> these concepts: - If your algorithm is not working, how could you find the error? - Once you have found an error in your algorithm, how do you decide what adjustment needs to be made? - How can the order of your steps affect what happens? - What aspects of a variable could cause problems with your algorithms?	Students should <i>apply</i> these terms in context: - Bug - Debug

- 4.4 The student will create a plan as part of the iterative design process, independently and/or collaboratively using a variety of strategies (e.g., pair programming, storyboard, flowchart, pseudocode, story map).

Context of the Standard

An iterative design process is a process in which there is repetition of steps of a process in order to generate a sequence of outcomes. Many occupations and content areas use an iterative design process, including computer science and engineering. In computer science, the development of programs uses an iterative process involving design, implementation, and review. The design stage occurs before writing code. The planning stage is when the programmers gather information about the problem and sketch out a solution. This design process may include the use of pseudocode - writing out the steps of a program in English to make sure the flow of control and logic make sense. During the implementation stage, the planned design is expressed in a programming language (code) that can be made to run on a computing device. During the review stage, the design and implementation are checked for adherence to program requirements, correctness, and usability. This review could lead to changes in implementation and possibly design, which demonstrates the iterative nature of the process.

This standard focuses on the planning portion of the iterative design process.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Design an program using a planning tool. • Review and revise a plan to better fit the needs of a task. • Communicate how an iterative design process can improve an program.	Students should <i>investigate</i> these concepts: • How can you use planning tools to create an program (just like you would for a story)? • Why is reviewing and revising your work important? • Why is planning out a story or program an important part of the writing process?	Students should <i>apply</i> these terms in context: • Planning tool • Storyboard • Graphic organizer • Pseudocode

4.5 The student will classify and arrange a group of items based on the attributes or actions.

Context of the Standard

Objects and actions have attributes; these attributes allow people to group items. Attributes may be physical properties, behaviors, or actions. Actions in computer science can be seen in step by step sequences (algorithms). Categorizing of attributes or actions relies on careful observation of patterns and similarities and differences. In this standard, students are expected to analyze groups of items and compare and contrast the attributes that led to the development of the group.

In block based programming environments, commands are grouped into categories based on function. In higher level programming languages, data are often classified by the information held within.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Name multiple ways to sort a set of objects. • Sort and group (classify) objects into appropriate sets (categories) based on multiple attributes. • Classify objects into subsets based on a secondary attribute. • Label attributes of a set of objects that has been sorted.	Students should <i>investigate</i> these concepts: • Why is it useful to sort objects into sets and why is it helpful in our daily lives? • (Given a set of objects) How many different ways can you find to organize these objects? • How are items organized with multiple attributes/actions? • What attributes/actions can you use take a set of things that you have sorted and then sort them into subsets?	Students should <i>apply</i> these terms in context: • Subset

4.6 The student will break down (decompose) a larger problem into smaller sub-problems, both independently and collaboratively.

Context of the Standard

Context of the Standard

When approaching a task it is sometimes easier to break the problem down into manageable chunks. Programs can also be broken down into smaller parts to facilitate their design, implementation, and review; this is referred to as decomposition. Decomposition helps in addressing aspects of program development, such as testing, by allowing people to focus on one piece at a time. Decomposition also enables different people to work on different parts at the same time. An example of decomposition at this level is creating an animation by separating a story into different scenes. For each scene, a background needs to be selected, characters placed, and actions programmed. The instructions required to program each scene may be similar to instructions in other programs. Programs can also be built by adding together smaller components to complete a task.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Analyze a problem and determine sets of smaller problems. - Explain how decomposition can be helpful when planning or testing a program?	Students should <i>investigate</i> these concepts: - If you have a big job to do, what are ways you can break it down to make it easier? - Why does breaking a problem down into smaller problems make the overall task easier?	Students should <i>apply</i> these terms in context: Decompose

4.7 The student will give credit to sources when borrowing or changing ideas (e.g., using information, pictures created by others, using music created by others, remixing programming projects).

Context of the Standard

Context of the Standard

As students start to work with different artifacts (reference materials, resources, etc.) they should understand that these sources of information were created by others. Authors, illustrators, and programmers are responsible for the creation of many sources of information that are used in the classroom and at home. As students choose to use some of these sources in their own work, they are expected to recognize the original creator of the source. This practice should be reiterated throughout a student's K-12 education and beyond.

Ethical complications arise from the opportunities provided by computing. The ease of sending and receiving copies of media on the Internet, such as video, photos, and music, creates the opportunity for unauthorized use, such as online piracy, and disregard of copyrights, such as lack of attribution.

Other topics related to copyright are plagiarism, fair use, and properly citing online sources. Knowledge of specific copyright laws is not an expectation at this level. This standard supports English standards as they learn about plagiarism in writing.

Students are not responsible for specific copyright laws or using citing practices in fourth grade.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Review a program they created and identify any portions that were created by others. • Explain why it is important to give credit to authors. • Describe when it is acceptable to use people's work, and how to give credit to sources.	Students should <i>investigate</i> these concepts: • How can you find the creator of an artifact? • What are examples of artifacts that need to need to have their creators credited? • Why is important to give credit for using someone else's idea, even if you aren't quoting them directly?	Students should <i>apply</i> these terms in context: • Author • Illustrator • Composer

Computing Systems

4.8 The student will model how a computing system works including input and output, processors, and sensors.

Context of the Standard

A system is defined as a regularly interacting or interdependent group of items forming a unified whole. Computing systems require inputs and outputs. Input and output, also referred to as I/O, is the communication between an information processing system, such as a computer, and the outside world, possibly a human or another information processing system.

Inputs are the signals or data received by the system. There is a wide variety of digital collection tools used for gathering and inputting digital data. Tools are chosen based upon the type of measurement they use as well as the type of data people wish to observe. These collection tools include the movements and clicks of your mouse and the keys you type on a keyboard. Sensors are also used in computing systems, such as in robotics, to detect information and serve as input devices for the system. A sensor that is be used with robotic devices is a light sensor that detects changes in brightness.

An output is whatever comes out of the system; for example, outputs include data and what can be seen on the computer screen or how the robotic device responds based on the input from the sensor.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Describe how a computing system may use different components to receive input including sensors. • Identify the processor as the component which manipulates input into output. • Describe how a computing system may produce output. • Model a simple computing system	Students should <i>investigate</i> these concepts: • What are examples of sensors that take in input? • What kind of input can a computer take in, and what is required to take in the different types? • What are the main components of a computing system? • When you input information into a computer, what path does it take to	Students should <i>apply</i> these terms in context: • Input • Output • Processor • Sensor

Essential Skills	Essential Questions	Essential Vocabulary
indicating inputs and outputs. Describe the role of a processor in a computing system.	become output? What are the different types of output that a computer can produce?	

- 4.9 The student will identify, using accurate terminology, simple hardware and software problems that may occur during use, and apply strategies for solving problems (e.g., rebooting the device, checking for power, checking for network availability, closing and reopening an app).

Context of the Standard
As with any system, there are times that a computer system does not work as intended. Although computing systems may vary, common troubleshooting strategies can be used on them, such as checking connections and power or swapping a working part in place of a potentially defective part. Rebooting a machine is commonly effective because it resets the computer. Since computing devices are composed of an interconnected system of hardware and software, troubleshooting strategies may need to address both. Students in fourth grade are expected to use accurate terminology to describe simple problems with computer hardware and software. Common troubleshooting strategies, such as checking that power is available, checking that physical and wireless connections are working, and clearing out the working memory by restarting programs or devices, are effective for many systems.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify when a device or program is not working properly. - Communicate that a device or program is not working. - Perform simple troubleshooting tasks (e.g., rebooting the computer).	Students should <i>investigate</i> these concepts: - How can you tell a computer is not working as intended? - How can you find out specifically why your computer is not working? - What are different troubleshooting tactics you should try if a program is not working? - Why is it important to be as specific as possible when you are describing a problem?	Students should <i>apply</i> these terms in context: - Reboot - Troubleshoot

Cybersecurity

4.10 The student will identify and explain problems that relate to inappropriate use of computing devices and networks.

Context of the Standard
Computer networks, including the Internet, can be used to connect people to other people, places, information, and ideas. In order to keep students safe, schools and divisions have rules on the appropriate use of technology. As students increase their use of the networks and interact with others outside of the school or home environment, digital safety is an increasing concern. Students should be aware of what is allowed and not allowed when using division/school technology. Appropriate use of technology as well as school and division rules when using technology should be reviewed with students on a regular basis. Consistent monitoring of students when engaged with technology should be conducted at all times.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify and explain causes and effects related to inappropriate use of computing devices. - Identify real-life situations they encounter while using computing devices that could cause problems in school or at home. - Describe how a technology-related problem could be avoided or prevented.	Students should <i>investigate</i> these concepts: - What is appropriate use of technology? - If you see someone using technology inappropriately in school, how should you notify the proper person? - What are some consequences of inappropriate use of computing technology? - What are examples from the news concerning inappropriate use of technology?	Students should <i>apply</i> these terms in context:

- 4.11 The student will create examples of strong passwords, explain why strong passwords should be used, and demonstrate proper use and protection of personal passwords.

Context of the Standard
Connecting devices to a network or the Internet provides great benefit, but care must be taken to protect private information such as a student's name, phone number, and address. Passwords are used to protect devices and information from unauthorized access. Computer programs can be used to guess passwords; therefore, strong passwords have characteristics that make them more difficult to guess. Many sites have rules as to the length and composition of passwords; these rules help create stronger passwords. The practice of not sharing passwords should be emphasized in the classroom and at home. Suggestions for creating strong passwords include: 1. Use uppercase and lowercase letters. 2. Use numbers. 3. Use symbols. 4. Use at least 8 characters. 5. Don't use words from a dictionary. 6. Don't use the same password twice. 7. Don't use personal information.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain how a password helps protect the privacy of information. • Respect other students' password privacy. • Explain how logging off devices can protect your information.	Students should <i>investigate</i> these concepts: • What are the attributes of a strong password? • Why should you change your password periodically? • Why should you have a different password for different accounts?	Students should <i>apply</i> these terms in context: • Password

Essential Skills	Essential Questions	Essential Vocabulary
- Classify passwords as strong or weak. - Create strong passwords to be used at school or at home.		

Data and Analysis

4.12 The student will use a computer to observe, analyze, and manipulate data in order to draw conclusions and make predictions.

Context of the Standard
When answering questions about text in history or English or investigating a question in science, evidence should be used to support your answer. Data are a form of evidence that can be used when answering questions or in making predictions. Data is often sorted or grouped to provide additional clarity. The same data could be manipulated in different ways to emphasize particular aspects or parts of the data set. Computers can be used to obtain, store, and manipulate data. Computers also are used to construct tables and graphs from data collected in class as well as existing data sets that have been compiled by others. The ability to determine what type of data is needed to answer a question and use a computer to find these data are skills needed in many career and academic fields.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Use a computer to organize data using various forms of data collection. • Conduct simple manipulations of data using the computer. • Analyze a data set to identify a pattern or make a prediction. • Use the data to answer a question or make a prediction.	Students should <i>investigate</i> these concepts: • How can you use the data you have collected to make a prediction or answer a question? • How does a computer help you to look at data in different ways? • What can you learn from looking at your data in different ways? • What does a computer allow you to do with data that is more difficult on paper?	Students should <i>apply</i> these terms in context: • Data • Prediction

- 4.13 The student will create an artifact using computing systems to model the attributes and behaviors associated with a concept (e.g., solar system).

Context of the Standard		
Scientists, computer scientists, mathematicians, and programmers construct and use models to better conceptualize and understand phenomena under investigation or to develop a possible solution to a proposed problem. Models include diagrams, physical replicas, mathematical representations, analogies, and computer simulations. Models are used to represent a system (or parts of a system) under study, to aid in the development of questions and explanations, to generate data that can be used to make predictions, and to communicate ideas to others.		
Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create an artifact to model a concept using a computing system. • Describe how a model reflects the attributes or behaviors of a concept.	Students should <i>investigate</i> these concepts: • What are examples of models that we see and use regularly? • What are examples of things in the world that you can model? • What information do you need to construct a model? • How does a computer model help us learn and predict things about systems?	Students should <i>apply</i> these terms in context: • Model

- 4.14 The student will use numeric values to represent non-numeric ideas in the computer (binary, ASCII, pixel attributes such as RGB).

Context of the Standard
Computers use numeric values to store information and perform operations. Information inputted into the computer from various components must be converted into numeric values in order for the computer to use the information and perform functions. Once the function is completed, the numeric values must be converted to a form of output that the user can understand. This output may be in the form of words, images, videos, or sounds. Examples of different ways non-numeric information such as letters or colors can be expressed include the use of different protocols such as binary, ASCII, or RGB. <i>Students are not expected to apply these protocols in fourth grade.</i>

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Understand that computers use numeric values to represent non-numeric ideas. - Give an example of when numeric values can be used to represent non-numeric ideas. - Apply using numeric values to represent non-numeric ideas to in a real-world example.	Students should <i>investigate</i> these concepts: - What is a numeric value? - Why are numbers used to represent non-numeric ideas in the computer? - Why does a computer convert input into a different format? - What are some examples of how numbers are used to represent non-numeric ideas in the computer?	Students should <i>apply</i> these terms in context: - Binary - Pixel - ASCII

Impacts of Computing

4.15 The student will give examples of computing technologies that have changed the world and express how those technologies influence, and are influenced by, cultural practices.

Context of the Standard

Societal and cultural problems provide a demand for the development of new technologies. These new computing technologies are created and existing technologies are modified in order to increase their benefits (for example, Internet search recommendations), decrease their risks (for example, autonomous cars), and meet societal demands (for example, smartphone apps). Increased Internet access and speed have allowed people to share cultural information but have also affected the practice of traditional cultural customs. An example of this is that people collaborate and communicate in different ways than they did decades ago. Storytelling and the sharing of information that was conducted between people in a face to face environment now happens virtually much of the time.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify computing technologies that have changed the world. - Explain how the technology is influenced by society. - Explain how the culture can affect the technology.	Students should <i>investigate</i> these concepts: - What are examples of computing technologies that changed the world? - How has technology, like mobile phones, changed society? - How does society influence the technology that we invent? - If you could design a new computing technology, what would it do, and why?	Students should <i>apply</i> these terms in context: Internet

- 4.16 The student will describe the positive and negative impacts of the pervasiveness of computers and computing in daily life (e.g., downloading videos and audio files, electronic appliances, wireless Internet, mobile computing devices, GPS systems, wearable computing).

Context of the Standard
The use of technology, including computers, has allowed for global communication and has revolutionized the everyday access of information, whether for business, scientific or personal use. Although there are many positive impacts in using technology, there are also times when computer use has impacted us in undesirable ways. As computer technology continues to advance and new generations of machines grow faster and have greater capabilities, the machines become more deeply fixed in daily life, magnifying both the benefits and the downside risks. Positive impacts include easy access to information, automated machinery, and fast and accurate data processing. Negative impacts include an increase in sedentary lifestyles, family and leisure interruption, and loss of privacy.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify how the use of computers and computing positively influences daily life. - Identify how the use of computers and computing negatively influences daily life.	Students should <i>investigate</i> these concepts: - How do computing devices make your life easier? - How have computing devices made people's lives more complicated? - What are ways to limit the negative influences of computing devices?	Students should <i>apply</i> these terms in context:

4.17 The student will describe social and ethical issues that relate to computing devices and networks.

Context of the Standard
People can work in different places and at different times to collaborate and share ideas when they use technologies that reach across the globe. These social interactions affect how local and global groups interact with each other. As with any social interaction, there are manners that people should use when interacting with others. The use of manners when collaborating or interacting with others through computing devices or networks is more complex since many times the communication is done without seeing the person on the other side of the communication. For example, communications should be clear and concise and should never represent the words and actions of others as your own. Care should be taken when sharing information so that the intent of the message is not misunderstood by the person on the other end of the communication. In addition, due to the anonymous nature of online communication, intimidating and inappropriate behavior in the form of cyberbullying may occur. Cyberbullying is a form of bullying that occurs when online communications are sent that are intimidating or threatening in nature.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe problems that arise from computer use. - Practice the use of good computer ethics when interacting with others.	Students should <i>investigate</i> these concepts: - What is cyberbullying? - How could computing technology make it easier for people to engage in negative behavior? - What should you do if you see other people using a computer to do harm to others?	Students should <i>apply</i> these terms in context: Cyberbullying

Networking and the Internet

4.18 The student will identify and explain different ways information can be transmitted using computing devices via a network (e.g., email, images, and videos).

Context of the Standard
Information can be transmitted through both physical and wireless pathways; these pathways are referred to as a network. Network pathways allow communications to occur between computers within the same building or to different locations around the world. These communications exist in a variety of forms to include emails, blogs, images, videos, and through social media platforms. The use of wireless technology allows people to communicate and collaborate, to support work, and to maintain friendships. <i>Students are not expected to know the exact mechanisms for conducting electronic transmissions in fourth grade.</i>

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify types of electronic communications (e.g., email, blogging, text messaging, and video messaging). - Discuss the different types of information that can be communicated through different transmission media. - Explain that networks are needed to transfer information within computing systems.	Students should <i>investigate</i> these concepts: - What are the different types of electronic communication? - How do you decide which method of communication is best for a particular situation? - How does information travel from computing device to computing device? - How does electronic communication allow for greater collaboration between people? - What is meant by a network?	Students should <i>apply</i> these terms in context: - Network - Email - Blog - Text message - Video conferencing

Grade Five

The fifth-grade standards place emphasis on constructing programs and utilizing algorithms to accomplish a task. Students continue to decompose larger problems into smaller tasks and recognize the impacts of computing and computing devices. Students in fifth grade model how computing systems work. The accurate use of terminology as well as the responsible use of technology will continue to be built upon. The foundational understanding of computing and the use of technology will be an integral component of successful acquisition of skills across content areas.

Algorithms and Programming

- 5.1 The student will construct sets of step-by-step instructions (algorithms) both independently and collaboratively,
- using sequencing;
 - using loops;
 - using variables to store and process data;
 - performing number calculations on variables (addition, subtraction, multiplication and division); and
 - using conditionals (if-statements).

Context of the Standard

Algorithms are commonly used in school and at home as students engage in step-by-step activities that are done on a routine basis. Students can create algorithms as they describe and sequence tasks that are part of daily activities. When an algorithm or a set of algorithms is tested, a program has been created.

Programs use sequencing and may include loops and variables; the use of these are dependent on the intended outcome of the program. Students entering fifth grade should have experience with the construction of loops using a wide variety of patterns to include repeating and growing patterns.

In fourth grade, students learn that variables are used to store and process data. Variables in a computer program are analogous to "Buckets" or "Envelopes" where information can be maintained and referenced. On the outside of the bucket is a name. When referring to the bucket, we use the name of the bucket, not the data stored in the bucket. Many programming languages provide variables, which are used to store, modify, and process data. The data type determines the values and operations that can be performed on that data. Examples of operations that may be used on variables include count and sum.

In fifth grade, algorithms become more complex through the addition of conditionals, or "if-statements." Conditionals act as gates in

Context of the Standard

programs. They test a true-false condition, if it is true then the code inside the gate, or conditional, runs. If the test is false, then the program skips the code and moves on to the next command. Conditionals refer to statements that require the computer to determine whether to run a specific set of instructions based upon certain criteria being met. Conditionals enable the computer to “make a decision” concerning what set of directions to follow.

Teacher note: the use of the term variable is used across disciplines in fifth grade and students should be aware of how this term can be interpreted or applied differently depending on the discipline context.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Construct algorithms to include loops, variables, and conditionals. • Identify a variable in an algorithm. • Apply the use of variables in a math calculation in an algorithm. • Assign one or more variables in a computer program to name or categorize data. • Apply the use of conditionals in an algorithm.	Students should <i>investigate</i> these concepts: • When should you use an if-statement in an algorithm? • How do we use variables to complete math problems on a computer? • When do you assign a variable to an expression or a set of data? • When you use an if-statement, how does the computer respond to the directions? • Why are if-statements useful when writing algorithms?	Students should <i>apply</i> these terms in context: • Algorithm • Variable • Conditional (if-statement)

- 5.2 The student will construct programs to accomplish a task as a means of creative expression using a block or text based programming language, both independently and collaboratively
- using sequencing;
 - using loops;
 - using variables;
 - using mathematical operations (addition, subtraction, multiplication and division) variable to manipulate a variable; and
 - using conditionals (if-statements).

Context of the Standard		
Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem. Examples of computation artifacts include programs, images, audio, videos, presentations, or web page files. Computing has the potential to provide students’ opportunities to extend their creative expression to solve problems, create computational artifacts, and develop new knowledge. As students create block and text based programs, they move from being mere consumers of content to engaging in the subject matter by creating computational artifacts. A computational artifact is anything created by a human using a computer. In fifth grade, students are expected to use block based or text based programming to develop basic programs that include sequences, loops, variables, and conditional statements.		
Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Use loops, variables, and conditionals when creating block or text-based programs. • Understand that computing devices can be used as a means for creative expression. • Explain different types of creative products that can be generated using a computing device (e.g. computer games, interactive stories, graphic design, programs, music, and movies). • Determine an original problem and create a solution using a text or block based program.	Students should <i>investigate</i> these concepts: • When might you use an if-statement in a program that is designed for creative expression? • What are examples of different creative products that you can make using a program?	Students should <i>apply</i> these terms in context: • Algorithm • Variable • Conditional (if-statement)

5.3 The student will analyze, correct, and improve (debug) an algorithm that includes sequencing, events, loops, conditionals, and variables.

Context of the Standard
In order to determine if an algorithm is an appropriate reflection of the steps that must occur in order to complete a task, the students should review the sequence of steps and any embedded loops, variables, and number calculations that compose the algorithm works as intended. During the review stage, the design and implementation are checked for adherence to program requirements, correctness, and usability. This review could lead to changes in implementation and possibly design, which demonstrates the iterative nature of the process. If the algorithm does not work as intended, the students should determine what changes could be

Context of the Standard
made to the algorithm in order to complete the task. These changes may include adding, deleting, rearranging, or changing a step in order to obtain the intended outcome. The process of revising a program so that it works as intended is called debugging.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Describe how an algorithm didn't work (e.g., character is not moving as intended). • Analyze an algorithm that is flawed and determine possible solution(s). • Implement a proposed adjustment to a sequence that did not work as intended. • Explain how a proposed adjustment increases the effectiveness of an algorithm.	Students should <i>investigate</i> these concepts: • If your program does not run, how could you correct it? • If your algorithm is not working as intended, how could you fix it? • Once you have found an error in your algorithm, how do you decide what adjustment needs to be made? • How can the sequence of your steps affect what happens?	Students should <i>apply</i> these terms in context: • Bug • Debug

- 5.4 The student will create a plan as part of the iterative design process, both independently and collaboratively using strategies such as pair programming (e.g., storyboard, flowchart, pseudo-code, story map).

Context of the Standard
Many occupations and content areas use an iterative design process, including computer science and engineering. In computer science, the development of programs uses an iterative design process involving design, implementation (programming), and review (debugging) until the program runs correctly. The design stage occurs before beginning to program. The planning stage is when the programmers gather information about the problem and sketch out a solution. This design process may include the use of pseudocode - writing out the steps of a program in English to make sure the flow of control and logic make sense. During the implementation stage, the planned design is expressed in a programming language (code) that can be made to run on a computing device. During the review stage, the design and implementation are checked for adherence to program requirements, correctness, and usability. This is the process of debugging discussed in 5.3. This review could lead to changes in implementation and possibly design, which demonstrates the iterative nature of the process.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Design a program using a planning tool. • Review and revise a plan to better fit the needs of a task. • Communicate how an iterative design process can improve an algorithm.	Students should <i>investigate</i> these concepts: • Why is planning out a story or program an important part of the writing process? • Why is reviewing and revising your work important? • What kinds of jobs require the use of iterative problem solving? • How do people in different careers use the iterative process? • What is the value in making small, targeted, additions or changes to your algorithm rather than large changes?	Students should <i>apply</i> these terms in context: • Planning tool • Storyboard • Pseudocode

- 5.5 The student will break down (decompose) a larger problem into smaller sub-problems, both independently and collaboratively.

Context of the Standard
Large programs are often difficult to imagine and create. Large programs can be broken down, or decomposed, into smaller parts in order to facilitate the design, implementation, and review process. These smaller portions of programs are easier to design and implement. They can then be incorporated with other small components to build toward the overall goal. Programs can also be created by incorporating smaller portions of programs that have already been created. Program decomposition also enables different people to work on different parts at the same time. An example of decomposition at this level is creating an animation by separating a story into different scenes. For each scene, a background needs to be selected, characters placed, and actions programmed. The instructions required to program each scene may be similar to instructions in other programs.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Analyze and decompose a problem into subproblems. - Explain why multiple smaller problems may be easier to solve than one large problem.	Students should <i>investigate</i> these concepts: - Why does breaking a problem down into smaller subproblems make the overall task easier? - How does decomposing a program into subproblems help programmers when debugging a program? - Why would using subproblems in a program be thought of as a time saving measure?	Students should <i>apply</i> these terms in context: Decompose

- 5.6 The student will give credit to sources when borrowing or changing ideas (e.g., using information, pictures created by others, using music created by others, remixing programming projects).

Context of the Standard
As students start to work with different artifacts (reference materials, resources, etc.) they should understand that these sources of information were created by others. Authors, illustrators, and programmers are responsible for the creation of many sources of information that are used in the classroom and at home. As students choose to use some of these sources in their own work, they are expected to recognize the original creator of the source. This practice should be reiterated throughout a student's K-12 education and beyond. Ethical complications arise from the opportunities provided by computing. The ease of sending and receiving copies of media on the Internet, such as video, photos, and music, creates the opportunity for unauthorized use, such as online piracy, and disregard of copyrights, such as lack of attribution. Other topics related to copyright are plagiarism, fair use, and properly citing online sources. Knowledge of specific copyright laws is not an expectation at this level. This standard supports English standards as they learn about plagiarism in writing. <i>Students are not responsible for specific copyright laws or using citing practices in fifth grade.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Review a program they created and identify portions that may have been created by others. • Explain why it is important to give credit to authors. • Describe when it is acceptable to use people's work, and how to give credit	Students should <i>investigate</i> these concepts: • How can you find the creator of an artifact? • What are examples of artifacts that need to need to have their creators credited? • Why is important to give credit for using someone else's idea, even if you aren't quoting them directly?	Students should <i>apply</i> these terms in context: • Author • Illustrator • Composer • Source

Essential Skills	Essential Questions	Essential Vocabulary
to sources. Recognize that different artifacts, including online, programs, and physical (i.e. books, paintings, webpages), have creators.	If you use a portion of someone else's algorithm, why do you need to give credit?	

Computing Systems

5.7 The student will model how a computing system works including input and output, processors, sensors and storage.

Context of the Standard

A system is defined as a regularly interacting or interdependent group of items forming a unified whole. Computing devices are defined as having input, processors, memory and output; these are considered part of a computer system. In computer science, input and output, also referred to as I/O, is the communication between an information processing system, such as a computer, and the outside world, possibly a human or another information processing system. This is how real world information is digitized, or translated in and out of binary.

Inputs are the signals or data received by the system. There is a wide variety of digital collection tools used for gathering and inputting digital data. Tools may be chosen based upon the type of data people wish to observe or by the designers of the system. These collection tools include the movements and clicks of your mouse and the keys you type on a keyboard. Sensors are also used in computing systems, such as in robotics, to detect information and serve as input devices for the system. For example, a robotic device depends on sensors, such as a light sensor, to detect changes in brightness.

In fifth grade, students are introduced to the concept of storage. Computers store data that can be retrieved later. It is also good practice to save data in multiple locations to protect against loss. The storage capacity of a computing device varies as does the amount of storage required for the saving of different media (pictures, videos, text documents, etc). Data can be stored locally on a hard drive or in a global network of servers that are accessed via the Internet, aka The Cloud.

Data is stored in binary. The connection should be made that variables in programs are how we store and access data when programming. A variable is a name given to a spot in the computer's memory. The programmer can access and change the data stored in that location by using the variable name.

Students do not need to know how to change numbers from base ten to binary; however, they do need to understand that an input device like a microphone takes sound waves and represents it in computer memory as binary.

Essential Skills

Essential Questions

Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe how a computing system may use different components to receive input including sensors. - Identify the processor as the component which manipulates input into output. - Describe how a computing system may produce output. - Model a simple computing system indicating inputs and outputs. - Explain how data can be stored in a computer for later use. - Recognize that different types of data require different amounts of storage.	Students should <i>investigate</i> these concepts: - What are examples of sensors or computer components that take in input? - What kind of input can a computer take in? - What are the different types of output that a computer can produce? - What is storage in regards to a computing device? - How does the amount of storage affect how well a computer functions? - How do storage requirements differ between different media?	Students should <i>apply</i> these terms in context: - Input - Output - Processor - Sensor - Storage

- 5.8 The student will identify, using accurate terminology, simple hardware and software problems that may occur during use, and apply strategies for solving problems (e.g., rebooting the device, checking for power, checking network availability, closing and reopening an app).

Context of the Standard
As with any system, there are times that a computer system does not work as intended. Although computing systems may vary, common troubleshooting strategies can be used on them, such as checking connections and power or swapping a working part in place of a potentially defective part. Rebooting a machine is commonly effective because it resets the computer. Since computing devices are composed of an interconnected system of hardware and software, troubleshooting strategies may need to address both. Students in fifth grade are expected to use accurate terminology to describe simple problems with computer

Context of the Standard
hardware and software. Common troubleshooting strategies, such as checking that power is available, checking that physical and wireless connections are working, and clearing out the working memory by restarting programs or devices, are effective for many systems.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify when a device or program is not working properly. • Communicate that a device or program is not working. • Perform simple troubleshooting tasks (e.g., rebooting the computer) • Differentiate hardware and software derived problems.	Students should <i>investigate</i> these concepts: • How can you find out specifically why your computer is not working? • What are different troubleshooting tactics you should try if a program is not working? • Why is it important to be as specific as possible when you are describing a problem? • How can you tell whether a problem is related to hardware or software? • What are examples of hardware/software problems?	Students should <i>apply</i> these terms in context: • Troubleshoot

Cybersecurity

5.9 The student will evaluate and solve problems that relate to inappropriate use of computing devices and networks.

Context of the Standard

Computer networks, including the Internet, can be used to connect people to other people, places, information, and ideas. In order to keep students safe, schools and divisions have rules on the appropriate use of technology. As students increase their use of the networks and interact with others outside of the school or home environment, digital safety is an increasing concern. Students should be aware of what is allowed and not allowed when using division/school technology.

Appropriate use of technology as well as school and division rules when using technology should be reviewed with students on a regular basis. Consistent monitoring of students when engaged with technology should be conducted at all times.

In fifth grade, students should begin to contemplate their role as members of a larger community of technology users and how they will navigate this world ethically and responsibly.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify and explain causes and effects related to inappropriate use of computing devices. - Identify real-life situations they encounter while using computing devices that could cause problems in school or at home. - Describe how a technology-related problem could be avoided or prevented.	Students should <i>investigate</i> these concepts: - What is appropriate use of technology? - If you see someone using technology inappropriately in school, how should you notify the proper person? - What are some consequences of inappropriate use of computing technology? - What are examples from the news concerning inappropriate use of technology? - If you were designing a system to stop	Students should <i>apply</i> these terms in context:

Essential Skills	Essential Questions	Essential Vocabulary
	inappropriate use of technology, what would it look like and why?	

- 5.10 The student will determine whether passwords are strong, explain why strong passwords should be used, and demonstrate proper use and protection of personal passwords.

Context of the Standard
Connecting devices to a network or the Internet provides great benefit, but care must be taken to protect private information such as a student's name, phone number, and address. Passwords are used to protect devices and information from unauthorized access. Computer programs can be used to guess passwords; therefore, strong passwords have characteristics that make them more difficult to guess. Many sites have rules as to the length and composition of passwords; these rules help create stronger passwords. The practice of not sharing passwords should be emphasized in the classroom and at home. Suggestions for creating strong passwords include: • Use uppercase and lowercase letters. • Use numbers. • Use symbols. • Use at least 8 characters. • Don't use words from a dictionary. • Don't use the same password twice. • Don't use personal information.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills:	Students should <i>investigate</i> these concepts:	Students should <i>apply</i> these terms

Essential Skills	Essential Questions	Essential Vocabulary
- Explain how a password helps protect the privacy of information. - Respect other students' password privacy. - Explain how logging off devices can protect your information. - Classify passwords as strong or weak. - Create and use strong passwords to be used in school and home.	- What are the components of a strong password? - Why should you change your password periodically? - Why should you have a different password for different accounts?	in context: Password

Data and Analysis

5.11 The student will use a computer to observe, analyze, and manipulate data in order to draw conclusions and make predictions.

Context of the Standard
When answering questions about text in history or English or investigating a question in science, evidence should be used to support your answer. Data are a form of evidence that can be used when answering questions or in making predictions. Data are often sorted or grouped to provide additional clarity. The same data could be manipulated in different ways to emphasize particular aspects or parts of the data set. Computers can be used to obtain, store, and manipulate data. These data can be used to construct tables and graphs from data collected in class; they can also be sources of existing data sets that have been compiled by others. The ability to determine what type of data is needed to answer a question and use a computer to find these data are skills needed in many career and academic fields.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Use a computer to organize data using various forms (i.e. tables, spreadsheets) of data collection. • Conduct manipulations of data using the computer. • Analyze a data set to identify a pattern or make a prediction. • Use the data or prediction to answer a question. • Display the same data on a computer using multiple representations (e.g. tables, bar graphs, line graphs).	Students should <i>investigate</i> these concepts: • How can you use the data you have collected to make a prediction or answer a question? • How does a computer help you to look at data in different ways? • What can you learn from looking at your data in different formats? • How can computers be used to view data using a variety of formats? • What does a computer allow you to do with data that is more difficult on paper?	Students should <i>apply</i> these terms in context: • Data

5.12 The student will create an artifact using computing systems to model the attributes and behaviors associated with a concept (e.g., rocks).

Context of the Standard
Scientists, computer scientists, mathematicians, and programmers construct and use models to better conceptualize and understand phenomena under investigation or to develop a possible solution to a proposed problem. Models include diagrams, physical replicas, mathematical representations, analogies, and computer simulations. Models are used to represent a system (or parts of a system) under study, to aid in the development of questions and explanations, to generate data that can be used to make predictions, and to communicate ideas to others.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Use a computing system to create an artifact to model a concept. • Describe how a model reflects the attributes or behaviors of a concept.	Students should <i>investigate</i> these concepts: • What are examples of models that we see and use regularly? • What are examples of concepts that you can model? • What kinds of things do you need to know before you begin to make a model? • How does a computer model help us learn and predict things about large, small, and complex systems?	Students should <i>apply</i> these terms in context: • Model

5.13 The student will use numeric values to represent non-numeric ideas in the computer (e.g., binary, ASCII, pixel attributes such as RGB).

Context of the Standard
Similar to humans, computers need a format in which to receive, interpret, and manipulate information. Computers use numeric values to store information and perform operations. Information inputted into the computer from various components must be converted into numeric values in order for the computer to use the information and perform functions. Once the function is completed, the numeric values must be converted to a form of output that the user can understand. This output may be in the form of words, images, videos, or sounds. Examples of different ways non-numeric information such as letters or colors can be expressed include the use of different protocols such as binary, ASCII, or RGB. <i>Students are not expected to apply these protocols in fifth grade.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Understand that computers use numeric values to represent non-numeric ideas. - Give an example of when numeric values can be used to represent non-numeric ideas. - Apply using numeric values to represent non-numeric ideas to in a real-world example.	Students should <i>investigate</i> these concepts: - What is a numeric value? - What are some examples of how numbers are used to represent non-numeric ideas in the computer? - Why are numbers used to represent non-numeric ideas in the computer? - Why does a computer convert input into a different format?	Students should <i>apply</i> these terms in context: - Binary - Pixel - ASCII

Impacts of Computing

5.14 The student will give examples and explain how computer science had changed the world and express how computing technologies influence, and are influenced by, cultural practices.

Context of the Standard
The needs and wants of different groups of people will have an effect on the types of computing technology they create and use. People tend to use technologies that will help facilitate the various cultural activities they engage in. In turn, the cultural needs of a group will drive development of new and more effective technologies. These new technologies could be to make communication more efficient, facilitate the sharing of ideas, automate common processes, or meet other societal demands. Increased access to the Internet has greatly increased the communication aspect of computing technology, and has also had an effect on the customs of many groups of people.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify computing technologies that have changed the world. - Explain how the technology is influenced by culture. - Explain how the culture can affect the technology. - Brainstorm solutions involving computing technology to solve a problem in your school.	Students should <i>investigate</i> these concepts: - What are examples of computing technologies that changed the world? - How has technology, like mobile phones, changed society? - How does society influence the technology that we invent? - If you could design a new computing technology, what would it do, and why?	Students should <i>apply</i> these terms in context: Internet

- 5.15 The student will evaluate and describe the positive and negative impacts of the pervasiveness of computers and computing in daily life (e.g., downloading videos and audio files, electronic appliances, wireless Internet, mobile computing devices, GPS systems, wearable computing).

Context of the Standard
The use of technology, including computers, has allowed for global communication and has revolutionized the everyday access of information, whether for business, scientific or personal use. Although there are many positive impacts in using technology, there are also times when computer use has impacted us in undesirable ways. As computer technology continues to advance and new generations of machines grow faster and have greater capabilities, the machines become more deeply fixed in daily life, magnifying both the benefits and the downside risks. Positive impacts include easy access to information, automated machinery, and fast and accurate data processing. Negative impacts include an increase in sedentary lifestyles, family and leisure interruption, and loss of privacy.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify how the use of computers and computing positively influences daily life. - Identify how the use of computers and computing negatively influences daily life. - Evaluate use of time in activities at school and at home to determine positive and negative impacts of these activities on health and wellbeing.	Students should <i>investigate</i> these concepts: - How do computing devices make your life easier? - How have computing devices made people's lives more complicated? - What are ways to limit the negative influences of computing devices? - What do you believe are good rules about technology use to make sure that we can use them wisely?	Students should <i>apply</i> these terms in context:

5.16 The student will explain social and ethical issues that relate to computing devices and networks.

Context of the Standard
People can work in different places and at different times to collaborate and share ideas when they use technologies that reach across the globe. These social interactions affect how local and global groups interact with each other. As with any social interaction, there are manners that people should use when interacting with others. The use of manners when collaborating or interacting with others through computing devices or networks is more complex since many times the communication is done without seeing the person on the other side of the communication. For example, communications should be clear and concise and should never represent the words and actions of others as your own. Care should be taken when sharing information so that the intent of the message is not misunderstood by the person on the other end of the communication. In addition, due to the anonymous nature of online communication, intimidating and inappropriate behavior in the form of cyberbullying may occur. Cyberbullying is a form of bullying that occurs when online communications are sent that are intimidating

Context of the Standard
or threatening in nature.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Describe problems that arise from computer use. • Determine solutions to common computer use issues.	Students should <i>investigate</i> these concepts: • What is cyberbullying? • How could computing technology make it easier for people to engage in negative behavior? • What should you do if you see other people using a computer to do harm to others?	Students should <i>apply</i> these terms in context: • Cyberbullying

Networking and the Internet

5.17 The student will compare and contrast the difference between a local network and a worldwide network.

Context of the Standard
A local network is a collection of computers and devices, such as printers, connected together. Many networks in a school or at home are connected using a combination of wired and wireless devices. When a computer connects to a network, it is online. Networks allow computers to share resources, such as hardware, software, data, and information. The sharing of resources saves time and money and also allows different people throughout your house or school to share resources. The Internet is a worldwide collection of networks that connects millions of businesses, government agencies, educational institutions, and individuals.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Compare and contrast the difference between a local network and a worldwide network. • Model a network at home or school showing different components (i.e. printers, computers, and server).	Students should <i>investigate</i> these concepts: • What is a network? • What is the difference between a local and a worldwide network? • What are examples of local and global networks? • What are the advantages and disadvantages of local and global networks?	Students should <i>apply</i> these terms in context: • Network • Local network • Worldwide network

Grade Six

The sixth-grade standards emphasize constructing programs and utilizing algorithms to accomplish a task. Students continue to decompose larger problems into smaller tasks and recognize the impacts of computing and computing devices. Students in sixth grade begin to understand the means of storing data as representations of real world phenomena. The accurate use of terminology as well as the responsible use of technology will continue to be built upon. The foundational understanding of computing and the use of technology will be an integral component of successful acquisition of skills across content areas.

Algorithms and Programming

- 6.1 The student will construct programs to accomplish a task as a means of creative expression or scientific exploration using a block based or text based programming language, both independently and collaboratively,
- combining control structures such as if-statements and loops; and
 - creating clearly named variables that represent different data types, including numeric and non-numeric data, and perform operations on their values.

Context of the Standard

Programs are collections of code organized in algorithms that can accomplish a variety of tasks. Programs can be developed to perform calculations, manipulate data, or simply to be creative. Programs can involve different control structures such as loops and if-statements; these control structures are blocks of programming that analyze variables within the program code to adjust and use accurate values as they change. Control structures help students develop their problem solving skills and foster computational thinking. Effective variable use, to include the use of identified variables to perform operations, makes the problem solving process easier and faster.

One central skill in learning to program is called *flow of control*. This refers to the order that commands are run by the computer. The order of the commands, or sequencing, can have dramatic impacts on whether a program runs correctly. By repeating commands the programmer has fewer lines to write, and less opportunities to make mistakes. Conditionals (if-statements) are added to a program to control whether or not commands are run. An if-statement acts as a door. If the condition is true, the door opens and commands connected to the statement are run, otherwise they are skipped. This allows programs to respond to user input and events.

In elementary school, students begin their study of programming through a focus on algorithms. They work both collaboratively

Context of the Standard

and individually to develop algorithms to reflect tasks in daily life; these algorithms become more complex as they recognize and use loops and events in the algorithms they construct. Although the use of plugged and unplugged activities is encouraged in early elementary, the expectation is that students use of block and text based programming as they progress in elementary years. Middle school continues to build on these skills as students use algorithms to build programs.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create programs which specify the order (sequence) in which instructions are executed within a block-based and/or text based program. • Appropriately apply variables in programs to meet desired outcomes. • Combine and nest If-statements and loops to create more complex programs. • Work with a partner or group to create a program.	Students should <i>investigate</i> these concepts: • What is meant by flow of control when developing programs? • What is the role of variables in program construction? • What kinds of data can be represented by variables? • What different types of data can be represented using variables? • What roles can different types of variables serve in program construction? • How does combining control structures increase efficiency in the construction of programs?	Students should <i>apply</i> these terms in context: • If statement • Loop • Control structure • Flow of control

6.2 The student will trace programs to predict outcomes and debug (correct and improve) for correctness.

Context of the Standard

During the program review stage, the design and implementation are checked for adherence to program requirements, correctness, and usability. This review could lead to changes in implementation and possibly design, which demonstrates the iterative nature of the process. If the program does not work as intended, the students should determine what changes could be made to the program in order to complete the task. These changes may include adding, deleting, rearranging, or changing a step in order to obtain the intended outcome. The process of revising a program so that it works as intended is called debugging.

One important skill students need is tracing code. Tracing means reading through commands using paper and pencil and predicting what the code will do. Students practice these skills by reviewing code written by themselves and others. This can be code the teacher creates, or through peer review.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Test and debug their own programs. • Test and debug programs written by others. • Predict the outcome or output of a program by examining the written code.	Students should <i>investigate</i> these concepts: • Why is there a need to debug programs? • How does program tracing help with the debugging process? • How can you predict the outcome of a program without running it?	Students should <i>apply</i> these terms in context: • Bug • Debug • Program tracing

6.3 The student will seek and incorporate feedback from team members and users to refine a program that meets user needs.

Context of the Standard

In middle school, students continue to develop effective group work habits across disciplines to help create products on a timetable that reflect all the group members. Programming is frequently done in a group setting. The practice of peer code review in computer science fosters effective communication, and allows students to see a variety of coding styles. The peer review process also allows students to identify common mistakes made by groupmates.

A consideration of the peer review process is determining whether the outcome of the program meets the needs of the end-user. Considerations of the end-user may include usability, accessibility, age-appropriate content, respectful language, user perspective, pronoun use, color contrast, and ease of use.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Work collaboratively with peers to review programs and adjust to meet intended outcomes. • Incorporate feedback into a program or the design of a computational artifact. • Analyze refined program to determine if it meets the intended outcomes as well as the end-user needs.	Students should <i>investigate</i> these concepts: • What are the advantages of working in a group setting? • What are strategies that help in doing collaborative work? • Why is it important to analyze feedback? • What end-user considerations must be taken into consideration when determining an outcome?	Students should <i>apply</i> these terms in context: • End-user

6.4 The student will incorporate existing code, media, and libraries into original programs, and give attribution.

Context of the Standard

When developing programs, students are able to use algorithms, code, and other media created by others. The inclusion of existing code or portions of a program into a new program allows students to speed up the development process since they don't have to create as much from scratch. The scale of imported code can range from small sections of code to large libraries containing entire functionalities. Incorporating and editing the work of others enables students to learn different approaches to writing programs and create more powerful programs. For example, when creating a game, students may incorporate portions of code that create a realistic jump movement from another person's game, and they may also import Creative Commons-licensed images to use in the background. Creative Commons is a set of intellectual property laws that govern the use of media created by others.

It is very important that, when using other peoples' creation, to give attribution to the original author. Students need to cite sources and give credit to the creators for all content they incorporate into their work.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Integrate portions of code, algorithms, and/or digital media from other sources in their own programs and websites. • Give attribution to the original creators to acknowledge their contributions.	Students should <i>investigate</i> these concepts: • What is a Creative Commons license? • Why is important to give attribution to the original creators to acknowledge their contributions?	Students should <i>apply</i> these terms in context: • Creative Commons • Plagiarism • Source • Cite

Computing Systems

6.5 The student will design projects that combine hardware and software components to collect and exchange data.

Context of the Standard

When designing programs or completing projects, students should consider what types of data might be used as input, how the data is to be stored and processed, and the output that meets the specific needs of the user. The collection and exchanging of data is used throughout the programming process. The collection and exchange of data involves input, output, storage, and processing; both software and hardware components are critical to the collection and exchange of data. Student code must be written in a specific way to handle the data's input and manipulation. These data may be simple text or numbers typed in through the keyboard. There are also data such as motion or sound that may be difficult to enter without specific sensors.

Components to collect and exchange data are constantly being developed to meet societal needs. Examples of current components include GPS, car sensors, probeware, smartwatches, and more.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Plan or sketch ideas combining hardware and software components that collect and exchange data. - Select the hardware and software components for project designs by considering factors such as functionality, cost, size, speed, accessibility, and aesthetics.	Students should <i>investigate</i> these concepts: - Why is important to know what kind of data you want to collect before you begin to design a project? - What are the different kinds of data you can collect using hardware and software?	Students should <i>apply</i> these terms in context: Sensors

Cybersecurity

6.6 The student will identify physical and digital security measures used to protect electronic information.

Context of the Standard

Students use passwords to protect access to personal data. The use of passwords and common rules for the construction of passwords was introduced in elementary school. In sixth grade, students extend their knowledge of using passwords to protect data by learning about different types of security measures. They will identify and compare physical and digital security measures and explore the reasons for protecting electronic information. Physical security measures, as the term suggests, include locks on laptops, locked server rooms, etc. Digital security measures include firewalls, encryption, and biometric tests like thumbprint readers or retinal scans, etc.

Data protection is also a significant concern when storing and transferring personal and professional data. The concept of data protection extends into a variety of security measures. These security techniques can be hardware or software based.

Suggestions for creating strong passwords include:

- Use uppercase and lowercase letters.
- Use numbers.
- Use symbols.
- Use at least 8 characters.
- Don't use words from a dictionary.
- Don't use the same password twice.
- Don't use personal information.

Essential Skills	Essential Questions	Essential Vocabulary
------------------	---------------------	----------------------

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify physical and digital security measures used to protect electronic information. - Compare advantages and disadvantages of a variety of security measures.	Students should <i>investigate</i> these concepts: - Why is there a need for strong passwords? - What are the requirements of a strong password? - What are some examples of physical security measures? - What are some examples of digital security measures?	Students should <i>apply</i> these terms in context: - Password - Firewall - Biometrics

Data and Analysis

6.7 The student will explain how binary sequences are used to represent digital data. *Exclusion: Conversions between binary and base-ten numbers are beyond the scope of these standards.*

Context of the Standard
Similar to humans, computers need a format in which to receive, interpret, and manipulate information. Computers use numeric values to store information and perform operations. Information inputted into the computer from various components must be converted into a binary sequence for processing. This conversion allows the computer to use the information and perform functions. Once all the functions and manipulation are complete, the binary sequence is translated back into a form of output that the user can understand, such as words, images, or sounds. Binary uses a sequence of 1's and 0's to represent data; each binary digit is referred to as a bit. All electronic data is binary information.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify examples of binary information. - Describe the characteristics and uses for a binary sequence. - Consider the possibilities and limitations that arise when sending binary messages.	Students should <i>investigate</i> these concepts: - What is a binary sequence? - What are some examples of binary information? - What is the significance of binary sequences in computing? - How, in basic terms, does a computer convert information from binary into a form that an end user can understand?	Students should <i>apply</i> these terms in context: Binary

6.8 The student will use computational tools to collect, organize, and clean data to make it more useful and reliable.

Context of the Standard
In order to manipulate data, create representations, and draw conclusions, data must be organized in a consistent fashion. When collecting it is likely that there will be irrelevant and/or erroneous data points in a data set which can distort the results of any analysis. Data scientists spend up to 80% of their time cleaning these irrelevant or erroneous data points. Data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. In elementary school students learn that data can be collected and stored in a variety of ways. Students will extend their data collection experience to include computational tools (e.g., surveys, spreadsheets). They will collect data electronically or convert data to an electronic format. The data will be organized and displayed through a graphical representation to make it easy to read and interpret. Removing irrelevant and/or erroneous data points is a new component to data analysis in middle school.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Collect data using a computational tool. - Organize data to make it easier to understand and use. - Transform (clean) data (remove and correct errors, unify response format, and eliminating blanks and duplicates) to process.	Students should <i>investigate</i> these concepts: - How can we collect data electronically? - What does it mean to clean data and why is it necessary? - How does data type affect how it should be organized and displayed? - What are our goals and expectations for data cleaning?	Students should <i>apply</i> these terms in context: - Data - Data cleaning

6.9 The student will explain the insight and knowledge gained from digitally processed data by using appropriate visualizations.

Context of the Standard
Data visualization is a general term that describes any effort to help people understand the significance of data by placing it in a visual context. Data visualization can allow for insight that would be more difficult than by looking at the raw data. Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier with data visualization models. Visualization model may include: circle graphs, bar graph, line plots, pictographs, Venn Diagram, etc. Software applications can also be used to assist in visualizing data. In sixth grade, students will evaluate visual representations of data to draw conclusions and determine the best visualization for various types of data. This is an opportunity to explore ways that data and graphical representations can be intentionally or unintentionally misleading.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify different types of visual representations of data. - Compare various visual representations and identify when each should be used. - Use data to create a visual artifact that answers a research question and communicates results and conclusions.	Students should <i>investigate</i> these concepts: - What are some ways we can visually represent data? - How can visualizations be used to make data easier to interpret and answer questions? - How can we select the best graph for a data set?	Students should <i>apply</i> these terms in context:

6.10 The student will use models and simulations to formulate, refine, and test hypotheses.

Context of the Standard
Models and simulations allow students to represent systems that are too large, too small, or otherwise difficult to study in a classroom setting. Computer models and simulations are necessary when the data sets are too large for human evaluation. There are limitless applications for computer models and simulations. A simulation is a virtual representation of a process that reflects how a real physical situation would most likely happen. Simulations are created using models that were developed based on data. Some examples of simulations are performance of cars in various weather conditions, rocket launches, growth of a population, and the ability of a vaccine to fight a disease. Models are also constructed to test hypotheses. The data generated from the model are used to evaluate these models are accurate, make adjustments, and make conclusions. Models and simulations need to be tested for accuracy and refined as necessary. Students entering sixth grade have interpreted and constructed models to explain concepts in both computer science and science. In sixth grade, students will use models to represent systems and simulations to represent processes and to support hypotheses.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create a hypothesis and test it using a computational model or simulation.	Students should <i>investigate</i> these concepts: • What are some systems or situations that can be modeled or simulated? • What is the difference between a model and a simulation? • How can we use data to improve a model or simulation?	Students should <i>apply</i> these terms in context: • Model • Simulation

Impacts of Computing

6.11 The student will explain how computing has impacted innovations in other fields.

Context of the Standard
Almost every industry has been affected by computing technology. Manufacturing, retail, finance, healthcare, and transportation are examples of fields that have been drastically changed by the incorporation of computers. Technologies that have influenced innovation include, but are not limited to: sensors, GPS, digital forensics, cryptocurrency, cybersecurity, databases, the cloud, the Internet, Virtual Reality, 3D printing, self-driving vehicles, etc. Computing has also impacted innovation. Examples of innovation that have affected different career fields include: providing the opportunity for faster production, access and quantities of data and information, advances our ability in communication and collaboration, and self-publishing. In sixth grade, students begin to understand the impact of computer science in multiple career fields. Computer science has provided opportunities for these fields to grow and evolve.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Explain how computing impacts innovation.	Students should <i>investigate</i> these concepts: How does computing impact innovation in non-computing oriented fields?	Students should <i>apply</i> these terms in context:

6.12 The student will explore careers related to data.

Context of the Standard
There are many options for careers that utilize data collection and analysis. In sixth grade, students will explore various aspects of careers such as work expectations, pay rate, and required education. The use of computer skills is not limited to computer fields; many fields require workers to use computing devices, analyze data, use models and simulations, and use different types of software and hardware when completing tasks. Careers in data science are in high demand. Examples of data-related careers include data analyst, data scientist, data engineer, data architect. Current information on education, pay, and employment projections can be found through the U.S.Bureau of Labor Statistics (https://www.bls.gov/emp/).

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Research careers in computing and non-computing fields.	Explore a <i>career field</i> to answer the following questions: - What are potential career pathways related to data? - What are the pros and cons to the field? - What is the salary? Work environment? - Will there be a strong demand for that career in the future? - What are the trends that could influence the demand in the future? - Will there be local jobs in that field? Where are most jobs in this field? - What skills, education, languages and experience are required in that field?	Students should <i>apply</i> these terms in context:

Networking and the Internet

- 6.13 The student will explain why the speed of data transmission across the Internet can vary depending on the type of data being transmitted.

Context of the Standard
Speed of data transmission is how fast data can be sent from one device to another. There are a variety of factors to consider when evaluating transmission speed, including the size of the file or artifact, available bandwidth (maximum rate of data transfer), and concurrent uploading or downloading activity taking place on a given network. Examples of different types of data: .jpeg, pdf, Word document, Google Doc, and video.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain different types of data and their speed when going from one device to another. • Describe factors that may affect the speed of data transmission.	Students should <i>investigate</i> these concepts: • Why is important to know different types of data and their rates of speed? • Besides the size of the data type, what else can affect the speed of transmission?	Students should <i>apply</i> these terms in context:

Grade Seven

The seventh-grade standards emphasize constructing programs and utilizing algorithms to accomplish a task. Students continue to decompose larger problems into smaller tasks and recognize the impacts of computing and computing devices. Students in seventh grade explore processing data as well as its transmission over networks. The accurate use of terminology as well as the responsible use of technology will continue to be built upon. The foundational understanding of computing and the use of technology will be an integral component of successful acquisition of skills across content areas.

Algorithms and Programming

- 7.1 The student will construct programs to accomplish a task as a means of creative expression or scientific exploration using a block based or text based programming language, both independently and collaboratively,
- combining control structures such as if-statements and loops including compound conditionals; and
 - creating clearly named variables that represent different data types, including numeric and non-numeric data, and perform operations on their values.

Context of the Standard

Programs are collections of code organized in algorithms that can accomplish a variety of tasks. Programs can be developed to perform calculations, manipulate data, or simply to be creative. Programs can involve different control structures such as loops and if-statements; these control structures are blocks of programming that analyze variables within the program code to adjust and use accurate values as they change. Control structures help students develop their problem solving skills and foster computational thinking. Effective variable use, to include the use of identified variables to perform operations, makes the problem solving process easier and faster.

One central skill in learning to program is called *flow of control*. This refers to the order that commands are run by the computer. The order of the commands, or sequencing, can have dramatic impacts on whether a program runs correctly. By repeating commands the programmer has fewer lines to write, and less opportunities to make mistakes. Conditionals (if-statements) are added to a program to control whether or not commands are run. An if-statement acts as a door. If the condition is true, the door opens and commands connected to the statement are run, otherwise they are skipped. This allows programs to respond to user input and events.

In seventh grade, students should add compound conditionals to their programs. A compound condition specifies a combination of

Context of the Standard
other conditions, allowing for two or more conditions to be tested in a single statement, such as “if-and” and “if-or.”

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create programs that include control structures or compound conditionals using block or text-based applications. • Work with a partner or group to create a program. • Appropriately apply variables in programs to meet desired outcomes. • Combine and nest If-statements and loops to create more complex programs.	Students should <i>investigate</i> these concepts: • How does combining control structures increase efficiency in the construction of programs? • What roles can different types of variables serve in program construction? • How do compound conditionals reduce the amount of code needed in a program?	Students should <i>apply</i> these terms in context: • If statements • Compound conditionals

7.2 The student will document programs to make them easier to follow, test, and debug.

Context of the Standard
Proper documentation helps keep track of all aspects of an application and improves on the quality of the program. Documentation allows programmers and reviewers to understand the intention behind a particular portion of code. Documentation comments are formatted to describe portions of the program but are not integrated into the program flow. Often, a programmer will write code and not return to it for lengths of time; documentation can be used to remind them of the intention behind their code and is very useful in

Context of the Standard
debugging, as it signals how the program should be acting.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Embed proper documentation into individual and group designed programs. • Use documentation when identifying personal work. • Interpret documentation within existing programs.	Students should <i>investigate</i> these concepts: • What does it mean to document programs? • How does documentation assist in the testing and debugging process?	Students should <i>apply</i> these terms in context: • Debugging • Documentation

7.3 The student will distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

Context of the Standard
Development of computational artifacts (program, image, audio, etc.) is often a collaborative effort. Working well in a group requires students to practice strategies such as delegation of responsibility, peer-review, and assigning self-imposed deadlines. These deadlines are used to construct a timeline that informs group participants of responsibilities and expectations when completing a project such as the development of a computational artifact. The creation of appropriate and realistic timelines is difficult to many students, particularly when working in groups. Students may need teacher guidance and scaffolding in the timeline construction process as well as regular reminders to look back and revise timelines as needed throughout the process.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Construct a timeline for a project or the development of a program that reflects the expectations and responsibilities of the members of a group. - Use and revise a timeline when completing a group project.	Students should <i>investigate</i> these concepts: - How do timelines help in the completion of a project or in the completion of a program? - What tasks or expectations should be included in a timeline? - How does the use of a timeline facilitate group work?	Students should <i>apply</i> these terms in context:

7.4 The student will decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

Context of the Standard		
Decomposition breaks problems down into smaller pieces (subproblems). Problem decomposition allows for complex tasks to be solved in manageable chunks. Once a program has been broken down, it enables different group members to work on different parts at the same time. Decomposition can occur in the planning and design stage, allowing for the separate subprograms to be more easily tackled, or when debugging a long and complex program. Students will practice this skill on programs they wrote and written by others.		
Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Breakdown code into parts to facilitate creation of a program. - Use decomposition in all parts of the program development process (i.e. plan, design, and review).	Students should <i>investigate</i> these concepts: - Why does breaking a problem down into smaller problems make the overall task easier? - How does decomposing a program into subproblem help programmers when debugging a program? - How can decomposition be applied throughout the development process? - Why would using subproblems in a program be thought of as a time saving measure?	Students should <i>apply</i> these terms in context: - Decompose - Subprogram

Computing Systems

7.5 The student will describe how the Internet connects devices and networks all over the world. *Exclusion: Specific devices used to implement the Internet are beyond the scope of these standards.*

Context of the Standard
The Internet is a global network of computers. All computer devices (including PCs, laptops, games consoles and smartphones) that are connected to the Internet form part of this network. Some of the computers in this network are called web servers. A web server is a computer which holds websites for other computers linked to the Internet to access. Holding a website is known as ‘hosting’. A web server may host one or many websites and webpages. Sending information to a web server is known as uploading. Receiving information from a web server is known as downloading.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Describe how the Internet connects devices and networks globally.	Students should <i>investigate</i> these concepts: What effect has the Internet had on global communication?	Students should <i>apply</i> these terms in context: Internet

Cybersecurity

7.6 The student will describe how physical and digital security measures protect electronic information.

Context of the Standard
Students will extend their knowledge of the need for physical and digital security measures to understanding and explaining that both types of measures protect our data and personal information. Students can explore different types of security measures and understand how each one protects us in different ways from different types of electronic breaches and attacks. Physical security measures include locking rooms with devices, badges, fingerprints, security cameras, paper shredding, etc. Digital security measures include firewalls, anti-virus software, strong passwords, anti-spyware, etc. In seventh grade, students will take this a step further from sixth in describing how these security measures can be applied in a setting.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Understand why data security is necessary. - Distinguish between physical and	Students should <i>investigate</i> these concepts: - Why is data security necessary? - How do physical security measures protect us?	Students should <i>apply</i> these terms in context: - Password - Firewall

Essential Skills	Essential Questions	Essential Vocabulary
digital security measures. - Explain how and why physical security measures protect us. - Apply examples of physical and digital security measures in a setting.	- How do digital security measures protect us? - Why should a student implement security measures when working with digital information?	

7.7 The student will identify existing cybersecurity concerns associated with Internet use and Internet-based systems and potential options to address these issues.

Context of the Standard
Cybersecurity is a growing industry in the world as more of our personal, financial, government, and military information is transmitted electronically and housed in data centers. Today's cybersecurity concerns are varied and complex and include actions like breaching of private information from governmental or business sources, the use of ransomware to withhold access to a computing system, or global threats from the hacking of confidential data by hostile entities. Cybersecurity encompasses many aspects of data protection in our society. Some examples are the protection of privacy, military information, credit card and banking information, social media profiles, etc. Measures in order to prevent loss of data or other cyberthreats physical and digital security measures must be practiced at all time. These can include logging out of accounts on public computers, strong passwords, and those outlined in Standard 7.6. In seventh grade, students will begin to understand how data is vulnerable and what steps are taken to protect it.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain the necessity of cybersecurity. • Explain risks associated with the use of public devices and unsecure WiFi. • Describe the role of physical and digital security measures in cybersecurity.	Students should <i>investigate</i> these concepts: • What are some examples of recent breaches in cybersecurity? • What are the risks of using public devices and public WiFi connections? • How can we protect our data on the Internet?	Students should <i>apply</i> these terms in context: • Cybersecurity • Identity Theft

Data and Analysis

- 7.8 The student will discuss the correctness of a model representing a system by comparing the model's generated results with data that were observed in the system being modeled.

Context of the Standard
Models and simulations allow students to represent systems that are too large, too small, or otherwise difficult to study in a classroom setting. Computer models and simulations are necessary when the data sets are too large for human evaluation. There are limitless applications for computer models and simulations. A simulation is a virtual representation of a process that reflects how a real physical situation would most likely happen. Simulations are created using models that were developed based on data. Some examples of simulations are performance of cars in various weather conditions, rocket launches, growth of a population, and the ability of a vaccine to fight a disease. Models are also constructed to test hypotheses. The data generated from the model are used to evaluate whether the models are accurate, to make adjustments, and draw conclusions. Models and simulations need to be tested for accuracy and refined as necessary. In sixth grade, students used models to represent systems and simulations to represent processes and to support hypotheses. In seventh grade, students will use real-world data to determine if the models and simulations are accurate and adjust the

Context of the Standard
models/simulations to improve accuracy.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Compare output of the model with observed data. • Determine if a model accurately represents a system. • Identify components of a model and discuss how each component affects the generated results.	Students should <i>investigate</i> these concepts: • How can we use model to make predictions? • What are some systems that can be modeled with a computer program? • How can we determine if a model is accurate? • What are some components of a model that can be modified?	Students should <i>apply</i> these terms in context: • Data • Parameter • Output • Input

7.9 The student will refine computational models based on the data they have generated.

Context of the Standard
Models are used in many disciplines to test and isolate specific systems while eliminating some of the complexity that can make results muddy or unclear. Because of this simplification, models need to be tested against real-world data. By designing a model and refining it based upon these data, researchers and designers can gain certainty that the model is valid and its results are meaningful. Students will create and refine computational models to simulate a system. Through this process they will gain insight into how mathematics, science, and computer science are used to understand our world.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Use actual data to test the model and compare the results. • Modify a model to more accurately represent a system when the model does not match the data. • Analyze a model to determine if the data generated reflects the physical phenomena.	Students should <i>investigate</i> these concepts: • What data can be used to test the model? • How can we determine if the model is accurate? • What changes can be made to the model to increase the accuracy?	Students should <i>apply</i> these terms in context: • Computational thinking • Command • Variable

Impacts of Computing

7.10 The student will explain how advances in technology have contributed to Virginia's prosperity and role in the global economy.

Context of the Standard		
Students will explore a variety of technology fields in which Virginia has acted as a leader. Virginia's transportation system, which includes highways, railroads, air transportation, and shipping, moves raw materials to factories and finished products to markets. Virginia has a large number of communications and other technology industries. Virginia exports agricultural and manufactured products, including tobacco, poultry, coal, and large ships. Advances in transportation, communications, and technology have facilitated migration and led to economic development in Virginia. The students will investigate these industries' history, purpose, and how Virginia emerged in a leadership role.		
Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify advances in technology in Virginia. - Explain contributions based on technology to Virginia's prosperity. - Explain the role of technology in Virginia within the global economy.	Students should <i>investigate</i> these concepts: - What role have technological innovations played in Virginia's economy? - What technology is generated in Virginia that impacts the global economy?	Students should <i>apply</i> these terms in context: - Technology - Global economy - Goods and services - Model - Simulation

- 7.11 The student will describe the development of new technologies in communication, entertainment, and business and their impact on American life.

Context of the Standard
Since before the Industrial Revolution, advances in technology have been influenced by and have influenced society. As the pace of technological advance increases and becomes more highly specified, these influences can be seen in all aspects of American life - how people interact, how data is stored, the importance of data security, financial transactions, and many more. Students will investigate new and developing technologies. Communication industry technological advancement may include but are not limited to: social media, machine learning, Internet of things, driverless cars, security and privacy, networking, branding, funnel marketing. Business industry technological advancements may include but are not limited to: traceability and safety software, database advancements, customer scheduling, big data and machine learning, automation, and augmented reality. Entertainment industry technological advancement may include but are not limited to: downloading digital music, video streaming, ticket sales, marketing and receiving, licensing, voice technology, and holograms. Students will examine the cause and effect nature between these technologies and various aspects of society, economy, and culture.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify and describe new technologies in communication industry. - Identify and describe new technologies in entertainment industry. - Identify and describe new technologies in business. - Analyze the impact of new technologies on American life.	Students should <i>investigate</i> these concepts: - What are examples of new technologies in the communication industry? Entertainment industry? Business industry? - How have new technologies in the _____ industry impacted American life? - Have the impact of new technologies been positive? Negative? Helpful? Useful?	Students should <i>apply</i> these terms in context: - Culture - Funnel marketing - Automation - Technology - Data

7.12 The student will explore careers related to the Internet.

Context of the Standard
There are many options for careers that utilize data collection and analysis. Students will explore various aspects of these careers such as work done, pay rate, and education needed. The use of computer skills is not limited to computer fields; many fields require workers to use computing devices, analyze data, use models and simulations, and use different types of software and hardware when completing tasks. Careers in data science are in high demand; companies are finding it difficult to find workers in this field. Several computer careers are outlined below. - A data analyst's role is to collect, process, and perform statistical data analyses with the goal of helping companies make better business decisions. - A data scientist possesses a combination of analytic, machine learning, data mining, and statistical skills in addition to experience with algorithms and coding.

Context of the Standard
• Data engineers are the designers, builders, and managers of the information or big data infrastructure. • A data architect creates the blueprints for data management systems to integrate, centralize, protect, and maintain data sources. Current information on education, pay, and employment projections can be found through the U.S.Bureau of Labor Statistics (https://www.bls.gov/emp/).

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Research careers in computing and non-computing fields.	Explore a <i>career field</i> to answer the following questions: • What are potential career pathways related to the Internet? • What do we mean when we say “related to the Internet”? • What are the pros and cons to the field? • What is the salary? Work environment? • Will there be a strong demand for that career in the future? • What are the trends that could influence the demand in the future? • Will there be local jobs in that field? Where are most jobs in this field? • What skills, education, languages and experience are required in that field?	Students should <i>apply</i> these terms in context: • Pay rate • Career Pathway • Internet • Workplace readiness skill • Salary

Networking and the Internet

7.13 The student will outline the advantages and disadvantages of transmitting information over the Internet, including speed, reliability, cost, and security.

Context of the Standard

There are a variety of factors to consider when evaluating data transmission over the Internet. Compared to physically sending documents or media, the Internet allows for relatively fast transfer at a low cost. However, sometimes the integrity of a file can be compromised if there is an error in transmission. There is also always a possibility that information can be hacked if it is sent via unsecured channels. Students will explore the different components to sending these messages, such as the medium for data transmission, and make determination of the advantages and disadvantages.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain different types of data and their speed when going from one device to another. • Compare the speed and reliability of various data transmission media. • Describe the advantages and disadvantages of transporting information over the Internet.	Students should <i>investigate</i> these concepts: • What are advantages in using the Internet to transmit information? • What are disadvantages in using the Internet to transmit information?	Students should <i>apply</i> these terms in context: • Internet speed • Data transmission • Reliability • Internet • IP address

- 7.14 The student will explain why protocols are necessary in data transmission. Model the role of protocols in transmitting data across networks and the Internet.

Context of the Standard
When computers communicate over networks, they break down messages into small chunks called packets. In order to reliably communicate among networked computing devices, all the devices need to create and interpret these packets based on a universally agreed-upon set of rules. Once these rules are in place, no one has to approve a new website or oversee additions to the network. The rules that govern communication among computing devices are called a protocol. Protocols allow computing devices to send and receive messages using the same series of steps for every instance of communication, whether they are sending messages across the room or across the planet. Examples of these include TCP/IP (Transmission Control Protocol/Internet Protocol), FTP (File Transfer Protocol), Hypertext Transfer Protocol (HTTP), and others. <i>Students are not expected to identify specific rules governing the functionality of transmission protocol in seventh grade. They should describe the overall purpose of a protocol in general terms.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify the purpose of a transmission protocol. • Describe the function of a simple transmission protocol.	Students should <i>investigate</i> these concepts: • How can you send something to someone you do not know? • Why do people need a protocol to send data back and forth?	Students should <i>apply</i> these terms in context: • Data transmission • Protocols • Network • Internet

- 7.15 The student will model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.

Context of the Standard
Due to the constraints of networks in handling large files, when data in many forms has to be sent, they are broken down into smaller structures of data - called packets - before transmission. Packets are reassembled to the original data chunk once they reach their destination according to the appropriate protocol. This process allows large items to be sent at relatively quick speeds and, due to built in redundancies of packets, also solves problems related to corruption of files occurring at some stage of sending. Students will model how these messages are broken up, transmitted, and put back together to maximize data integrity and efficiency.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain the purpose of data packets. • Model a process of breaking a message into packets and message reassembly.	Students should <i>investigate</i> these concepts: • How is information sent from one computer to another over the Internet? • What happens if a message is too large to carry all at once?	Students should <i>apply</i> these terms in context: • Data integrity • Data efficiency • Computer network • Constraints • Network

Grade Eight

The eighth-grade standards emphasize constructing programs and utilizing algorithms to accomplish a task. Students continue to decompose larger problems into smaller tasks and recognize the impacts of computing and computing devices. Students in eighth grade continue to work with data including how it can be vulnerable and how it can be protected. The accurate use of terminology as well as the responsible use of technology will continue to be built upon. The foundational understanding of computing and the use of technology will be an integral component of successful acquisition of skills across content areas.

Algorithms and Programming

- 8.1 The student will construct programs to accomplish a task as a means of creative expression or scientific exploration using a block based or text based programming language, both independently and collaboratively,
- combining control structures such as if-statements and loops including nested conditionals and loops;
 - Using clearly named variables that represent different data types, including numeric and non-numeric data, and perform operations on their values; and
 - Create procedures with parameters.

Context of the Standard

Programs are collections of code organized in algorithms that can accomplish a variety of tasks. Programs can be developed to perform calculations, manipulate data, or simply to be creative. Programs can involve different control structures such as loops and if-statements; these control structures are blocks of programming that analyze variables within the program code to adjust and use accurate values as they change. Control structures help students develop their problem solving skills and foster computational thinking. Effective variable use including naming conventions, makes the problem solving process easier and faster.

In elementary school, students begin their study of programming through a focus on algorithms. They work both collaboratively and individually to develop algorithms to reflect tasks in daily life; these algorithms become more complex as they recognize and use loops and events in the algorithms they construct. Although the use of plugged and unplugged activities is encouraged in early elementary, the expectation is that students use of block and text based programming as they progress in elementary years.

In seventh grade, students added compound conditionals to their programs. A compound condition specifies a combination of other conditions, allowing for two or more conditions to be tested in a single statement, such as “if-and” and “if-or.” In eighth grade, students begin to use nested control structures. Placing a structure such as a loop within another loop allows for a greater level of

Context of the Standard

complexity in the program's function. These often substitute for what would have been very large sections of repeated code. Procedures are named sections of code that allow a programmer to call it from multiple locations and repeat the functionality. Parameters offer an increased level of flexibility in these procedures by passing in additional information.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create programs which specify the order (sequence) in which instructions are executed within a block-based and/or text based program. • Combine and nest If-statements and loops to create more complex programs. • Write programs to accomplish tasks. • Use if statements and loops to complete a task. • Define procedures with parameters in program construction.	Students should <i>investigate</i> these concepts: • How can students use if or else statements to control programs? • How can procedures be called from various locations in a program? • What are the advantages of nesting a conditional statement within another conditional statement?	Students should <i>apply</i> these terms in context: • Pseudocode • Loop • If-else statement • Nesting

8.2 The student will systematically test and refine programs using a range of test cases.

Context of the Standard
As part of the iterative design process, programs should be reviewed and tested to determine if the design goal is met through the generation of specific output. Testing programs using a range of test cases is a necessary step for assessing program correctness. A test case is a single input with an expected output to test the correctness of a program. Programmers need to test a program to make sure it works for all the possible values received as input. This can include values within the range of expected values as well as those outside the given range. Often, programmers will use “edge cases” - parameters that test the extremes of a scenario - to test their programs, ensuring that they will work with even the largest and smallest inputs. In beta test situations programmers use random people to test a program to find “bugs.” Programmers record “versions” of programs as they update and improve their code.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify an appropriate range of test cases to use with a program. • Use a range of test cases and test a program for correctness. • Refine a program based on results from test cases.	Students should <i>investigate</i> these concepts: • Why is there a need to debug? • Why is it important to test a variety of test cases? • How does testing the program with a wide range of values help confirm its effectiveness?	Students should <i>apply</i> these terms in context: • Debugging • Command

- 8.3 The student will explain how effective communication between participants is required for successful collaboration when developing programs.

Context of the Standard
Because of the highly collaborative nature of programming projects, team members need to employ effective communication strategies. In order to avoid redundant work or gaps in the necessary code, all groups need to have open communication to help standardize the flow of the program. In order to develop a program, tasks can be broken down and created in modules by different groups of students, which requires clear and consistent communication between these groups. Additionally, the peer review fosters effective communication, and helps students see a variety of coding styles as well as identify common mistakes. Program documentation allows for writing that does not affect the flow of a program. This writing is intended to clarify the purpose of a section of code, note a troublesome section for review, or otherwise communicate between group members. In 8th grade, students should be employing these effective communication strategies as well as explain why these techniques are essential for successful programming.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Work collaboratively to produce a computational artifact. • Work with peers to write a program. • Use proper code documentation. • Communicate effectively to solve a problem.	Students should <i>investigate</i> these concepts: • What are examples of communication skills that assist in group programming? • How can students use effective communication skills to solve common mistakes in programming?	Students should <i>apply</i> these terms in context: • Documentation

8.4 The student will use flowcharts and/or pseudocode to address complex problems as algorithms.

Context of the Standard

In computer science, the development of programs uses an iterative design process involving design, implementation (programming), and review (debugging) until the program runs correctly. The design stage occurs before beginning to program. The planning stage is when the programmers gather information about the problem and sketch out a solution. This design process may include the use of pseudocode - writing out the steps of a program in English to make sure the flow of control and logic make sense. Flowcharts are another tool programmers may use when designing an algorithm or computer program. The flowchart outlines the steps that are needed in the development of an algorithm or program.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Breakdown code into parts to enable creation of a program. • Design an algorithm using a planning tool. • Review and revise a plan to better fit the needs of a task.	Students should <i>investigate</i> these concepts: • How does pseudocode help programmers organize their thoughts?	Students should <i>apply</i> these terms in context: • Flowchart • Pseudocode

Computing Systems

- 8.5 The student will, using the elements of computing devices such as primary memory, secondary storage, processor, input and output devices, and network connectivity, analyze the advantages and limitations of a given computing system.

Context of the Standard

Computing devices can have large variation in their functionality, speed, processing ability, data storage capacity, network speed, and other such details. This variability can be related to the hardware components or the choice of software that is being used. These choices are affected by the intended use of a device - a computer that is set up for gaming will have different hardware and software capabilities and limitations than one that is designed for the purposes of data manipulation or architectural renderings.

In 8th Grade, students will examine these choices and analyze the pros and cons of different computing systems.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Understand the components of a computer. • Understand how the components of a computer work and interact with each other. • Understand how the components of a computer system are impacted by its primary task.	Students should <i>investigate</i> these concepts: • What are the components of a computing device? • How do the components of a computer interact with each other? • What are the advantages and disadvantages computing systems create individually? Locally? Globally? • What features would be best for a gaming/digital art/data science computer	Students should <i>apply</i> these terms in context: • Motherboard • CPU • GPU • RAM • NIC

Cybersecurity

8.6 The student will evaluate physical and digital security measures used to protect electronic information.

Context of the Standard

Students will extend their knowledge of the need for physical and digital security measures to understanding and explaining that both types of measures protect our data and personal information. Students can explore different types of security measures and understand how each one protects us in different ways from different types of electronic breaches and attacks. Physical security measures include locking rooms with devices, badges, fingerprints, security cameras, paper shredding, etc. Digital security measures include firewalls, anti-virus software, strong passwords, anti-spyware, etc.

In eighth grade, students will analyze the advantages and disadvantages of different physical and digital security systems.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Distinguish between physical and digital security measures. • Identify examples of physical and digital security measures. • Understand why data security is necessary. • Explain how security measures protect electronic information. • Evaluate the physical and digital measures in existing computing system setups and make recommendations on improvements.	Students should <i>investigate</i> these concepts: • Why is data security necessary? • How do physical security measures protect us? • How do digital security measures protect us? • Why should a student implement security measures when working with digital information?	Students should <i>apply</i> these terms in context: • Digital footprint • Hacking

- 8.7 The student will identify impacts of hacking, ransomware, scams, fake vulnerability scans, and the ethical and legal concerns involved. *Exclusion: Students do not need to implement solutions.*

Context of the Standard
Students will identify issues and impacts associated with electronic crimes such as hacking, phishing, identity and password theft, ransomware, scams, etc. Hacking is an attempt to exploit a computer system or a private network inside a computer. Ransomware is malicious software that encrypts a computer's files unless its owner pays a ransom. It is the unauthorized access to or control over computer network security systems for some illicit purpose. This discussion will include the ethical and legal concerns that have arisen as more sensitive data is stored and transmitted electronically. Topics may include the advantages and disadvantages of the Internet of Things (IoT).

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify cybersecurity concerns. • Identify reasons data must be protected. • Identify risks of using public devices and WiFi. • Identify options to reduce the risks of Internet use. • Explain the consequences of misuse of electronic information. • Understand the legal and ethical implications of crimes involving stealing/distributing/using electronic data.	Students should <i>investigate</i> these concepts: • What are some current cybersecurity concerns? • What are the risks of using public devices and public wi-fi connections? • How can we protect our data on the Internet? • What are the consequences of a data breach or misuse of information? • What is hacking? • Why do hacking and other digital exploits occur?	Students should <i>apply</i> these terms in context: • Cybersecurity • Internet • Identity theft • IP address • URL

Data and Analysis

8.8 The student will

- a. explain the difference between a model and a simulation, and
- b. create computational models to conduct simulations.

Context of the Standard

Students will understand and explain the difference between a model and a simulation. Modelling means creating a physical replica or equations of a situation or activity. A simulation is a virtual representation of a process that reflects how a real physical situation would most likely happen. Simulations are created using models that were developed based on data. Programmers will create computational models to conduct simulations of familiar systems. Computational models can be created in pseudocode (written) or equations, in block languages (e.g., Scratch), or in text-based languages (e.g., Java, Python). There are also content specific tools for modeling such as MatLab for mathematics. Some examples of computation models might be modeling and simulating a bungee jump, investigating the temperature of melting ice, or investigating the forces and attractions in the states of matter.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify examples of models and explain how a model can represent a system. • Determine the best tool to create the model. • Identify the essential components of the model. • Use data to create an accurate model. • Compare models and simulations. • Identify examples of simulations and provide examples of possible	Students should <i>investigate</i> these concepts: • What are some examples of systems we can model? • What are some examples of simulations? • How do models differ from simulations? • What are some tools for creating computational models? • What components do we need to consider for our model? • What will be the input and the output of our model?	Students should <i>apply</i> these terms in context:

Essential Skills	Essential Questions	Essential Vocabulary
simulations. Create a computational model to provide simulated data.		

Impacts of Computing

8.9 The student will describe tradeoffs between allowing information to be public, and keeping information private.

Context of the Standard
The Fourth Amendment of the US Constitution protects every American’s right to privacy. However, this amendment was written long before the advent of the Internet, where personal information is easily accessed and, in some cases, exploited. Modern tech companies, such as Facebook, Google, Amazon, and others, mine users’ personal data and online habits to better advertise content and products towards individuals. Students will explore what information they think should be publicly available. Students discuss the benefits and drawbacks to keeping information private when compared to public release.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify and describe what information is appropriate, safe and responsible to share publicly. - Identify and describe what information is appropriate, safe and responsible to keep private. - Analyze the tradeoffs between	Students should <i>investigate</i> these concepts: - What kind of personal information is regularly collected by web-based businesses? - What value is your personal information to another person or business? - What information is appropriate to be public? Private?	Students should <i>apply</i> these terms in context: - Privacy - Public domain - Private sector

Essential Skills	Essential Questions	Essential Vocabulary
allowing access to private and public information.	Why should we allow certain information to be public vs. private? What are the consequences?	

8.10 The student will evaluate online and print sources for appropriateness and credibility.

Context of the Standard
While the Internet has made research much easier than in the past, this access has also allowed for non-vetted and factually inaccurate sources to be presented as equal in validity to more legitimately reviewed and researched outlets. Students will find sources that pertain to a particular topic and evaluate them for inclusion in their work. Students need to assess multiple aspects of a source to test its credibility. Criteria such as those listed in the Currency, Relevancy, Authority, Accuracy, and Purpose (CRAAP) Model should be employed when evaluating online resources.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify the author of a print and an online source. - Assess the appropriateness of various online resources.	Students should <i>investigate</i> these concepts: - How can you identify a valid source on the Internet? - Do Internet providers have a responsibility to control the spread of false or misleading information? If they do, what are some potential downsides to this action? - What criteria should be used to determine if a website is a valid source	Students should <i>apply</i> these terms in context:

Essential Skills	Essential Questions	Essential Vocabulary
	of information?	

8.11 The student will discuss the social impacts and ethical considerations associated with the field of cybersecurity.

Context of the Standard
Cybersecurity is a growing industry in the world as more of our personal, financial, government, and military information is transmitted electronically and housed in data centers. Today's cybersecurity concerns are varied and complex and include actions like breaching of private information from governmental or business sources, the use of ransomware to withhold access to a computing system, or global threats from the hacking of confidential data by hostile entities. Cybersecurity encompasses many aspects of data protection in our society. Some examples are the protection of individual privacy, military information, credit card and banking information, and social media profiles. In order to prevent loss of data, identity theft, or other cyberthreats physical and digital security measures must be practiced at all times as described in Standard 8.7. In eighth grade, students will weigh the advantages of cybersecurity in protecting individuals and systems against potential disadvantages of the over-restriction of content and delivery.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Understand the importance of encryption. - Understand the importance of anti-virus software. - Explain the need of cybersecurity in protecting computing systems.	Students should <i>investigate</i> these concepts: - How should a list of passwords be protected? - What are examples, globally, of societies that restrict access to information? - Does restricting access to information	Students should <i>apply</i> these terms in context: - Virus - Trojan horse - Encryption

Essential Skills	Essential Questions	Essential Vocabulary
	benefit or harm a society?	

8.12 The student will explore careers related to the field of cybersecurity.

Context of the Standard
Due to the rise in cybersecurity threats and the increasing value of information and resources that are potentially vulnerable to these threats, the career field in cybersecurity has grown tremendously over the past period of time. Examples of these kinds of jobs include security software developer, information security analyst, ethical hacker (white hat), computer forensics analyst, and many others. In eighth grade, Students will explore various aspects of these careers such as work done, pay rate, and education needed. Current information on education, pay, and employment projections can be found through the U.S.Bureau of Labor Statistics (https://www.bls.gov/emp/).

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify cybersecurity jobs available today - Explain related training needed prior to working in a cybersecurity field.	Explore a <i>career field</i> to answer the following questions: - What are the different cybersecurity jobs available? - What education is needed to work in a cybersecurity field? - What do jobs in the various cybersecurity fields pay? - What kind of work that is usually done in an office can be done from home using the Internet?	Students should <i>apply</i> these terms in context:

Networking and the Internet

8.13 The student will identify existing cybersecurity concerns associated with Internet use and Internet-based systems and potential options to address these issues.

Context of the Standard
Cybersecurity risks can take many forms and have many consequences. The consequences of hacking can be serious and depend on what level of access the hackers have achieved. Examples of cybersecurity breaches include the theft of social security and credit information from Experian and the hack of Sony Pictures by the North Korean government. Computer viruses (scripts that may be hidden in existing files or programs) and other malware (malicious software installed on a computer) generally have two goals: to propagate from system to system and to perform some action on each system they infect. Similarly, denial-of-service attacks can cripple online business for long periods of time, which can greatly affect a business's financials. In eighth grade, students will explore possible risks to the data involved in their use of the Internet. They will also explore the risks and cyber threats for which companies plan and prepare.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Describe what may happen when a hacker gains access to a computer. • Explain what measures can be taken to prevent hacking. • Describe what happens to a website or server during a denial-of service attack.	Students should <i>investigate</i> these concepts: • How can hackers damage a computer? • What are the different types of malware? • What happens when a phisher steals one's personal information? • What can happen to a website or server during a denial-of-service attack?	Students should <i>apply</i> these terms in context: • Hacker • Malware • Denial-of-Service attack • Phishing

Middle School Computer Science Elective (MSCSE) Standards

The standards below outline the content for a flexible elective course with optional modules for 6-week, 9-week, 18-week, or 36-week implementations. These standards build on the concepts of computer science developed in prior grade levels and in the integrated standards for middle school students. Teachers are encouraged to select programming languages and environments, problems, challenges, and activities that are appropriate for their students to successfully meet the objectives of the standards.

The content for the initial 6-week module has an emphasis on computer programming. Students will review and build on skills developed throughout elementary school. Teachers may choose a block-based or text-based programming environment based on the prior experience of the students and the selected problems. For a 9-week module, students will study the history of computers and computer science, with a focus on the impact of Virginians. In the 18-week module, students will build additional programming skills within the framework of computer science principles. For an 18-week module, students will complete one or more projects to include programming, hardware and software integration, and collaboration.

Programmable computing tools will be used to facilitate design, analysis, and implementation of computer programs. Students for exploring and creating computer programs, facilitating reasoning and problem solving, and verifying solutions should use these tools.

6-week Core Module

Algorithms and Programming

MSCSE.1 The student will design and iteratively develop programs that combine control structures, including loops and conditionals.

Context of the Standard
Programs are collections of code organized in algorithms that can accomplish a variety of tasks. Programs can be developed to perform calculations, manipulate data, or simply to be creative. Programs can involve different control structures such as loops or if-statements and variables; these control structures are blocks of programming that analyzes variables within the program code to adjust and use accurate values as they change. Control structures help students develop their problem solving skills, and foster computational thinking. Effective variable use makes the problem solving process easier and faster. In middle school, students continue their study of programming through a focus on algorithms. They work both collaboratively and individually to develop algorithms to reflect tasks in daily life; these algorithms become more complex as they recognize and use

Context of the Standard
loops and events in the algorithms they construct.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Develop, test, and revise a program using an iterative process. • Create a program that uses a combination of control structures. • Utilize a combination of mathematical and logical structures in a program. • Test a program for correctness. • Adjust and debug programs to be more efficient.	Students should <i>investigate</i> these concepts: • How does an iterative design process facilitate program development?	Students should <i>apply</i> these terms in context: • Control structure • Iterative process

MSCSE.2 The student will investigate variables and data types, including simple operations on strings.

Context of the Standard
Variables are programming elements capable of storing a piece of data. A variable is like a container with a name, in which the contents may change, but the name (identifier) does not. The identifier makes keeping track of the data that is stored easier, especially if the data changes. Naming conventions for identifiers, and thoughtful choices of identifiers, improve program readability.

Context of the Standard

Variables can hold a few different types of data: numbers, Boolean values (i.e. true and false), text characters, or strings of characters to make a word or phrase. Different data types will require the variable holding it to use different amounts of memory, and they operate differently from each other.

A string is also a data type used in programming used to represent text rather than numbers. Quotations are used to represent data in a string. Operations can be conducted by using both data and strings. For example, adding two numbers together will produce the sum of their values, where two strings added will result in their values placed end-to-end (“cat” + “dog” = “catdog”).

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify different types of data to be held in variables. • Conduct simple operations using non-numeric data types.	Students should <i>investigate</i> these concepts: • What types of values can a variable hold? • How may different data types interact when completing operations?	Students should <i>apply</i> these terms in context: • Variable • Strings

MSCSE.3 The student will implement a program that accepts input values, stores them in appropriately named variables, and produces output.

Context of the Standard
Computer systems are complex machines that primarily perform four main tasks: take input, store values, process data, and give output. Computer programs also typically follow a standard process called the input-process-output model. Data can be inputted into a program from the user, a data file, or a sensor. The processing stage can involve mathematical calculation, or logical control through the use of variables. These variables are created by the user and should be given names indicative of their content (e.g., a variable named person_height vs. var1). Once results are calculated, they should be produced for use. This output could be in the form of visual text, graphics, audio, or data values to be used by another program.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Develop a program that takes assorted user input, assigns variables, and provides output that meets the expectations of an assigned task. • Create appropriately named variables to store values used in the program. • Produce output made from various stored variables.	Students should <i>investigate</i> these concepts: • What factors should you consider when naming variables? • What are examples of different outputs generated by an app?	Students should <i>apply</i> these terms in context: • Variable

MSCSE.4 The student will document programs in order to make them easier to trace, test, and debug.

Context of the Standard
When creating programs, it is highly recommended to use documentation and comments. Documentation allows programmers and reviewers to understand the intention behind a particular portion of code. Documentation comments are formatted to describe portions of the program but are not actually integrated into the program flow. Often, a programmer will write code and not return to it for lengths of time; documentation can be used to remind them of the intention behind their code and is very useful in debugging, as it signals how the program should be acting. The use of documentation when creating and redesigning a program greatly improves the efficiency of code review. Well-documented programs are easier to follow, and individual portions can have their function identified. If that function is what is malfunctioning in the program, debugging can be completed more quickly and easily.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Use proper program documentation forms. • Use documentation to explain the purpose and function of a section of code.	Students should <i>investigate</i> these concepts: • How does program documentation make programming easier? • Why do comments make it easier to read a program written by someone else?	Students should <i>apply</i> these terms in context: • Documentation

Additional Content for 9-week Module

Impacts of Computing

MSCSE.5 The student will discuss issues of bias and accessibility in the design of existing technologies.

Context of the Standard
The construction of a program or innovation not only needs to focus on the parameters of that task or problem; it also needs address the needs and backgrounds of the intended users. Bias and accessibility are two issues that must be taken into consideration when designing a program. Bias is defined as prejudice in favor of or against one thing, person, or group compared with another. Examples of this bias could be as obvious as a programmer creating characters in a game that only reflect particular genders or ethnicities, or more subtle, like the choice of language used in a program's output, or the choices made by a seemingly random program that actually favor a particular group or preference. Accessibility refers to the design of products, devices, services, or environments for people with disabilities. These disabilities may include visual, auditory, motor, and cognitive disabilities. The Web Content Accessibility Guidelines (WCAG) defines criteria to ensure that Web content is more accessible to people with disabilities.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify a bias present in an existing technology. - Identify an accessibility issue in an existing technology. - Suggest a fix to an accessibility issue in a current technology.	Students should <i>investigate</i> these concepts: - What types of bias can be seen in a current technology that you use on a regular basis? - How do companies and programmers address accessibility concerns? - Why is it beneficial to remove bias from a technological application?	Students should <i>apply</i> these terms in context: - Bias - Accessibility

MSCSE.6 The student will describe and explain the history of computer science, including naming significant historical figures and describing their impact on the field.

Context of the Standard		
The use of devices to aid in computation date back to ancient times with mechanical counting devices like the abacus. These devices grew in complexity, and had their purposes expand in function. Charles Babbage created the idea of a computer that could be programmed to accomplish a variety of tasks. Ada Lovelace extended that idea, and published the first “algorithm,” or set of instructions to be carried out by a computer. Alan Turing, a British World War II code-breaker, theorized that computers and algorithms could be broadened to accomplish any task. Computers became smaller, more powerful, and more available to the public. The Internet was created as a way to connect computers across great distances. A new system of organization on the Internet was created in 1991 by Sir Tim Berners-Lee called the World Wide Web. This was a set of rules that unified how Internet communication could happen on a worldwide scale. Programs on the web also became faster and more powerful over time. Now, computers have gotten very small but very powerful. They are embedded into more everyday devices to increase connectivity and functionality. Computer science has greatly shaped society, and computers will continue to become more integral in our everyday lives.		
Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe the function of early computational devices. - Create a timeline showing the major developments in computing technology.	Students should <i>investigate</i> these concepts: - How did the purpose of computers change over time? - How did advances in computing technology affect the pace of change in related fields (mathematics, science, engineering, etc.)? - Do you believe that advancement in computing was only possible because of the accomplishments of particular people, or would these advancements have happened regardless?	Students should <i>apply</i> these terms in context: Computer

Additional Content for 18-week Module Algorithms and Programming

MSCSE.7 The student will use flowcharts and/or pseudocode to address complex problems as algorithms.

Context of the Standard
In computer science, the development of programs uses an iterative design process involving design, implementation (programming), and review (debugging) until the program runs correctly. The design stage occurs before beginning to program. The planning stage is when the programmers gather information about the problem and sketch out a solution. This design process may include the use of pseudocode - writing out the steps of a program in English to make sure the flow of control and logic make sense. Flowcharts are another tool programmers may use when designing an algorithm or computer program. The flowchart outlines the steps that are needed in the development of an algorithm or program.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Create a visualization of the path through a program.	Students should <i>investigate</i> these concepts: - Why should you plan a design for a program before starting to write it? - How are roads and traffic signs like a flowchart in programming? - How does pseudocode help programmers organize their thoughts?	Students should <i>apply</i> these terms in context: Pseudocode

MSCSE.8 The student will incorporate existing code, media, and libraries into original programs, and give attribution.

Context of the Standard
As programs grow more complex, the choice of resources that aid program development becomes increasingly important. These resources include existing code, available media, and libraries. Repositories of sections of code are available for use online. Libraries are larger groupings of usable code that serve a particular purpose. Libraries contain collection of files, programs, routines, scripts, or functions that can be referenced in the programming code. They are often packaged up and distributed for easy inclusion in a program. Regardless of the source of a resource that may be used in the construction of a program, it is important that proper attribution is given to any borrowed components.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Integrate portions of code, algorithms, and/or digital media from other sources in their own programs and websites. - Give attribution to the original creators to acknowledge their contributions.	Students should <i>investigate</i> these concepts: - Why should you give attribution to the original creator of a product? - How does the use of existing code or libraries accelerate program development?	Students should <i>apply</i> these terms in context: - Code library - Cite - Source - Plagiarism

MSCSE.9 The student will systematically test and refine programs using a range of test cases.

Context of the Standard
As part of the iterative design process, programs should be reviewed and tested to determine if the design goal is met through the generation of specific output. Testing programs using a range of test cases is a necessary step for assessing program correctness. A test case is a single input with an expected output to test the correctness of a program. Programmers need to test a program to make sure it works for all the possible values received as input. This can include values within the range of expected values as well as those outside the given range. Often, programmers will use “edge cases” - parameters that test the extremes of a scenario - to test their programs, ensuring that they will work with even the largest and smallest inputs. In beta test situations programmers use random people to test a program to find “bugs.” Programmers record “versions” of programs as they update and improve their code.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Predict the outcome or output of a program by examining written code. - Use multiple test cases to test the logic and mathematics of a program.	Students should <i>investigate</i> these concepts: - Why is there a need to debug? - How does program tracing help with the debugging process? - How can you predict the outcome of a program without running it?	Students should <i>apply</i> these terms in context: - Bug - Debug

Networks and the Internet

MSCSE.10 The student will model the role of protocols in transmitting data across networks and the Internet.

Context of the Standard
Protocols are sets of rules or procedures defined for particular actions. Protocols describe established commands and responses between computers on a network, such as requesting data or sending an image. Even simple messages sent over the Internet have a number of protocols used to facilitate the transmission of the message. These protocols work together and rely on each other to handle translation and transmission. There are many examples of protocols including TCP/IP (Transmission Control Protocol/Internet Protocol) and HTTP (Hypertext Transfer Protocol), which serve as the foundation for formatting and transmitting messages and data, including pages on the World Wide Web. Routers also implement protocols to record the fastest and most reliable paths by sending small packets as tests. <i>The student is not expected to know details of how specific protocols work in computing systems.</i>

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify the role of protocols in network communication. - Summarize how protocols make a decentralized Internet possible (i.e., no governing authority over the Internet).	Students should <i>investigate</i> these concepts: - How would the Internet function differently if there were no set protocols? - What are some examples of protocols that humans use to communicate with one another?	Students should <i>apply</i> these terms in context: Protocol

Cybersecurity

MSCSE.11 The student will apply multiple methods of encryption to model the secure transmission of information.

Context of the Standard
In order to protect the security of data transmission on the Internet, encryption is used to hide sensitive data. There are tradeoffs between digital security and usability, speed, and cost. As needs and resources are considered, encryption techniques can be chosen appropriately. Encryption is an important aspect of communicating in a public system like the Internet. Without it, anybody would be able to view personal identifying information of anyone using the system from the packages that they send over the Internet. Some methods of encryption include Caesar cipher and Vigenere cipher. Caesar cipher uses an alphabetic shift to mask the true contents of a message. This is the easiest form of encryption to use and break. Vigenere uses a keyword to drive character substitution making the encryption more difficult to crack.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills:	Students should <i>investigate</i> these concepts:	Students should <i>apply</i> these terms in context:

Essential Skills	Essential Questions	Essential Vocabulary
- Identify different types of data that could be encrypted. - Use simple encryption strategies to encode a message.	- How might a security expert protect data on a network? - Why is encryption necessary when using computers for communication? - What types of data should be encrypted?	- Encryption - Data security

MSCSE.12 The student will explain how physical and digital security measures protect electronic information.

Context of the Standard
Students will extend their knowledge of the need for physical and digital security measures to understanding and explaining how both types of measures protect our data and personal information. Students can explore different types of security measures and understand how each one protects us in different ways from different types of electronic breaches and attacks. Physical security measures include locking rooms with devices, badges, fingerprints, security cameras, paper shredding, etc. Digital security measures include firewalls, anti-virus software, strong passwords, anti-spyware, etc.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify examples of physical and digital security measures. - Explain why data security is necessary.	Students should <i>investigate</i> these concepts: - What are examples of physical security protections and how do they protect data? - What are examples of digital security protections and how do they protect data?	Students should <i>apply</i> these terms in context: - Password - Firewall

Essential Skills	Essential Questions	Essential Vocabulary
	Why should a student implement security measures when working with digital information?	

Data and Analysis

MSCSE.13 The student will collect data using computational tools and transform the data to make it more useful and reliable.

Context of the Standard
Data collection is essential to help provide insight into patterns and other phenomena. These data can be collected by hand or through the use of computational tools such as sensors or surveys. Automated data collection (e.g., smartwatches regularly collect data on a person's heart rate, steps, etc.) is very efficient at gathering large sets to show patterns on a larger scale. The data often needs to be sorted, counted, and otherwise manipulated in order to make it easier to analyze. This manipulation can be done using spreadsheets, graphing, or other specialized software.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Use a computational tool to collect a large set of data. - Transform data into different formats using computational tools.	Students should <i>investigate</i> these concepts: - Why is raw data typically difficult to use in analysis? - How can we collect data electronically? - What are examples of tools that can be used to collect data? - What are different ways data sets can be organized and displayed?	Students should <i>apply</i> these terms in context:

MSCSE.14 The student will refine computational models based on the data they have generated.

Context of the Standard
Once collected, data must be analyzed in order to make predictions and observations. These data can be used to build models that can replicate a phenomenon. Modeling means creating a physical replica or equations of a situation or activity. Programmers will create computational models to describe real-world systems. Computational models can be created in pseudocode (written) or equations, with robots, in block languages (i.e. Scratch), or in text-based languages (Java, Python, etc.). There are also content specific tools for modeling such as MatLab for mathematics.). Some examples of computation models might be modeling and simulating a bungee jump, investigating the temperature of melting ice, or investigating the forces and attractions in the states of matter. Models are also constructed to test hypotheses. The data generated from the model are used to evaluate whether the models are accurate, to make adjustments, and draw conclusions. Models need to be tested for accuracy and refined as necessary. If a model generates an unexpected result based on data, then the model may need adjustment. Programmers engage in an iterative process as they analyze data and revise models multiple times in order to generate accurate models.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Manipulate a data set in order to create a model. • Adjust a computational model to more accurately reflect a data set.	Students should <i>investigate</i> these concepts: • How can you tell if a model needs adjustment? • What components of a computational model can be adjusted? • What factors are considered when evaluating the accuracy of a model?	Students should <i>apply</i> these terms in context: • Data

MSCSE.15 The student will represent data using multiple encoding schemes.

Context of the Standard
When data is being stored on a computer, there are several options for how it will be organized. Encoding schemes are systems of rules for converting data into a format that is usable by someone else. Some schemes are simple and hold basic text but have limitations based on memory used per character (ASCII). Some hold a wider variety of characters including emojis, foreign language characters, and specialized symbols; this encoding scheme has a greater flexibility due to size (Unicode). Some encoding schemes are designed to hold a variety of data types but take up considerably more memory space (HTML). <i>Students are not expected to make manual conversions between data forms using encoding schemes.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Describe how an encoding scheme can represent data in other forms.	Students should <i>investigate</i> these concepts: - How can a computer represent data in its numeric-only format? - What are examples of different numeric systems used by computers to represent non-numeric things?	Students should <i>apply</i> these terms in context: Encoding scheme

Impacts of Computing

MSCSE.16 The student will compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

Context of the Standard
Computer use has permeated nearly every aspect of people's daily lives. This has brought many benefits to society, with increased levels of communication and access to information. There are many tradeoffs pertaining to consistent access to computing technology. These could include identity theft, cyberstalking, catfishing, fraudulent purchases, etc. Inadvertent data collection makes many people feel as though their privacy is being invaded, particularly since companies can track user data and sell it to advertisers or companies for analysis.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Compare and contrast benefits and risks of everyday use of computing technology.	Students should <i>investigate</i> these concepts: - What computing technologies do you use every day? - What are the benefits of consistent access to technology? - What are the drawbacks to consistent access to technology?	Students should <i>apply</i> these terms in context:

MSCSE.17 The student will collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact or visualization.

Context of the Standard
When conducting data analysis, a larger set of data allows analysts to make generalizations about the situation/phenomena to which the data describes. Strategies such as surveys or crowdsourcing allow students to collect data from sources outside their work group. Often, outside opinions or ideas can help a creator or group break through a deadlock by providing a fresh perspective. By soliciting these ideas through crowdsourcing, a programming team can know that their program represents the views and preferences of a wide swath of people. <i>Teacher note: Reference division and school protocols prior to engaging in surveying or other crowdsourcing means with students.</i>

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Use a form or survey to collect data from a wide variety of people.	Students should <i>investigate</i> these concepts: Why is it better to collect large sets of data when conducting analysis?	Students should <i>apply</i> these terms in context: Crowdsourcing

MSCSE.18 The student will describe tradeoffs between allowing information to be public and keeping information private and secure.

Context of the Standard
The Fourth Amendment of the US Constitution protects every American's right to privacy. However, this amendment was written long before the advent of the Internet, where personal information is easily accessed and, in some cases, exploited. Modern tech companies, such as Facebook, Google, Amazon, and others, mine users' personal data and online habits to better advertise content

Context of the Standard
and products towards individuals. Students will explore what information they think should be publicly available. Students discuss the benefits and drawbacks to keeping information private when compared to public release.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify and describe what information is appropriate, safe and responsible to share publicly. - Identify and describe what information is appropriate, safe and responsible to keep private.	Students should <i>investigate</i> these concepts: - What are the benefits of sharing personal information online? - What kind of personal information is regularly collected through technology use? - What value is your or your family's personal information to another person or business? - What information is appropriate to be public? Private? - Why should we allow certain information to be public vs. private? What are the consequences?	Students should <i>apply</i> these terms in context:

Computing Systems

MSCSE.19 The student will systematically identify and correct problems with computing devices and their components.

Context of the Standard
As with any system, there are times that a computer system does not work as intended. Although computing systems may vary, common troubleshooting strategies can be used on them, such as checking connections and power or swapping a working part in place of a potentially defective part. Since computing devices are composed of an interconnected system of hardware and software, troubleshooting strategies may need to address both. Students are expected to use accurate terminology to describe simple problems with computer hardware and software. Common troubleshooting strategies, such as checking that power is available, checking that physical and wireless connections are working, and clearing out the working memory by restarting programs or devices, are effective for many systems.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify when a device or program is not working properly. • Perform simple troubleshooting tasks. • Apply troubleshooting strategies as needed in a classroom setting.	Students should <i>investigate</i> these concepts: • What are different troubleshooting tactics you should try if a program is not working? • Why is it important to be as specific as possible when you are describing a problem?	Students should <i>apply</i> these terms in context: • Troubleshooting

MSCSE.20 The student will explore the relationship between hardware and software using the Internet of Things.

Context of the Standard
An increasing number of devices have gained Internet connectivity. It is now possible to communicate with everyday objects like house lights, thermostats, or a garage door opener. These devices can also assess their surroundings using input components such as sensors (hardware), apply their programming logic (software), and relay output to an appropriate device to perform an action (hardware and software). They can also relay information to a user via an app who can decide on appropriate action.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe what makes a device part of the “Internet of Things.” - Describe the difference between smart and normal devices.	Students should <i>investigate</i> these concepts: - How does an Internet-connected device behave differently from a non-connected device? - What is the “Internet of Things?”	Students should <i>apply</i> these terms in context: Internet of Things

36-week Module

Algorithms and Programming

MSCSE.21 The student will

- a. work in a team to distribute tasks;
- b. maintain a timeline; and
- c. use iterative design to solve problems, including peer review and feedback.

Context of the Standard

Development of computational artifacts (program, image, audio, etc.) is an iterative process and often a collaborative effort. Working well in a group requires students to practice strategies during the planning, writing, and review process such as delegation of responsibility, peer-review, revision of contributions, and assigning self-imposed deadlines.

Due to the highly collaborative nature of programming projects, team members need to employ effective communication strategies. In order to avoid redundant work or gaps in the necessary code, all groups need to have open communication to help standardize the flow of the program. In order to develop a program, tasks can be broken down and created in modules by different groups of students, which requires clear and consistent communication between these groups. Additionally, the peer review fosters effective communication, and helps students see a variety of coding styles as well as identify common mistakes.

The creation of appropriate and realistic timelines is difficult to many students, particularly when working in groups. Students may need teacher guidance and scaffolding in the timeline construction process as well as regular reminders to look back and revise timelines as needed throughout the process. These deadlines are used to construct a timeline that informs group participants of responsibilities and expectations when completing a project such as the development of a computational artifact.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Work in teams to plan, design, and revise a program to complete an assigned task.	Students should <i>investigate</i> these concepts: • How does an iterative process help in the completion of a project? • How does the use of a timeline facilitate	Students should <i>apply</i> these terms in context: • Iterative design

Essential Skills	Essential Questions	Essential Vocabulary
Construct a timeline that reflects the expectations and responsibilities of the members of a group.	group work?	

MSCSE.22 The student will decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

Context of the Standard
Decomposition breaks problems down into smaller pieces (subproblem). A subprogram is a sequence of instructions whose execution is invoked from one or more remote locations in a program. Problem decomposition allows for complex tasks to be solved in manageable chunks. Once a program has been broken down, it enables different group members to work on different parts at the same time. Decomposition can occur in the planning and design stage, allowing for the separate subprograms to be more easily tackled, or when debugging a long and complex program. Students will practice this skill on programs they wrote and written by others.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Breakdown problems into subproblems to facilitate creation of a program. - Use decomposition in all parts of the program development process (i.e.	Students should <i>investigate</i> these concepts: - Why does breaking a problem down into smaller problems make the overall task easier? - How does decomposing a program into subprograms help programmers when	Students should <i>apply</i> these terms in context: - Decompose - Subprogram

Essential Skills	Essential Questions	Essential Vocabulary
plan, design, and review).	debugging a program? • How can decomposition be applied throughout the development process? • Why would using subproblems in a program be thought of as a time saving measure?	

MSCSE.23 The student will create procedures with parameters to organize code and make it easier to reuse.

Context of the Standard
Functions are like variables, except instead of storing data they store lines of code. Programmers use functions to help “chunk” large, self-contained sections of code to help make code more readable and to allow code to be used multiple times without having to re-write it. This increases code reusability by allowing an algorithm to be referenced and used wherever appropriate. Parameters are additional information that comes with the command call (e.g., squareRoot() vs. squareRoot(9)). These parameters provide the ability for a function to be used in a variety of situations.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create functions within a programming environment to accomplish a task. • Create functions with parameters to use in a variety of scenarios.	Students should <i>investigate</i> these concepts: • How do parameters increase the capability of a function?	Students should <i>apply</i> these terms in context: • Function • Parameter

Computing Systems

MSCSE.24 The student will recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

Context of the Standard
Development of computing technology is an ongoing process. New technology components often introduce new software features. Based on feedback from the use of a new feature, developers will make adjustments for the next round of software updates. This feedback may be from user recommendations or from analysis of specific uses of data and functions. The use of feedback when designing programs is a practice that can be used in multiple settings. Students can make similar recommendations for improvements to apps they use on their phones.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Analyze the design of a computing device for the purpose of making recommendations. - Make recommendations for improvements to a computing device.	Students should <i>investigate</i> these concepts: - What improvements can you make to a computing device? - What is the role of data analysis in determining changes in computing devices?	Students should <i>apply</i> these terms in context: User interface

MSCSE.25 The student will design projects that combine hardware and software components to collect and exchange data.

Context of the Standard

Context of the Standard
Hardware can be used in tandem with software to facilitate data analysis in an efficient way. Sensors are hardware components designed to collect data that would otherwise be difficult to collect by hand. This could be by the type of the data or the amount of data to be collected.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Plan or sketch ideas combining hardware and software components that collect and exchange data. - Select the hardware and software components for project designs by considering factors such as functionality, cost, size, speed, accessibility, and aesthetics.	Students should <i>investigate</i> these concepts: - How can hardware and software components be combined in the completion of a specific task or the completion of project? - Why is important to know what kind of data you want to collect before you begin to design a project? - What are the different kinds of data you can collect using hardware and software?	Students should <i>apply</i> these terms in context: - Sensors - Probeware

Computer Science Foundations

The Computer Science Foundations standards outline the content for a one-year course with an emphasis on computer programming within the context of broader concepts of computer science. The standards build on the concepts of computer science developed in prior grade levels. The standards provide a transition from block-based programming to a text-based programming language and familiarize the student with developing and executing computer programs. Teachers are encouraged to select programming languages and environments, problems, challenges, and activities that are appropriate for their students to successfully meet the objectives of the standards.

Programmable computing tools will be used to facilitate design, analysis, and implementation of computer programs. Students for exploring and creating computer programs, facilitating reasoning and problem solving, and verifying solutions should use these tools.

Computing Systems

CSF.1 The student will

- a. compare the structures, functions, and interactions between application software, system software, and hardware; and
- b. explore the relationship between hardware and software using the Internet of Things.

Context of the Standard
Computing systems are comprised of many modular, interconnected pieces of hardware and software. These pieces work together in a hierarchical system to accomplish complex tasks. When computers communicate with one another over networks, data is passed from application software, down to system software, and finally down to a hardware layer that communicates directly with the hardware layer of the receiving machine. The receiving machine then passes data up to its own system software and application software repeating the process in reverse. Often, any piece of data will pass up and down the hardware and software layers of many machines before reaching its final destination (see CSF.2). Internet-connected devices (such as Internet of Things devices) are a good example of this phenomenon because the device itself will often relay data to another device that controls its behavior. For example, an IoT thermostat (hardware) will push temperature data to an application (software) running on a smartphone. In turn, the application on the smartphone will control the thermostat at the user's request by sending data over the Internet.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Illustrate the hierarchy of hardware, system software, and application software. • Describe the relationship between hardware and software. • Analyze the path that data travels along in a typical instance of networked communication.	Students should <i>investigate</i> these concepts: • What are the roles of hardware and software when using a network? • Why do computing systems make use of a layered hierarchy when communicating with other systems?	Students should <i>apply</i> these terms in context: • Hardware • Software (System & Application) • Network • Internet of Things (IoT) • Hierarchy

Networks and the Internet

CSF.2 The student will model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.

Context of the Standard
Computers can send and receive messages when connected to one another. These networked devices pass information among themselves using a specific set of rules called a protocol. Reliable communication through networks depends on all sent information arriving at its destination, regardless of heavy traffic or damaged connections. Computers also need a reliable way of reporting and resolving communication errors. To achieve these two prerequisites, computers break down messages (e.g., emails, images, music) into smaller chunks of data called packets. These packets make their way to their destination computer separately using different routes, after which the receiving computer puts them back together in order. In large networks like the Internet, packets may travel among dozens of different computing devices; each device passes them on until all of the packets reach their destination. If a packet is missing and the message is incomplete, the receiving computer will request that the sender resubmit the message.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe the process of sending a file through a network. - Create a model of networked communication. - Explain how sending data in packets ensures reliable communication among computing devices.	Students should <i>investigate</i> these concepts: - What are the physical qualities of a reliable network? - What are some ways networked communication might be disrupted?	Students should <i>apply</i> these terms in context: - Internet - Network - Packet - Protocol - Reliability

CSF.3 The student will explain the role of protocols in transmitting data across networks and the Internet.

Context of the Standard
Even the sending of a simple email message requires multiple levels of communication between devices. In order for a message to maintain its fidelity, it is essential that there be a series of protocols, or agreed upon standards of syntax, format, packet size, etc, used at each level. Anyone can join at any time and successfully communicate with anyone else on the network without deciding on rules for communication in advance. Internet communication is made possible by several protocols, including TCP, IP, UDP, and DNS. These sets of rules interact to build a model used by almost all Internet communication (OSI model). Without them, it would be very difficult for computing devices to communicate with one another without explaining in advance how to interpret information sent over the network.

Essential Skills	Essential Questions	Essential Vocabulary
------------------	---------------------	----------------------

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Define <i>protocol</i> in the context of computer science. • Summarize how protocols make a decentralized Internet possible. • Explain the role of protocols in networked communication.	Students should <i>investigate</i> these concepts: • What are some examples of protocols that humans use to communicate with one another? • How might the Internet function differently if every nation had its own set of protocols?	Students should <i>apply</i> these terms in context: • Network • Packet • Internet • Reliability • Scalability • Protocol

CSF.4 The student will evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.

Context of the Standard
Networks are made up of interconnected computing devices. Some are familiar, like laptops, smartphones, and desktop computers. Others, called servers, are designed to send files over the Internet. Servers are Internet-connected computing devices that store web pages and other files. They process requests from computers and then distribute results over the Internet. The computing devices in a network work together to ensure that communication is fast and reliable. How they are arranged in that network is referred to as the network topology. Many people have computing networks in their homes, where several devices are connected to the same access point. This small home network is called a local area network (LAN). Each device in a LAN has a unique address called an IP (Internet protocol) address. Neighborhoods might have dozens of local area networks, one or more for each home. Just like the devices within it, each local area network has a unique IP address. Local area networks are connected to switches, which serve as hubs for LANs in a neighborhood. This web of LANs is called a wide area network (WAN). A city might have hundreds of wide area networks, each with its own unique IP address. Wide area networks all eventually connect to routers. The Internet is a vast web of routers, each one passing along packets (see CSF.2 & CSF.3) to the appropriate destination. This system is reliable and scalable because each machine only has to keep track of the devices it is directly connected to. A

Context of the Standard
smartphone doesn't need to know the IP addresses of all the switches and routers between it and the server it wants to connect to; it only has to know the IP address of the server (stored in an easily accessible public database) and the IP address of its local wireless access point.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Model how information is transmitted among computing devices that make up a network. • Evaluate how addressing facilitates the scalability and reliability of the Internet. • Illustrate the arrangement or topology among elements of a network.	Students should <i>investigate</i> these concepts: • How many devices might a packet travel through on the way to its destination? • Are Internet communications public or private? How do you know?	Students should <i>apply</i> these terms in context: • IP address • Local Area Network (LAN) • Reliability • Router • Scalability • Server • Switch • Wide Area Network (WAN)

Cybersecurity

CSF.5 The student will identify and explain ways that sensitive data (assets) can be threatened by malware and other computer attacks, using appropriate terminology.

Context of the Standard

Context of the Standard

People can exploit vulnerabilities in computing devices, systems, and use cases with malicious intent. Some exploits will focus on disrupting network services, while others are perpetrated with the goal of gaining access to sensitive assets. These people make use of malware (malicious software installed on a computer) and viruses (scripts that may be hidden in existing files or programs) to spy on users, access private files and data, or lock victims out of their computers. For example, ransomware is malicious software that encrypts a victim's files and demands a ransom before the data is decrypted. In contrast, a virus might create a backdoor in a system that allows an attacker to gain access to it at a later date.

Attackers might also monitor network activity, recording email addresses, phone numbers, and other unprotected data to use in future attacks. For example, someone could set up a fake wireless access point, and will exploit the privileges afforded to a trusted network access point to spy on data transmitted over that network connection. Attackers will also make more mundane attacks; they may guess passwords to gain access to emails, make fake phone calls to gain information about victims, or send emails with fake links or viruses stored in seemingly safe attachments.

The strategies people use to exploit computing systems are always evolving, and security experts are tasked with understanding how to adapt to new attacks and vulnerabilities.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Identify different types of cyber-attacks. • Explain why attackers might want to gain personal information/sensitive data from their target. • Describe different cyber-attacks and describe how they might affect a computing system.	Students should <i>investigate</i> these concepts: • How might a security expert protect data on a network? • If someone discovers a vulnerability in a piece of software, should they make it public, or keep it secret?	Students should <i>apply</i> these terms in context: • Data • Exploit • Malware • Virus • Vulnerability

CSF.6 The student will give examples of ways to protect sensitive data (assets) from malware and other computer attacks and evaluate them according to multiple criteria.

Context of the Standard
Bad actors make use of many strategies to gain access to personal data. People seeking to protect their data can take several proactive steps which ensure that their data is safe from common attacks. Creating secure passwords unique to every account, configuring the firewalls on personal computing devices, not connecting to unfamiliar wireless access points, installing reputable anti-virus and anti-malware software (anti-x software), and practicing safe-decision making online by not clicking unfamiliar links or responding to unsolicited communication are all good strategies for protecting oneself online. However, there is no way to ensure that any online communication is private with 100% confidence. Part of protecting oneself online is knowing that all online communication is potentially public, and behaving accordingly. In evaluating possible protective measures, the possible threats should be considered as well as system vulnerabilities and risks of breach. Most people will not be victims of large-scale data breaches. Highly visible or vulnerable individuals attract more attention from bad actors; politicians, corporate executives, and celebrities must often take extraordinary steps to adequately protect their data from potential exploitation. People who fail to protect themselves from attacks will often be attacked multiple times, as their personal information is disseminated online.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Compare and contrast safe and unsafe computing practices. • Describe ways to protect personal information on a computing device. • Synthesize and communicate strategies for protecting personal data	Students should <i>investigate</i> these concepts: • What are some strategies for protecting your sensitive data? • Who are some people who might be popular targets for cyber-attack or espionage?	Students should <i>apply</i> these terms in context: • Anti-X Software • Cybersecurity • Cyber Attack • Data • Vulnerability

Essential Skills	Essential Questions	Essential Vocabulary
to someone unfamiliar with cyber best practices. Evaluate a protective measure in its effectiveness against cyberthreats.		Firewall

CSF.7 The student will explain typical tradeoffs between usability and security and recommend security measures in a given scenario based on these (or other) tradeoffs.

Context of the Standard
The people who use computing devices are often the most significant vulnerability in a system. Computing devices have their own host of security vulnerabilities, but users can be tricked, misled, or otherwise manipulated into granting bad actors access to their personal data. When interacting with computing systems, users experience a tradeoff between security and usability. If a computing devices makes use of multiple layers of security, users may begin to encounter barriers that make the computer more difficult to use. For example, a highly filtered Internet connection will keep users from accessing malicious websites, but may also inadvertently block some safe websites. False positives like these are inevitable in a highly secure system. Anti-virus software will often scan downloaded files for malicious content, forcing users to wait until the scan is done before they open the file. High levels of security can create significant barriers to usability; as a result, security measures are often calibrated to match the data being secured. For example, an online banking profile is more secure (and as a result, will be less usable) than an online gaming account. To ensure that sensitive data is not accessed by unwanted people, users often must provide private credentials (i.e., passwords) when accessing their information. When tasked with protecting particularly sensitive data, software designers will often add additional layers of security, like security questions or two-factor authentication (i.e., when a user is required to produce two sets of credentials, like a password and a secret code texted to their phone).

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain how security measures might affect information transmission speed within a network. • Investigate and describe how security measures might be adjusted for different types of information. • Identify different strategies for securing sensitive information.	Students should <i>investigate</i> these concepts: • What are some examples of software you use that require credentials? What information are these credentials protecting? • Do you experience any tradeoffs between usability and security?	Students should <i>apply</i> these terms in context: • Authentication • Credentials • Malicious • Tradeoff • Useability

CSF.8 The student will write or adapt a program to validate its input and to avoid certain kinds of vulnerabilities.

Context of the Standard
People who want to compromise a system will often exploit vulnerabilities built into software by misusing or exploiting existing avenues for user input. Examples of these types of attacks include SQL injection and command line injection, where attackers use existing software protocols to trick the program into revealing information about an underlying database or forcing the computer to execute code unintentionally. To protect against these kinds of attacks, designers will validate user input; in other words, designers will program the software to reject inappropriate input or fix it in predictable ways. Designers will often create a list of approved input values, or a list of input values that should be rejected. This ensures that the software is used only for its intended purpose. Another example of input validation is when software rejects insecure passwords during the account creation process. The software will tell users to pick a new password because the one they initially chose would be easy to guess or exploit.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Design a program that takes user input and validates it against a list of approved input options. • Write a program that returns affirmative output given an input that is validated against an approved list.	Students should <i>investigate</i> these concepts: • What are some personal best practices for validating digital communication (e.g., emails or texts from unknown senders)? • Can you identify an example of input validation in the real world?	Students should <i>apply</i> these terms in context: • Command Line Injection • Exploit • Input • Software • SQL Injection • Validate • Vulnerability

Data and Analysis

CSF.9 The student will evaluate the tradeoffs in how data elements are organized and where data is stored.

Context of the Standard
On an individual program level, data are typically stored in some sort of a structure. These structures may include arrays, lists, or others. Arrays store multiple values in sequential memory and are referenced by an index or position. Multiple options are considered for the storage and access of these data. The choice of a data structure can be influenced by data access speed, memory usage, organizational complexity, and other factors. On an individual computer level, data are stored on local hard drives in a variety of file formats. Simple file formats typically use less memory space but only store basic forms of the data with little formatting. Larger file formats provide a variety of formatting options and extra protection but typically are only accessible through specific programs. On a network level, data are often stored on “cloud” servers and accessed remotely for manipulation and analysis. This is beneficial for work that needs to be conducted by multiple people, but requires network access to be possible. With consistent network access, there is need for data security and encryption to protect the data from unwanted access.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify different data structures used by programmers. - Compare and contrast the aspects of different data structures. - Compare and contrast local vs. cloud data storage. - Identify how data element storage differs depending on the program or system being used.	Students should <i>investigate</i> these concepts: - How can a programmer store multiple values in one structure? - What factors lead to choosing one structure over another one? - What are the benefits to using cloud storage instead of local storage? - What are the drawbacks to using cloud storage instead of local storage?	Students should <i>apply</i> these terms in context: - Array - List - Local storage - Cloud storage

CSF.10 The student will create interactive data visualizations using software tools to help others better understand real-world phenomena.

Context of the Standard
Software programs and hardware devices are constantly creating data. Smartwatches for example, generate data about movement, location, sleep, heart rate, and many other kinds of user input. In order to interpret the raw information generated by software and hardware devices, programmers create data visualizations—visual representations of data. Some visualizations are traditional (e.g., bar graphs, pie charts, scatter plots), while others are more dynamic, including sounds, haptic feedback, or input which allow users to explore the data interactively. Some programmers will use data to create data-based art, which focuses less on communicating trends and more on creating evocative visual, sounds, or experiences. A visualization may even make use of data being gathered in real time, where sensors or programs update a database and the visualization program updates by pulling new data constantly. For example, a visualization might show domestic flights by pulling data from an air traffic control database.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Generate or identify criteria for effective and/or evocative data visualization. • Create a visualization given a data set. • Use a data visualization to draw conclusions about the underlying raw data.	Students should <i>investigate</i> these concepts: • What types of data visualizations can you find on the Internet that is refreshed in real time (e.g., weather, traffic)? • What type of daily data could a school create, and how might you visualize that data?	Students should <i>apply</i> these terms in context: • Data Set • Data Visualization • Interactive Data • Raw Data

CSF.11 The student will use data analysis tools and techniques to identify patterns in data representing complex systems.

Context of the Standard
Software programs may generate incredibly large amounts of data. Software companies will use this data to make decisions about marketing, design, and future software updates. Scientists also use software to generate large amounts of data to better understand natural phenomena. The problem is that these large amounts of data are impossible to fully understand in a raw format, and too large and complex for simple graphical visualizations to communicate their depth. Data scientists and programmers will often use the term “big data” to refer to these massive datasets, which often contain many different variables and thousands or even millions of data points. In order to draw conclusions from large data sets and communicate those conclusions to stakeholders, programmers will create data visualizations (see CSF.10) that expand on or reimagine traditional data visualization methods (e.g., scatter plots, bar graphs, pie charts, line graphs). These data visualizations will often compare multiple parameters from a multivariate data set, or show how data changes over time using time-series data.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe why it might be difficult to draw conclusions from big data. - Draw conclusions about large multivariate data sets given visualizations.	Students should <i>investigate</i> these concepts: - Is facial recognition an ethical use of data analysis? Why or why not? - What might you learn about a website from data that tracks where users look on the page?	Students should <i>apply</i> these terms in context: - Analysis - Big Data - Bivariate - Data Visualization - Multivariate - Time-Series

Algorithms and Programming

CSF.12 The student will develop a program working individually and in teams using a text-based language.

Context of the Standard
Programming is how people create new tools and experiences with computers. Programming languages are the creative medium of computer programming, allowing people to solve problems by generating new tools that run on computing devices. Many times, programmers work in teams by dividing the work among the team members. For example, each team member might work on one page of the application. This allows individuals to focus on making one aspect of the program as good as it can be, without worrying about the details of the rest of the application. After everyone has finished their pieces, the team will come together and collaboratively combine their pieces into a cohesive whole.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Organize a plan to develop a program individually and as part of a team. - Construct a tool that solves a problem using text-based programming. - Identify future possibilities for adding features or fixing issues in a given program.	Students should <i>investigate</i> these concepts: - Why might you test a product while building it rather than waiting until you finish the product? - What are some of the differences between programming alone versus programming with a team?	Students should <i>apply</i> these terms in context: - Iterative process - Program - Prototype - Text-based language

CSF.13 The student will identify the expected output of a program given a problem and some input.

Context of the Standard
Software programs take input from users, and then give the users feedback based on that input. Fundamentally, programming structures 1) take parameters, and 2) return values based on those parameters. If you created a program to add two numbers together, the user would specify the two numbers in question, and then the program would return the sum of the two values. Programmers elaborate on this logical structure to create software programs that give users complex, often multisensory feedback. The ability to analyze code and predict its output given an input value is a fundamental programming skill. Programmers will exercise this skill at all levels of expertise. Additionally, comparing the expected output of a program to the actual output of that program helps programmers educate themselves and learn about how they can edit their code and create software that works as expected. Programmers (especially beginners) should always be asking themselves: “What do I expect to happen when I execute this code, and what actually happened?” If a program returns an unexpected value, it often means that the programmer must go back and debug (repair) the program by re-evaluating their initial assumptions about how individual pieces of code function.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Predict the output of a program that incorporates conditional statements given input values. • Create a program that returns given output values by taking given input values as parameters. • Repair a program that returns unexpected output values.	Students should <i>investigate</i> these concepts: • What questions should be considered when testing a program you wrote? • What are the next steps for you as a designer if the program does not work as expected?	Students should <i>apply</i> these terms in context: • Input • Output • Parameter • Return • Conditional • Execute • Debug

CSF.14 The student will design and iteratively develop programs for practical intent or personal expression, incorporating feedback from users.

Context of the Standard
Creating new software involves several steps, including designing, prototyping, testing, and debugging. Some programmers use the notion of a minimum viable product, or MVP, as a guide to help focus their creative efforts. An MVP is the simplest version of a tool that solves a problem. For example: if someone wants a tool to quickly move from place to place over long distances, a car might be a final solution. A car, however, is a very complex machine and may be difficult to design, prototype, and test in a reasonable amount of time. A minimum viable product for this problem might instead be a skateboard. It solves the problem, but is much easier to design, prototype, and test. After the skateboard is successful, the team might add on to it to solve the next problem (e.g., steering needs to be improved, so add a handle and make a scooter). This iterative process of designing, prototyping, testing, and debugging ensures that programmers and designers avoid wasting time and resources on a complex solution to a simple problem. For students learning programming, a minimum viable product is an excellent way of ensuring that students are frequently testing, debugging, and refining their software as they work toward short-term, iterative goals. By incorporating user feedback and testing throughout the build process from the very beginning, programmers ensure that they are not wasting time either working on code

Context of the Standard
that does not address users' problems, or writing code that will not work in the end. Programmers always try to hypothesize about an expected output, and then validate that hypothesis by testing code (CSF.13).

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Design a program using a planning process (wireframe, logic flowchart, pseudocode). • Create a program based on a design plan starting with a MVP (Minimum Viable Product). • Demonstrate the iterative development process by participating and getting feedback from user testing and debugging.	Students should <i>investigate</i> these concepts: • How is feedback used through the iterative design process? • What is an example of an iterative process that we use in our regular, non-programming lives? • Who should programmers talk to before designing and programming an app? Why?	Students should <i>apply</i> these terms in context: • Debugging • Feedback/User Testing • Iterative Design • Minimum Viable Product • Prototype

- CSF.15 The student will design and implement algorithms using
- sequencing of instructions;
 - conditional execution; and
 - iteration.

Context of the Standard

An algorithm is a series of instructions for the computer to execute given some sort of input value. A simple algorithm might take in a series of numbers and return the sum of those numbers. More often, however, algorithms are more complicated. For example, a social media site might use inputs like browsing history, posts, comments, or video viewing history to decide which advertisements to show to a specific user. Software designers often create visual artifacts like flow charts that communicate the algorithmic process before they start writing code, in order to demonstrate the logical process the algorithm utilizes.

Conditional execution refers to a part of an algorithm where the code asks the computer to compare two values, and execute two different pieces of code based on the relationship between those two values. For example, a sorting algorithm in a manufacturing setting might compare the weight of an object to a standard weight, and then sort that object based on whether it is heavier or lighter than the standard weight. Iteration means performing the same algorithm on a series of inputs in a loop. An algorithm might iterate over a long list of values, giving output for each of the values using the same algorithmic process.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Create a program that moves through a sequence of instructions when implemented. - Design a program that selects one of several values based on conditions using a flowchart. - Create an algorithm that iterates	Students should <i>investigate</i> these concepts: - What is the difference between instructions and an algorithm? - Think of an algorithm that you use in daily life. What are the input and output values? - What are the pros and cons for using an algorithm to solve a problem?	Students should <i>apply</i> these terms in context: - Algorithm - Conditional Execution - Flowchart - Iteration - List - Sequence

Essential Skills	Essential Questions	Essential Vocabulary
through a list of values producing output.		

CSF.16 The student will implement a program that accepts input values, stores them in appropriately named variables, and produces output.

Context of the Standard
An algorithm that accepts input values, stores them in variables, and produces output builds into a program (CSF.15). Input values can take many forms; clicks, mouse movement, physical orientation, typed words, voice, or images can all be quantified or otherwise digitally represented as input for an algorithm. Variables may be words, abbreviations, or symbols that store data. They allow coders to refer to data in a human-readable way, rather than as long strings of binary or hexadecimal numbers. A variable can be called almost anything, so coders will often use words that represent the data being stored in the variable. For example, a variable that stores weight data from a sensor might be called “weight” or “wgt.” Often, coders will choose names that allow code to be more or less read like a sentence, making it easier for colleagues and collaborators to understand what a given piece of code is supposed to do. Output can take many forms as well. Sometimes algorithms will output data designed to be processed by another computer program, while other times the output is intended to communicate data to human users. In the latter case, output might be visual (i.e., on a screen or light), haptic (e.g., vibrations), or aural (e.g., ringtones). Output is highly important; it makes the invisible work done by the computer visible to human users. Output is also used to help with debugging by giving programmers insights into how to fix programs that do not produce expected outputs.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Develop a program that takes user input which is then stored in a	Students should <i>investigate</i> these concepts: What factors should you consider when naming variables?	Students should <i>apply</i> these terms in context: Concatenation

Essential Skills	Essential Questions	Essential Vocabulary
variable. • Create appropriately named variables to store input data from a user. • Produce output made from various stored variables using both strings and integers (e.g. concatenation).	• Are there types of information/data that can't be stored in a variable? Explain. • Think of an app you use. What type(s) of output does it generate?	• Integer • Input • Output • String • Variable

CSF.17 The student will trace the execution of an algorithm, illustrating output and changes in values of named variables.

Context of the Standard
Code tracing is the practice of analyzing a section of code and identifying the values of critical variables at certain points in the algorithmic process. Tracing is the first step toward debugging or otherwise decoding a computational process; it involves making hypotheses about how the code will change input values during execution, and ultimately involves predicting the output of an algorithmic process (CSF.13). As programmers trace code using a trace table, they will identify how different variables change over time, and then compare that hypothesis to outputs given by the program. If the expected output differs from the actual output, programmers will either update their hypothesis or change the code to reflect their intended design.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Construct a trace table that follows the changes of variables in an algorithm. • Explain the output of a given algorithm, given different input	Students should <i>investigate</i> these concepts: • What situations call for a programmer to use code tracing? • What are other techniques might programmers use to trace the change in variables while creating code?	Students should <i>apply</i> these terms in context: • Algorithm • Code Tracing • Execution of code • Trace Table

Essential Skills	Essential Questions	Essential Vocabulary
values.		

CSF.18 The student will apply the basic operations used with numeric and non-numeric data types in developing programs.

Context of the Standard
Operators are mathematical symbols which ask computers to perform specific tasks given some sort of input. Most operators are fairly straightforward, and they generally fall into one of three categories: arithmetic, Boolean, or logical. Arithmetic operators mostly involve processing numerical data. They take in multiple inputs (i.e., numbers) and generate one output depending on the operator being used. Arithmetic operators include addition (+), subtraction (-), multiplication (*), and division (/). For example, a computer would produce “4” given the input “2 + 2.” The operator (+) tells the computer to sum the two input numbers. Unlike arithmetic operators, which produce numerical outputs, Boolean operators produce an output of either “true” or “false.” Boolean operators include “==”, “<”, “>”, “<=”, “>=”, “!=”, and more. For example, the expression “2 < 4” would return a value of “true,” because 2 is indeed less than 4. The expression “5 == 6” would return a value of “false,” because 5 does not equal six. Boolean operators may be notated differently in different programming languages, so it is wise to consult the language documentation if Boolean operators do not return expected values. Logical operators are used to combine Boolean expressions to allow for more complex processing. Logical operators include “and” (&&) and “or” (), among others. Logical operators return Boolean values (true or false). The expression “5 == 6 && 4 == 4” would return a value of false, because only one of the Boolean expressions returned a value of true. In contrast, the expression “5 == 6 ‘word’ == ‘word’” would return a value of “true” because the “or” operator () returns a value of “true” if any one of the expressions returns “true.” Like Boolean operators, logical operators differ among programming languages (though there is some overlap).

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create an algorithm that changes the value of a named variable using arithmetic operations. • Design a code segment that uses a Boolean operator and a logical operation as part of a conditional statement.	Students should <i>investigate</i> these concepts: • How is Boolean logic used to accomplish a task? • What types of questions should not be answered with a Boolean response?	Students should <i>apply</i> these terms in context: • Arithmetic Operators • Boolean Operators • Conditional Statement • Logical Operators

CSF.19 The student will use predefined functions to simplify the solution of a complex problem.

Context of the Standard
Functions are like variables, except instead of storing data they store lines of code. Programmers use functions to help “chunk” large, self-contained sections of code to help make code more readable and to allow code to be used multiple times without having to re-write it. Most functions take input, and give some sort of output value. Functions almost always have parentheses after them; for example, a function that sums a list of values might look like this: “sum()” To use this function, you would put the values you would like to sum in the parentheses: “sum(2, 6, 7, 2, 1).” Given this input, the function would produce “18” as an output. Many programming languages come with predefined functions that programmers can use without having to worry about the underlying code that makes them work. For example, Math.random() returns a random number between 0 and 1 (in Java). Some programming languages have more predefined functions than others. In addition to the language-specific predefined functions, programmers may download packages that contain even more predefined functions, extending the library of functions made available by the language code base. For example, SciKitLearn is a Python library package that provides predefined functions for data analysis and machine learning.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Solve a problem by calling premade functions in a code segment or program. - Explain the outputs of common premade functions in a given programming language.	Students should <i>investigate</i> these concepts: - Think of a multi-step process in your daily life that has a single name (e.g. make dinner). What are the input and output values of this process? - What roles do functions serve in computer programs?	Students should <i>apply</i> these terms in context: - Function - Library - API

CSF.20 The student will apply simple algorithms to a collection of data.

Context of the Standard
Algorithms are a powerful tool for analyzing data. There are many ways for algorithms to process data; most of these examples are specific to a given domain or problem. For example, a programmer might create an algorithm that processes large amounts of climate data, producing an output that summarizes the impact CO2 emissions have had on global temperatures over time. Sorting algorithms are a simpler example; they take in a large list of data, and return that same data sorted into categories. Sorting algorithms are often part of much larger computational systems, but are also an example of data processing that is accessible to novice programmers building their first algorithm.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe the purpose of a set of code containing an algorithm. - Create a program that applies a	Students should <i>investigate</i> these concepts: What are some reasons to use algorithms for analyzing data, rather than using other means?	Students should <i>apply</i> these terms in context: - Algorithm - Array

Essential Skills	Essential Questions	Essential Vocabulary
simple algorithm to items in a list.	Who is responsible for the actions of a biased algorithm?	- Data - List

CSF.21 The student will create programs

- demonstrating an understanding that program development is an ongoing process that requires adjusting and debugging along the way; and
- using version control to create and refine programs.

Context of the Standard
Version control is a tool that programmers use to 1) keep track of how a software program changes and develops over time, 2) preserve a record of changes made to a program in case it becomes necessary to restore older versions in the event of data loss, and 3) facilitate collaboration among many team members as they all work on the same piece of software. There are many kinds of version control tools, “git” being the most popular. Git keeps track of software as it changes, allowing programmers to revert to previous states if necessary. Additionally, it allows programmers to add in code to a common base independently without overwriting other team members’ contributions. Tools like git help programmers build software iteratively, adding and testing one element at a time.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Create a program while using best practices of the program development cycle (design, code, test, debug). - Use proper version control	Students should <i>investigate</i> these concepts: - Why can't programmers use a tool like Google Docs to effectively collaborate on a code segment? - How might the development process be	Students should <i>apply</i> these terms in context: - Debugging - Incremental - Iterative

Essential Skills	Essential Questions	Essential Vocabulary
methodologies to manage the iterative development process of a individual or collaborative program.	described as linear instead of cyclical?	• Program Development Cycle • Testing • Version Control

Impacts of Computing

CSF.22 The student will use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.

Context of the Standard

Making software is a complex task that requires the help of many people with different areas of expertise. Programmers have technical knowledge about how to write code, but before they begin creating applications designers often create high-level, conceptual models to help guide the programming process. After programmers create a prototype, user experience analysts will test the code to see if it meets the needs of users and solves the problems the designers set out to address. Oftentimes, artists will create assets that fill out the application either by making the software easier to use or more pleasant to interact with.

Software developers use a host of tools and techniques to manage this collaborative process. Version control software (CSF.21) plays a large role along with project management tools (at the time this framework was written, Trello, Monday, and Jira are all popular tools). Designers may also make use of a design or production methodology, a process that guides the development and implementation of ideas during software development. Popular frameworks (at the time this framework was written) include Agile, Lean, and Kanban.

Digital tools and highly structured methodologies create a common language for designers and developers around the world. Development teams are often made up of people from many different geographic locations collaborating asynchronously with a common goal in mind. However, software developers also use these tools to “outsource” certain tasks to people who will work for less money, especially people who live in countries which do not legally protect their workers from exploitation.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Design a simple program collaboratively using either a flowchart or pseudocode. • Create a program collaboratively	Students should <i>investigate</i> these concepts: • How could programmers work collaboratively on a project without being near each other? • How can people in varied backgrounds	Students should <i>apply</i> these terms in context: • Collaboration • Crowdsourcing • Flowchart

Essential Skills	Essential Questions	Essential Vocabulary
using paired programming or another group-focused programming tool.	and professions help identify mistakes in a program?	• Paired Programming • Prototype • Pseudocode

CSF.23 The student will evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.

Context of the Standard
Computing is pervasive in global society, fundamentally shaping economies, relationships, commerce, governance, and culture. There are many positive benefits to computing's impact, along with significant negative consequences. The digital economy is incredibly large, and technology skills are often prerequisites to employment in many fields. Large technology companies often invest in public education to create workers that will fill their expanding need for labor in creative and non-creative roles. Technology companies also mediate many social relationships in contemporary life; this allows people to learn many things using online resources, but also creates space for people to isolate themselves within often toxic or regressive ideologies. The data industry allows users to have access to powerful machine learning tools to help solve seemingly intractable problems, but also violates individuals' privacy and agency. In many ways, technology companies play outsized roles in shaping political and social discourse, having an as-yet-unmeasured impact on cultural discourses within communities.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Discuss the trade-off between technology use and personal privacy rights how that might impact an individual's agency. • Evaluate how a specific computing	Students should <i>investigate</i> these concepts: • Why is it important for technology creators/designers to think about the implications (both positive and negative) their innovations have on the way we live?	Students should <i>apply</i> these terms in context: • Economical Impacts • Ethical Impacts • Cultural Impacts • Personal Impacts

Essential Skills	Essential Questions	Essential Vocabulary
technology impacts society from ethical, global, and/or economic perspectives.		• Social Impacts • Trade-off

CSF.24 The student will explain the beneficial and harmful effects that intellectual property laws can have on innovation, including the impact of open source software.

Context of the Standard
Intellectual property laws are intended to protect creative work from unauthorized replication or reuse. When software is “open source,” it means that the developers have relinquished their exclusive intellectual property rights and have made their work publicly available for other programmers to use, alter, or reuse in different contexts. Intellectual property laws make it difficult to innovate because software developers cannot use the software in question to learn from or build on until the rights expire or the owners release the source code. On the other hand, it is difficult to make a living creating software without some measure of protection against people taking credit for work that is not their own. The software development industry has a complex relationship to intellectual property. Often, the product that technology companies are creating is not the software itself, but access to data from users. In these cases, software development companies are incentivized to release their source code so that more programmers can potentially work for the company in the future (Facebook is a good example). Software developers are also contributors to open source projects as an act of community-building. Open source software is an excellent resource for people who are learning to code, and it makes it easy for people to create tools that solve problems for users or other software developers. When software developers release open-source code, they will often specify a license which explains the terms under which the open source software may be used (e.g., MIT license).

Essential Skills	Essential Questions	Essential Vocabulary
------------------	---------------------	----------------------

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Discuss the benefits and disadvantages of open source software. • Understand the purpose of intellectual property laws.	Students should <i>investigate</i> these concepts: • How does the ownership of intellectual property your work in creating programs and completing projects?	Students should <i>apply</i> these terms in context: • Intellectual Property Laws • Proprietary • Open Source Software

CSF.25 The student will explain the privacy concerns related to the collection and generation of data through automated processes that are not always evident to users.

Context of the Standard
Human-computer interaction generates massive amounts of data. In the contemporary digital economy, companies use this data to support the advertising industry, targeting customers with ads based on their Internet use and history. User data is a commodity that is traded among different companies who either purchase access to data, or raw data itself to use as they train machine learning algorithms to automate different services. Concerns related to privacy include identity theft, cyberstalking, catfishing, fraudulent purchases, etc. Inadvertent data collection makes many people feel as though their privacy is being invaded, particularly if that data is being collected without the user's knowledge. An example of inadvertent data collection can be seen with voice-activated devices. These devices have active microphones which can listen for key terms, but collect ambient audio as well. This audio can be used to gather other data that can be analyzed by developers, businesses, or organizations.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills:	Students should <i>investigate</i> these concepts:	Students should <i>apply</i> these terms in context:

Essential Skills	Essential Questions	Essential Vocabulary
- Investigate a well-known app and examine the data that is collected from users. - Explain how personal data might be used to make decisions that have positive and negative impacts on user groups.	- What exists in your digital footprint based on data collected from the apps and programs you use? - What are the best ways to protect your data privacy?	- Algorithm - Data Analysis - Data Collection - Privacy - User

Computer Science Principles

The Computer Science Principles standards outline the content for a one-year course with an emphasis on the principles underlying computer science. The standards build on the concepts outlined in the Computer Science Foundations standards.

Students in this course will expand their programming skills and begin to think about and analyze their own problem solving process. Students continue to develop the ideas and practices of computational thinking and consider how computing impacts the world.

Teachers are encouraged to select programming languages and environments, problems, challenges, and activities that are appropriate for their students to successfully meet the objectives of the standards.

Programmable computing tools will be used to facilitate design, analysis, and implementation of computer programs. Students for exploring and creating computer programs, facilitating reasoning and problem solving, and verifying solutions should use these tools.

Computing Systems

CSP.1 The student will develop and apply criteria for evaluating a computer system for a given purpose.

Context of the Standard
Computer systems are designed and built to specifications that allow for many different purposes. These computer systems range from general to more specific uses, and are available in a variety of form factors (e.g., laptop, desktop, tablet). When building or purchasing a computer for a specific task, it is important to match the desired performance with the components and their specifications so that the system isn't underpowered or ineffective in running a certain set of applications. It is also important to take into consideration the size and mobility of a computer system, the ability to add or replace parts if needed, and the type and size of storage in a computing system. All computers need a motherboard which houses most of the components inside of a computer system. Built onto the motherboard are the CPU and GPU (processor), RAM and ROM (memory), and hard drive (storage). Other components like the power supply, cooling system, sound card, speakers, monitor, etc. can further add to the complexity of the system but are often secondary to the main needs listed above.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Define the components in a typical computing system. • Design a questionnaire for a client to ascertain what type of system they would need. • Illustrate and diagram the inside of a computing system (laptop or desktop). • Design a computer system created for a specific need or client. (i.e. gaming computer, college student).	Students should <i>investigate</i> these concepts: • How may quantum computing change the typical computer systems we currently use? • What are some other criteria for choosing computing systems beyond technical specifications? Is the method of manufacture or the quality of working conditions a factor?	Students should <i>apply</i> these terms in context: • Central Processing Unit (CPU) • Components • Computer Systems • Graphical Processing Unit (GPU) • Hard Drive • Hardware • Motherboard • Random Access Memory (RAM) • Read Only Memory (ROM)

CSP.2 The student will illustrate ways computing systems implement logic, input, and output through hardware components.

Context of the Standard
Computer systems are complex machines that perform four main tasks: take input, store values, process data, and give output. Typically known as the IPOS model, these tasks are carried out within the blink of an eye in many cases and result in the user receiving the result on a computer's monitor, a blinking light, or some other programmed action. Input is quickly defined as the starting point of a computing task. Information can be gathered through components like sensors, the keyboard and mouse, cameras, etc. or can be requested from stored data on the hard drive. This input is obtained from outside of the computing system and delivered to the computer's components for processing.

Context of the Standard
Processing occurs after input and works symbiotically with the task of storage (temporary). As mathematical, graphical, and system operations are handled by the CPU and GPU, RAM storage, registers, and cache memory holds values to be processed further. The processing that occurs happens because of a written program that performs a series of tasks based on logical operations. Output is usually the final task that occurs in the IPOS model. When data is processed it is transformed into something new that is deemed useful by the software or user it is delivered to the outside world. Typically, the output is the part of the data that normal humans interact with on a daily basis through a monitor/screen but can also include the storage of a file on the computer's hard drive or other storage mechanism.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Diagram a program running a simple mathematical operation using the IPOS model. • Describe the roles of individual components in the processing system.	Students should <i>investigate</i> these concepts: • How do different types of storage enable the processing system to perform complex tasks? • What types of computing systems might not have traditional input and output mechanisms like a keyboard or screen?	Students should <i>apply</i> these terms in context: • Component • Computing System • Input • Logic • Output • Processing • Storage

Networks and the Internet

CSP.3 The student will explain abstractions enabling

- a. one computer to communicate with another over an Internet connection; and
- b. different layers of Internet technology to build on one another.

Context of the Standard

A system of interconnected computing devices is called a network. Computers use sets of rules called protocols to send data over networks. Servers are computers that processes requests from other computers and then distribute information (e.g., webpages, web apps) using packets over the Internet. Routers direct this traffic on the Internet between networks and help to determine the most efficient path for the information travelling between nodes.

For the Internet to be reliable, computers need a way of reporting and resolving communication errors. To achieve these two prerequisites, computers break down messages (e.g., emails, images, music) into smaller chunks of data called packets. A destination address is determined by a request sent to the domain name system (DNS) that supplies the listed IP address from the domain name typed into the URL. These packets make their way to their destination computer separately using different routes determined by routers, after which the receiving computer puts them back together in order following TCP/IP standards. If a packet is missing and the message is incomplete, the receiving computer will request that the sender resubmit the message. All of this happens in accordance with predetermined protocols for communicating over the Internet.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain how computers send messages to one another over the Internet. • Describe how the Internet is a network of networks built with many layers of Internet technology.	Students should <i>investigate</i> these concepts: • How does the initiating computer locate and display webpages from a URL request? • What happens if part of the physical network is broken due to a cut line or other damage?	Students should <i>apply</i> these terms in context: • DNS • IP • Protocol • Network • Router

	• How does the physical Internet span long distances, like across oceans?	• Server • TCP
--	---	---

CSP.4 The student will explain design principles enabling large-scale operation of the Internet to connect devices and networks all over the world.

Context of the Standard
The Internet is a vast web of networks made up of interconnected computing devices (see CSP. 3) that depends on the reliability through its protocols. Reliable communication through these networks depends on all sent information arriving at its destination, bypassing heavy traffic or damaged connections. In order to reliably communicate among networked computing devices, all the devices need to create and interpret these packets based on a universally agreed-upon set of rules that are scalable no matter how many devices are connected. Once these rules are in place, no one has to approve a new website or oversee additions to the network. Anyone can join at any time and successfully communicate with anyone else on the network without deciding on rules for communication in advance. This system is reliable and scalable because each machine only has to keep track of the devices it is directly connected to. A smartphone doesn't need to know the IP addresses of all the switches and routers between it and the server it wants to connect to; it only has to know the IP address of the server (stored in an easily accessible public database) and the IP address of its local wireless access point. The protocols that are developed to govern communication transfer over the Internet are not controlled by any one country or business. Instead, an international group of network professionals and researchers called the Internet Engineering Task Force (IETF), produce highly technical documents that provide a framework for an expanding and smooth running Internet. These relevant documents influence the way people and companies all over the world design, use, and manage the Internet.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe the underlying concepts that support an ever-growing system of networks called the Internet. - Explain how the Internet is designed to be reliable for fast communication across the globe. - Define the purpose and make-up of the IETF.	Students should <i>investigate</i> these concepts: - What would be the advantages and disadvantages of a specific country or entity determining the protocols and architecture of the Internet? - How might the experience of the Internet be different if every country had their own way of communicating over the Internet?	Students should <i>apply</i> these terms in context: - IETF - IP Address - Protocol - Redundancy - Reliability - Scalability

Cybersecurity

CSP.5 The student will explain symmetric and asymmetric encryption as they pertain to messages being sent on a network.

Context of the Standard
Encryption is an important aspect of communicating in a public system like the Internet. Without it, anybody would be able to view personal identifying information of anyone using the system from the packages that they send over the Internet. There are two main types of encryption that is standardly used to protect online communication: symmetric and asymmetric. Symmetric encryption is a form of encryption where a secret key is established to encrypt and decrypt information between two parties over the Internet. The majority of encryption that occurs over the world wide web uses symmetric encryption because it is less demanding on the processing of the involved computer's CPUs and makes fewer demands from the networked systems, therefore becoming more efficient than asymmetric methods. Triple DES, AES, and Blowfish encryption algorithms are the most common of this type of method. Downfalls of this type of encryption includes key exhaustion where every use of the encryption key leaks information that could be used to potentially recreate the original key. Asymmetrical encryption, known as public key encryption, uses a public and private key to secure communication. In this method of communication security, two similarly paired, but not identical (asymmetric), keys are created to encrypt the information. One of

Context of the Standard
the keys is a public key that is available to the public while the other key is kept private. Knowledge of your own private key and the public key are what is needed to decrypt and encrypt messages between two parties. Two of the most familiar forms of asymmetrical encryption is the Diffie-Hellman and the RSA.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Compare and contrast the characteristics of symmetrical and asymmetrical encryption methods. • Explain the importance of encryption for messages sent using the Internet.	Students should <i>investigate</i> these concepts: • Why do new encryption methods need to be increasingly more complex? • What personal identifying information could someone find about you based on your use of the Internet?	Students should <i>apply</i> these terms in context: • Asymmetrical Encryption • Encryption • Personal Identifying Information • Private Key • Public Key • Symmetrical Encryption

Data and Analysis

CSP.6 The student will discuss the methods and tradeoffs of collecting and analyzing data elements on a large scale.

Context of the Standard
Human-computer interaction generates massive amounts of data. In the contemporary digital economy, companies use this data to support the advertising industry, targeting customers with ads based on their Internet use and history. User data is a commodity that is traded among different companies who either purchase access to data, or raw data itself to use as they train machine learning

Context of the Standard

algorithms to automate different services.

There are a myriad of methods for generating user data. Cookies, form responses, and even “time-on-task” data all play a role in the data economy. As the amount of data grows, the processing power required to analyze it increases exponentially. In addition to requiring more processing power as data becomes more complex, data storage has security and privacy implications. In past years, companies have revealed that data breaches are a real risk to user privacy. There have also been cases where companies violate terms of use and record or analyze data users expect to remain private. As such news breaks, companies risk decreasing valuation and damaging their reputation.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explore how large data sets are collected and analyzed. • Explain the tradeoffs experienced by users, programmers, companies, and communities when companies collect large amounts of data.	Students should <i>investigate</i> these concepts: • Who is responsible for ensuring that companies use data responsibly? • Are there kinds of information companies should not be able to gather from users? • Who benefits from advances in machine learning? Who is harmed?	Students should <i>apply</i> these terms in context: • Algorithm • Cookie • Database

CSP.7 The student will select data collection tools and techniques to generate data sets that support a claim or communicate information and implement a relational database to work with data.

Context of the Standard
A relational database is a set of related tables containing data around a particular topic. While small databases can be represented using a tools like spreadsheets or charts, the databases used in computational applications are often vast. Computer programmers and data analysts will often use tools like SQL to store and retrieve subsets of data as a part of the analysis process. Software generates data by prompting input from users, and storing that input in a database. Many times, this database is stored in an Internet-connected server. As users generate data, their devices connect to the database using an API (application programming interface), and send the data over the Internet to be stored and processed in a secure location. The US Census uses such a method to gather information about US demographics over the Internet.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create an information hypothesis from a set of data that they have created/generated. • Design a relational database that is connected to show how input data from an app might be stored.	Students should <i>investigate</i> these concepts: • What are some ways software programs create data? • Who owns data generated by users as they interact with a software program?	Students should <i>apply</i> these terms in context: • Relational database • SQL • API

- CSP.8 The student will discuss how data representations can be interpreted in a variety of forms, convert between data representations, and analyze the representation tradeoffs among various forms of digital information.

Context of the Standard
Software programs and hardware devices are constantly creating data. Smartwatches for example, generate data about movement, location, sleep, heart rate, and many other kinds of user input. In order to interpret the raw information generated by software and hardware devices, programmers create data visualizations—visual representations of data. Some visualizations are traditional (e.g., bar graphs, pie charts, scatter plots), while others are more dynamic, including sounds, haptic feedback, or input which allow users to explore the data interactively. Some programmers will use data to create data-based art, which focuses less on communicating trends and more on creating evocative visual, sounds, or experiences. A visualization may even make use of data being gathered in real time, where sensors or programs update a database and the visualization program updates by pulling new data constantly. For example, a visualization might show domestic flights by pulling data from an air traffic control database. The software sends these requests using an API (application programming interface), a series of specialized commands which allow programs to access a database. The method that designers use to communicate data reveals and conceals different aspects of the raw data. Data visualization is an interpretive process, where the designer or programmer imposes meaning on raw data in order to use the data as a communicative tool. Data visualization research investigates how different visualizations create different impressions among viewers, and generate best practices for ethically and thoughtfully communicating information with data visualization.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Discuss how data representations can aid in the interpretation of collected data. • Examine the tradeoffs created by distilling information through a	Students should <i>investigate</i> these concepts: • How do websites use various data representation methods to promote their bias? • Are there methods that can be used to enable unbiased interpretation of data?	Students should <i>apply</i> these terms in context: • API • Data Representation • Raw Data • Visualization

Essential Skills	Essential Questions	Essential Vocabulary
visualization in comparison to analyzing data from a data set.		

Algorithms and Programming

- CSP.9 The student will design and implement algorithms with
- compound conditional execution; and
 - a variety of loop control structures.

Context of the Standard
An algorithm is a sequence of instructions that takes input, processes that input, and produces output. Compound conditional execution refers to a code snippet that checks input against a nested series of conditional (e.g., ‘if’) statements. These conditional statements evaluate Boolean statements, which return a true or false value depending on the input (e.g., $2 < 1$ returns a ‘false’ value) and run code based on the return value (one set of code for true, one set of code for false). Sometimes, conditional statements contain multiple Boolean statements which are evaluated against one another using logical operators (e.g., ‘and’, ‘or’). Iteration refers to a looped process that executes the same code given a sequence of inputs, like using a ‘for’ loop to iterate through a table or list (PRG.16). Complex iteration using nested loops might involve using nested loops to iterate through a multi-dimensional table.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Create a program that makes use of compound conditional execution and complex iteration. - Predict the result of complex Boolean	Students should <i>investigate</i> these concepts: What is an example of a decision-making process that might be expressed using compound conditional statements (e.g., college entrance criteria, athletic	Students should <i>apply</i> these terms in context: - Conditional - Boolean - Iteration

Essential Skills	Essential Questions	Essential Vocabulary
- and/or logical expressions/conditions. - Trace the values of named variables over the course of a program that uses nested loops.	- team tryouts)? - Why do programmers use loops and iteration rather than writing out long lists of repetitive code?	- Input - Output

CSP.10 The student will solve a complex problem by decomposing it into subtasks consisting of predefined functions and user-defined functions.

Context of the Standard
Functions are like variables, except instead of storing data they store lines of code. Programmers use functions to help “chunk” large, self-contained sections of code to help make code more readable and to allow code to be used multiple times without having to re-write it. Most functions take input, and give some sort of output value. Functions almost always have parentheses after them; for example, a function that sums a list of values might look like this: “sum()”. To use this function, you would put the values you would like to sum in the parentheses: “sum(2, 6, 7, 2, 1)”. Given this input, the function would return “18” as an output. Programmers will often create their own functions to help improve readability or reuse code more efficiently. Many programming languages come with predefined functions that programmers can use without having to worry about the underlying code that makes them work. For example, Math.random() returns a random number between 0 and 1 (in Java). Some programming languages have more predefined functions than others. In addition to the language-specific predefined functions, programmers may download packages that contain even more predefined functions, extending the library of functions made available by the language code base.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create a program that uses existing functions or libraries in a given programming language/framework. • Research, download, install, and use a third-party open-source library or framework. • Create a program that makes use of defined functions.	Students should <i>investigate</i> these concepts: • Why do programmers use libraries? • Can you think of a situation where you would not want to use an existing library or framework (e.g., how do you know a given API is secure)?	Students should <i>apply</i> these terms in context: • Function • Library • Parameter • Return

CSP.11 The student will store, process, and manipulate data contained in a data structure.

Context of the Standard
Data is stored in many different forms. The simplest is a variable, storing only one value. Slightly more complex than a variable is an array, list, or table. These terms (often synonymous, sometimes not depending on the programming language in question) refer to a variable that stores more than one value in an ordered list. For example: suppose you want to store the age of each participant in a conference session. Rather than creating variables for each participant, you might store the ages in a list. Lists are one-dimensional, storing only one value in each position. To access the values stored within a list, you reference the position the value is stored within. When data is stored in a list, it is very easy to access those values and manipulate the values stored therein using a programming structure called an “iterator.” Iterators take a table or list as an input, and perform the same process using each value in the list one at a time. The most common iterator is the “for” loop, but there are many other techniques. In all cases, storing data in tables rather than in variables allows programmers to create abstractions, reusing code for each table value rather than writing code for each variable one at a time.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create programs that use data structures. • Create a program that uses an iterator. • Generate a program that creates and processes data structures given pseudocode examples.	Students should <i>investigate</i> these concepts: • Why do programmers store data in lists or tables instead of individual variables? • Why do programmers use iterators rather than writing everything out? • What are some challenges that arise when debugging iterators?	Students should <i>apply</i> these terms in context: • Data • Iterator • List/Array

CSP.12 The student will systematically debug a program using an appropriate set of data.

Context of the Standard		
Debugging is an essential part of the software development cycle. When programmers debug a software program, they look at the code and use a variety of techniques (e.g., adding print statements, code tracing) to validate their assumptions about how the program is operating. If they can confirm that the program is working as expected, then the programmer can move on to add new features. If the program produces an unexpected output, the programmer will correct the mistake and add new lines of code to help get more detailed debugging information until the program produces the expected output value. Debugging is, fundamentally, the process of making hypotheses and then validating those hypotheses using output from the software program. Code tracing (PRG.8) and using tracing tables are other important aspects of debugging that helps programmers focus in on potentially buggy or difficult to understand code snippets.		
Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Debug a given software program. • Create a software program that includes mechanisms to assist in debugging. • Generate procedures for testing and debugging software.	Students should <i>investigate</i> these concepts: • What are some ways to help avoid a situation where you don't know how to debug your software? • What are some ways to identify and validate the assumptions you make about how your program works as you work on debugging?	Students should <i>apply</i> these terms in context: • Code tracing • Debug • Tracing Tables

Impacts of Computing

CSP.13 The student will explain how computing has impacted innovations in other fields positively and negatively, and enables collaboration between a variety of people.

Context of the Standard
Computing is a ubiquitous force in global society, fundamentally shaping economies, relationships, commerce, governance, and culture. There are many positive benefits to computing's impact, along with significant negative consequences. The modern digital economy is incredibly large, and technology skills are often prerequisites to employment in many fields because of the ways tasks can be performed more efficiently. Computing technology allows for collaboration on many different levels in and outside of the workplace. Email, messaging, and other forms of communications technologies enable immediate communication with another person despite their geographical location. Workflow technologies give opportunities for crowdsourcing difficult or large tasks and real-time collaboration of projects online. Computing advances in communication enables collaboration between a variety of people. Computing has generated entire new sectors of the economy. Data generated through human-computer interaction fuels a vast advertising industry, and companies buy and sell products online as much as they do in person (if not more). Computing itself is an industry, where programmers, manufacturers, and designers participate in a global supply chain that creates smartphones, software,

Context of the Standard
and computers for people to use across many domains in almost every aspect of their lives. As with other global industries, power, capital, and the benefits of computing are spread unevenly. Many workers are exploited by the social and economic conditions catalyzed by computing.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain how computing innovations impact our global society in both negative and positive ways. • Explore the ways in which collaboration is supported by new technologies.	Students should <i>investigate</i> these concepts: • Think of a product that has a large number of users that was created through collaboration and/or crowdsourcing. What are the positive and negative social and economic impacts of the project? • What effects does collaboration have on the development of computing products?	Students should <i>apply</i> these terms in context: • Collaboration • Crowdsourcing • Data • Digital Economy • Innovation

CSP.14 The student will evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society, including the impacts of cloud computing.

Context of the Standard
While it may seem that we live in a world where everyone is connected via technology, there are still individuals, communities, and countries who lack reliable online communication technologies and infrastructure in many places around the globe. This disparity creates a digital divide between those who can access Internet related services and those who cannot. Dominant geopolitical forces leverage technology in ways that are unavailable to communities without equal access to networks, computing power, electricity, education, and human/financial capital. In addition to creating unequal access to computing resources, contemporary computing industries are distributed unevenly across the globe. Creative computing industries (design, software development, game design, hardware design) are supported by manufacturing and programming industries that are part of a global supply chain. Like most other global supply chains, political, social, and economic capital are distributed unevenly. Companies that create software and hardware products generate profits by outsourcing manufacturing and programming labor to countries with highly trained and underpaid labor capital, while concentrating creative power and prestige in their home country. Networks and cloud computing (the practice of providing services powered by servers over the Internet) create many opportunities for people to access data, but it exists in a climate where the data that is available is under strict control by companies that own the data and provide access to data as a service in exchange for private user data. While some people experience incredible improvements in technology and access to services through the Internet, it is important to realize that this access is not evenly distributed.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Examine the global distribution of computing resources and evaluate its impact on equity, access, and	Students should <i>investigate</i> these concepts: - Who made the devices you use every day? What is their job like? - Who made the software you use every	Students should <i>apply</i> these terms in context: - Cloud Computing - Data

Essential Skills	Essential Questions	Essential Vocabulary
- influence on individual locations. - Explore the tradeoffs of cloud computing for collaboration in the context of access and equity.	day? What is their job like?	- Digital Divide - Manufacturing - Software Development - Supply chain

CSP.15 The student will explain how intellectual property concerns affect the tools for and products of computing, including combining existing content to create new artifacts and the impact of open source and free software.

Context of the Standard
Intellectual property laws are intended to protect creative work from unauthorized replication or reuse. When software is “open source,” it means that the developers have relinquished their exclusive intellectual property rights and have made their work publicly available for other programmers to use, alter, or reuse in different contexts. When combining existing content to create new artifacts it is important to understand creative commons licenses and copyright rules. Intellectual property laws can both impede and protect innovation. On one hand, it is difficult to make a living creating software without some measure of protection against people taking credit for work that is not their own. Without this protection, less time would be spent on research and development and creators wouldn't receive the financial compensation for their creations. The software development industry has a complex relationship to intellectual property. Often, the product that technology companies are creating is not the software itself, but access to data from users. In these cases, software development companies are incentivized to release their source code so that more programmers can potentially work for the company in the future (Facebook is a good example). Software developers are also contributors to open source projects as an act of community-building. Open source software is an excellent resource for people who are learning to code, and it makes it easy for people to create tools that solve problems for users or other software developers. When software developers release open-source code, they will often specify a license which explains the terms under which the open source software may be used. The MIT license is a popular example.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain copyright law as it applies to software. • Define “open source.” • Identify different methods of protecting intellectual property as they relate to computing (e.g., license, copyright, patent).	Students should <i>investigate</i> these concepts: • Why do programmers create open-source software? • What are some of the reasons a programmer might decide to make software “closed-source?” • Is it ethical for programmers to create open-source alternatives to paid software (e.g., Inkscape and Corel)?	Students should <i>apply</i> these terms in context: • Intellectual Property • Open Source • License • Patent • Copyright

CSP.16 The student will evaluate the social and economic implications of privacy in the context of safety, law or ethics.

Context of the Standard
Computing has had many impacts on contemporary social and economic climates. Computing has changed the way communities form and sustain themselves, has created new economies and industries, and has fundamentally changed the way governments and societies do the work of creating and implementing public policy. The digital economy is built on the acquisition and commodification of user data. Users pay for services with their data, which companies use to fuel the digital advertising industry. In this economic context, privacy has evaporated and has been replaced with confidentiality. Users can be sure that their Internet histories, interactions on various social media platforms, and their messages are part of a dataset being used by the companies offering these services to help generate profits. When companies collect user data, users lose control over who has their personal information and what they can do with it. Even in the absence of data breaches or leaks, users cannot assert control over their data. As the data economy evolves, companies are developing machine learning algorithms trained by massive amounts of user data. Machine learning algorithms are subject to “algorithmic bias,” where bias in data or in algorithm implementation cause the algorithm to produce biased results. These machine learning algorithms are being applied across many domains, and many

Context of the Standard
researchers question the ability of corporations and governments to create tools that treat users ethically in the presence of economic and social pressures to do otherwise.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Discuss how data is used in economic and social contexts. • Identify impacts of computing on political, social, and economic contexts. • Discuss how the data economy impacts individuals.	Students should <i>investigate</i> these concepts: • Is privacy a right? • Who owns user data generated through interaction with software applications? • Who is responsible for people who misuse programs and exploit users?	Students should <i>apply</i> these terms in context: • Data Economy • Machine Learning Algorithm • Algorithmic Bias

Computer Science Programming

The Computer Science Programming standards outline the content for a one-year course with an emphasis on computer programming in a text-based language. The standards build on the concepts outlined in the Computer Science Foundations and Computer Science Principles standards.

This course continues the study of computer programming and prepares students to write programs of increasing complexity to solve problems of personal interest and professional relevance in a variety of technical fields. Additionally, this course provides the knowledge and experience to prepare students for further studies in computer science.

Teachers are encouraged to select text-based programming languages and environments, problems, challenges, and activities that are appropriate for their students to successfully meet the objectives of the standards. The majority of this course will address Algorithms and Programming. While the standards below do not include new content related to Computing Systems or Networks and the Internet, they may be used to provide context for additional exploration of these topics.

Cybersecurity

PRG.1 The student will describe and use best practices of program development that make some common flaws less likely and explain how this improves computer security.

Context of the Standard
Programming best practices exist so that programmers can create software that is efficient and safe. These best practices are built on decades of mistakes and oversights, and new best practices are being generated all the time. Some examples of best practices include: validating user input, avoiding direct command-line access, organizing program code into different files, and storing data in a different place than executable code. These best practices correct common vulnerabilities (e.g., command line injection, SQL injection, cross-site scripting) and help make code more readable.

Essential Skills	Essential Questions	Essential Vocabulary
------------------	---------------------	----------------------

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create software programs that meet basic requirements for security based on best practices.	Students should <i>investigate</i> these concepts: • What are some examples of times when programs were misused by users to exploit other users? • How should developers respond when it becomes clear that something they made is being misused?	Students should <i>apply</i> these terms in context: • Computer Security • Program Development • Validate • Vulnerability

Data and Analysis

PRG.2 The student will create programs that model the relationships among different elements in collections of real-world data.

Context of the Standard
One way computer programs are useful is that they automate repetitive tasks. One example of this use case is using a program to process and model (visualize) large amounts of data. Software programs and hardware devices are constantly creating data. Smartwatches for example, generate data about movement, location, sleep, heart rate, and many other kinds of user input. In order to interpret the raw information generated by software and hardware devices, programmers create data visualizations—visual representations of data. Some visualizations are traditional (e.g., bar graphs, pie charts, scatter plots), while others are more dynamic, including sounds, haptic feedback, or input which allow users to explore the data interactively. Some programmers will use data to create data-based art, which focuses less on communicating trends and more on creating evocative visual, sounds, or experiences. A visualization may even make use of data being gathered in real time, where sensors or programs update a database and the visualization program updates by pulling new data constantly. For example, a visualization might show domestic flights by pulling data from an air traffic control database. The software sends these requests using an API (application programming interface), a series of specialized commands which allow programs to access a database.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create a program that models the relationship among different elements in a collection of data. • Create a visualization given a data set. • Use a data visualization to draw conclusions about the underlying raw data.	Students should <i>investigate</i> these concepts: • What types of data visualizations can you find on the Internet that is refreshed in real time (e.g., weather, traffic)? • What type of daily data could a school create, and how might you visualize that data?	Students should <i>apply</i> these terms in context: • API (Application Programming Interface) • Data Set • Data Visualization • Input • Interactive Data • Model • Raw Data

PRG.3 The student will translate numbers between machine representations and human-accessible representations.

Context of the Standard
Computers do not store data in human-readable formats. While people store data using written language, computers store data using a variety of systems that differ across use cases. At a fundamental level, these numbers are binary, consisting of zeros and ones. Sometimes, computers will store data in hexadecimal formats, or in other more verbose formats like JSON (Javascript Object Notation). Programmers are often tasked with the job of translating these machine representations into human-readable data by creating programs that take data as input and create language as output. For example, an online attendance database takes machine data and outputs a table or other visualization letting teachers and administrators know which students are in school and which ones are absent.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify different data formats (e.g., binary, hexadecimal, JSON) in context. - Translate given machine data into human language. - Store human input in a machine-readable format.	Students should <i>investigate</i> these concepts: - What are some reasons machines store data in a format other than language? - Why do you think there are so many different kinds of data?	Students should <i>apply</i> these terms in context: - Binary - Hexadecimal - Human-accessible Representation - Machine Representation

Algorithms and Programming

PRG.4 The student will design and implement a program working individually and in teams using a text-based language.

Context of the Standard
Programming is how people create new tools and experiences with computers. Programming languages are the creative medium of computer programming, allowing people to solve problems by generating new tools that run on computing devices. Many times, programmers work in teams by dividing the work among the team members. For example, each team member might work on one page of the application. This allows individuals to focus on making one aspect of the program as good as it can be, without worrying about the details of the rest of the application. After everyone has finished their delegated task, the team will come together and collaboratively combine their pieces into a cohesive whole.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills:	Students should <i>investigate</i> these concepts:	Students should <i>apply</i> these terms in context:

Essential Skills	Essential Questions	Essential Vocabulary
- Organize a plan to develop a program individually and as part of a team. - Create a program that solves a problem using text-based programming. - Reflect on a completed program to identify possible future opportunities for improvement.	- What are some of the benefits and drawbacks to working alone or in a team developing software? - What are some strategies for dividing labor when working on a software development team?	- Compile - Execute - Integration - Programming Language - Software Application

PRG.5 The student will explain the software life cycle and how it applies to iterative development processes.

Context of the Standard
Creating new software involves several steps, including designing, prototyping, testing, and debugging. Some programmers use the notion of a minimum viable product, or MVP, as a guide to help focus their creative efforts. An MVP is the simplest version of a tool that solves a problem. For example: if someone wants a tool to quickly move from place to place over long distances, a car might be a final solution. A car, however, is a very complex machine and may be difficult to design, prototype, and test in a reasonable amount of time. A minimum viable product for this problem might instead be a skateboard. It solves the problem, but is much easier to design, prototype, and test. After the skateboard is successful, the team might add on to it to solve the next problem (e.g., steering needs to be improved, so add a handle and make a scooter). This iterative process of designing, prototyping, testing, and debugging ensures that programmers and designers avoid wasting time and resources on a complex solution to a simple problem. Creating a minimum viable product is an excellent way of ensuring that programmers are frequently testing, debugging, and refining their software as they work toward short-term, iterative goals. After creating the initial MVP, programmers continue to build upon their work iteratively until the software meets its design specifications and is ready for release. After release, the software enters a maintenance stage, where programmers fix issues that come up as consumers use the product. Eventually, the software becomes obsolete due to technological changes and the software enters its final stage of life, where programmers stop providing updates and

Context of the Standard
support as the program is retired.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Explain the software life cycle. • Describe the process for designing and creating an original software program. • Given a problem, generate a minimum viable product.	Students should <i>investigate</i> these concepts: • What is the minimum viable product of a piece of software you use a lot? • Why might you test a product while building it rather than waiting until you finish the product? • What are some of the differences between programming alone versus programming with a team?	Students should <i>apply</i> these terms in context: • Iterative process • Minimum Viable Product • Program • Prototype

- PRG.6 The student will design and implement an algorithm
- a. with compound conditional execution, and analyze and evaluate complex Boolean conditions; and
 - b. using complex iteration, including nested loops.

Context of the Standard
An algorithm is a sequence of instructions that takes input, processes that input, and produces output. Compound conditional execution refers to a code snippet that checks input against a nested series of conditional (e.g., ‘if’) statements. These conditional statements evaluate Boolean statements, which return a true or false value depending on the input (e.g., $2 < 1$ returns a ‘false’ value) and run code based on the return value (one set of code for true, one set of code for false). Sometimes, conditional statements contain multiple Boolean statements which are evaluated against one another using logical operators (e.g., ‘and’, ‘or’). Iteration refers to a process that executes the same code given a sequence of inputs, like using a ‘for’ loop to iterate through a table or list

Context of the Standard
(PRG.16). Complex iteration using nested loops might involve using nested loops to iterate through a multi-dimensional table.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create a program that makes use of compound conditional execution and complex iteration. • Predict the result of complex Boolean and/or logical expressions/conditions. • Trace the values of named variables over the course of a program that uses nested loops.	Students should <i>investigate</i> these concepts: • What is an example of a decision-making process that might be expressed using compound conditional statements (e.g., college entrance criteria, athletic team tryouts)? • Why do programmers use loops and iteration rather than writing out long lists of repetitive code?	Students should <i>apply</i> these terms in context: • Algorithm • Boolean • Conditional • Input • Iteration • Output

PRG.7 The student will implement programs that accept input from a variety of sources and produce output based on that input.

Context of the Standard
Input and output can take many forms in the digital realm. A smartphone is an example of a device that utilizes multiple inputs. These may include: location, touch, orientation, temperature, pressure, visual (through the cameras), aural (through the microphone), and many other kinds of input create numerical data which is fed into software. The device takes these inputs and generates output to users, letting them know what processes are running on the device and prompting users to create more input data. The relationship between input and output is called a feedback loop. Users create input, and receive output which then prompts more input in a cycle that eventually meets users goals. If the feedback loop is smooth, users might say the program is intuitive or fun to use. If the feedback loop is clumsy or choppy, users might choose to stop using the program.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Identify methods of input being processed in an example program (e.g., Instagram). - Create a program that takes user input and produces output based on that input. - Diagram the feedback loop of a software program.	Students should <i>investigate</i> these concepts: - Why do programmers want to create a feedback loop for users? - Are highly effective feedback loops good or bad for people? Give an example. - What are some of the visible and invisible inputs being processed by software you use in school?	Students should <i>apply</i> these terms in context: - Feedback loop - Input - Output

PRG.8 The student will trace the execution of iterative and recursive algorithms, illustrating output and changes in values of named variables.

Context of the Standard
Code tracing is the practice of analyzing a section of code and identifying the values of critical variables at certain points in the algorithmic process. Tracing is the first step toward debugging or otherwise decoding a computational process; it involves making hypotheses about how the code will change input values during execution, and ultimately involves predicting the output of an algorithmic process. As programmers trace code, they will identify how different variables change over time, and then compare that hypothesis to outputs given by the program. If the expected output differs from the actual output, programmers will either update their hypothesis or change the code to reflect their intended design. Iterative algorithms are processes that are applied in the same way to a series of data (PRG.6 & 16). Recursive algorithms are processes that use themselves as part of their process. Recursion can seem complex; the classic example of a recursive algorithm is one that solves the factorial of a given input. When given a number as input, the program checks to see if the number is 1 or greater than 1. If the number is 1, the program returns the number '1' as its output. If the number is greater than one, it runs the factorial algorithm using the input number minus 1 and returns the output multiplied by the input number. For example, say the user inputs

Context of the Standard
the number '2' to the factorial algorithm. The number is greater than one, so the program will run the factorial program using 1 as input (the input number minus 1), and multiply the output by 2 (the input number). Because the input number is 1, the factorial algorithm returns '1', and so the final output is '2' (1 times 2).

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Generate working recursive or iterative algorithms given pseudocode. • Create programs that makes use of iterative or recursive algorithms. • Debug iterative and recursive algorithms.	Students should <i>investigate</i> these concepts: • Why do programmers use recursion rather than just writing the whole thing out? • What is the difference between recursion and iteration?	Students should <i>apply</i> these terms in context: • Algorithm • Iteration • Input • Output • Recursion • Variable

- PRG.9 The student will perform complex computations
- on numbers, including modular division and random number generation; and
 - on strings, including substring manipulation and processing individual characters.

Context of the Standard

Programming often involves manipulating numbers and strings (data as text) to prepare it for processing or as part of an algorithmic process. Modular division involves using the modulo operator (%) to find numbers that divide evenly. The modulo returns the remainder after division of two numbers; $6 \% 3$ returns 0, while $6 \% 4$ returns 2. Given n , if $n \% 2$ equals 0 then n is an even number.

Random number generation is used to create normally distributed data sets, or to create procedural processes based on non-repeating sequences. Many random number-generating functions are pseudo-random, meaning they use a random seed that can be set or reset. Sequences of random numbers using the same seed will be identical (though still random and normally distributed) from run to run. This is to say that while pseudo-random number sets are comprised of random numbers, they are comprised of the same random numbers in the same order provided they are based on the same seed.

String manipulation involves taking a string apart and using its constituent pieces as data in a program. String manipulation can be used to sanitize user input (e.g., removing all articles from a string of written text) or to find keywords in a given text. Strings can also be combined (called concatenated) with other strings or numbers to construct sentences.

Programmers can also use regular expressions (RegExs) to find specific patterns of characters in strings. There are many different kinds of regular expressions, and many ways to use them which go beyond the scope of this document.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: Create a program that makes use of random number generation, modular math, and/or substring manipulation.	Students should <i>investigate</i> these concepts: - How could modular math be used in encryption? - What are some possible uses for	Students should <i>apply</i> these terms in context: - Modulus - Randomization

Essential Skills	Essential Questions	Essential Vocabulary
Explain how and why an example program uses modular math, substring manipulation, and/or random number generation.	- substring manipulation? - Why might a program need to create random numbers?	- Regular Expression - String - Substring - Variable

PRG.10 The student will demonstrate an understanding of different data types by using appropriate constructs to convert between them when appropriate.

Context of the Standard
Data can be stored in many different forms in a computer program. These data types have different uses, and understanding the relationships between each form of data is very important. Some examples include referential text, strings, numerical values, and Boolean values. Referential text is any word in code that refers to some sort of data. Referential texts include variable names, function names, list names, etc. These words derive their meaning from the data they represent. Referential text is different from a string, which is text data. Strings can be stored in referential text without changing the value of the string; if a programmer types “print(name),” the program will output the value stored in the variable ‘name’. If the program reads “print(‘name’),” the program will output the word ‘name’ rather than the value stored in the variable ‘name’. The quotation marks tell the computer that the data within the “print()” function is a string, and does not represent some other kind of data. Boolean values (i.e., true and false) are not strings (text data) or numerical data; they are their own separate type of data.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Predict the different types of data being used in a program. - Validate different types of data using	Students should <i>investigate</i> these concepts: What is the difference between strings and numerical values? Why do programming languages treat these data	Students should <i>apply</i> these terms in context: - Boolean - Function

Essential Skills	Essential Questions	Essential Vocabulary
conditional statements and/or debugging methods. Create a program that uses variables (Boolean, string, numerical), lists, and function definitions.	types differently? What are some ways of keeping track of the data types for different variables in a program?	- Integers - Numerical Values - String - Variable

PRG.11 The student will analyze a large-scale computational problem, identify generalizable patterns, and implement a solution.

Context of the Standard
Computational problems are varied and may exist in any domain in or outside of computer science. Often, computational problems emerge out of real-world problems that programmers seek to solve using software. Solving computational problems involves finding out what steps the computer must execute in order, and then finding patterns in how the computer should process data. Once the programmer identifies patterns, they can abstract the computational procedure and allow the code to apply to multiple contexts or become more efficient as it processes large amounts of data. For example, imagine that a programmer tries to create software that will sort objects according to likeness. At first, the programmer needs to figure out a way for the computer to look at a single object for a single attribute (e.g., color) and apply a label. To prototype this early iteration of the program, the programmer might feed the program one object at a time to see if it reliably applies the appropriate label to it. Once this stage works, the programmer's goal is to apply this same code for sorting objects based on a single attribute (e.g., color) to the problem of sorting objects based on multiple attributes (e.g., color, size, and mass). The code should be generally the same for all three of these attributes; abstracting the process of analyzing objects from one use case (color) to multiple use cases (color, size, mass) solves a computational problem. The next stage might involve adding an option to analyze many objects rather than one at a time.

Essential Skills	Essential Questions	Essential Vocabulary
------------------	---------------------	----------------------

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Analyze a computational problem to identify patterns and generate pseudocode. - Create a program that is useful across many use cases given a program that is useful in one context (i.e., abstract the implementation of a given program).	Students should <i>investigate</i> these concepts: - Why do programmers want to create abstractions of their solutions to computational problems? - How do you know when a given function or solution needs to be abstracted?	Students should <i>apply</i> these terms in context: - Abstraction - Computational Problem - Function - Iterative design - Real-world Problem

PRG.12 The student will implement an algorithm that uses existing functions and accesses existing libraries or APIs to satisfy its requirements.

Context of the Standard
Functions are like variables, except instead of storing data they store lines of code. Programmers use functions to help “chunk” large, self-contained sections of code to help make code more readable and to allow code to be used multiple times without having to re-write it. Most functions take input, and give some sort of output value. Functions almost always have parentheses after them; for example, a function that sums a list of values might look like this: “sum().” To use this function, you would put the values you would like to sum in the parentheses: “sum(2, 6, 7, 2, 1).” Given this input, the function would produce “18” as an output. Many programming languages come with predefined functions that programmers can use without having to worry about the underlying code that makes them work. For example, Math.random() returns a random number between 0 and 1 (in Java). Some programming languages have more predefined functions than others. In addition to the language-specific predefined functions, programmers may download packages that contain even more predefined functions, extending the library of functions made available by the language code base. An API is a library of functions that accesses a database stored outside of the program. Many databases have APIs that allow programmers to access data over the Internet to use in different programming applications.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Research, download, install, and use a third-party open-source library or framework. • Create a program that uses existing functions or libraries in a given programming language/framework.	Students should <i>investigate</i> these concepts: • Why do programmers use libraries rather than writing everything from scratch? • Can you think of a situation where you would not want to use an existing library or framework (e.g., how do you know a given API is secure)?	Students should <i>apply</i> these terms in context: • API • Function • Library

PRG.13 The student will write functions, both with and without parameters, and both with and without return values, that represent abstractions useful to the solution of a larger problem.

Context of the Standard
Functions are keywords that store chunks of code (PRG.12). Many times, as programmers write code that performs a specific function, it becomes necessary to abstract that portion of code such that it can be easily reused throughout a software program with different parameters. Abstraction allows programmers to hide all but the relevant data in order to reduce complexity and increase efficiency. For example, a function that produces the mean of a given set of 12 numbers could be abstracted so that it gives the mean of a given set of any number of values. In this case, the function would take a set of numbers as input, count the number of values, and divide the sum of the values by the quantity. In this way, the function becomes useful in many use cases rather than just when the given set of numbers is 12 values long. In the above example, the function returned a value equal to the mean of the given set of values. Functions do not always return values; sometimes, they will change the value of existing global or local variables already declared in the code. For example, one might write a function that sets the coordinates of a graphical object, or one that tells the computer to wait for a given length of time before continuing to execute the program.

Context of the Standard
Functions make it easier for programmers to create complex systems. Once a function has been abstracted, programmers can use it without thinking about the details of how the function works. By creating many different interacting functions, programmers can create modular systems that generate emergent complexity.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create a program that makes use of defined functions with and without parameters/return values. • Debug a program that makes use of defined functions with and without parameters/return values.	Students should <i>investigate</i> these concepts: • Why do programmers write functions rather than writing everything out in order? • What are some strategies for making it easier to debug defined functions?	Students should <i>apply</i> these terms in context: • Abstraction • Function • Parameter • Return

PRG.14 The student will create programs demonstrating an understanding of the interactions between classes in object-oriented design, and by implementing classes with instance data and methods to satisfy a design specification.

Context of the Standard
Object-oriented programming (OOP) is a common programming paradigm that makes use of objects and methods to perform complex tasks. Objects are abstract, discrete units of data which have values called attributes. These attributes are persistent, and unique to the object. Methods are special functions written within a class to define actions that data type are capable of. Classes are templates that create objects. One example of an object in a programming context is a bank account. The bank account itself is abstract; you cannot hold or touch it. Instead, it is a data structure that has attributes: a balance, an overdraft limit, or perhaps an accruing interest rate. The class that was used to create this bank account object also created all the other bank accounts for all other bank customers. All bank accounts have the same attributes, though the values of these attributes are different for each customer. When you deposit money into the bank account, the computer program might execute a method that changes the balance attribute based on how much money you deposited. The deposit method only acts on the specified object even as the program reuses the same code for every deposit. Object oriented programming is popular because it structures computer programs such that it matches human experiences of working with units of data (i.e., the objects and materials we work with in everyday life). It allows programmers to think in abstract terms and create programs that make use of similar logical patterns to those humans experience in everyday life. OOP is especially popular in videogame design, where many unique data structures must interact according to global rules (e.g., physics).

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Define object, class, and method. • Create a program that makes use of objects and methods given predefined classes. • Create a class, and implement it in a	Students should <i>investigate</i> these concepts: • Think of household pets; do fish, dogs, cats, and birds belong in the same class? Why or why not? • What would the attributes of a “plant” class be? What are some methods that	Students should <i>apply</i> these terms in context: • Attribute • Class • Instance • Method

Essential Skills	Essential Questions	Essential Vocabulary
program.	you might include?	Object

PRG.15 The student will use code written by others by reading the documentation and incorporating it into their programs using proper citation of the reused code.

Context of the Standard
Programmers almost never create software programs without using examples from other programmers. These examples are often posted on online forums where programmers ask for help solving problems or in the form of open-source programs. It is incredibly important that programmers are able to navigate the social space of programming to expand their knowledge of different solutions to common problems. In order to facilitate knowledge sharing, programmers document their code by adding explanatory comments and documentation. Comments are small notes within code that explain portions of programs so that other programmers can learn how the software works. Documentation is often a longer text separate from the code, containing information about how to use the program and how to apply the code within the program to other contexts. Oftentimes, programmers will also include licensing information in documentation, letting other programmers know how they intend their program to be used. Some of these licenses require that programmers give credit to the creators of open-source software used in the application, while other licenses are more permissive.

Essential Skills	Essential Questions	Essential Vocabulary

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Research, install, and use third-party libraries or frameworks in an original program. • Find and use examples from online resources to solve a problem in an original program. • Write documentation for an original program.	Students should <i>investigate</i> these concepts: • Why is creating detailed documentation so important? • What are some examples of bad or unhelpful documentation? What makes for good documentation?	Students should <i>apply</i> these terms in context: • Citation • Documentation • Forum • License

PRG.16 The student will read and store data in 1D and 2D collections, and design and implement algorithms to process and manipulate those collections.

Context of the Standard
Data is stored in many different forms. The simplest is a variable, storing only one value. Slightly more complex than a variable is an array or list. These terms (often synonymous, sometimes not depending on the programming language in question) refer to a variable that stores more than one value in an ordered list. For example, suppose a program is designed to store the age of each participant in a conference session. Rather than creating variables for each participant, the program might store the ages in a list. Lists are one-dimensional, storing only one value in each position. To access the values stored within a list, reference the position the value is stored within (i.e., the fifth value in the “ages” table can be found at “ages[5]” or “ages[4],” depending on if the programming languages starts counting table values at “0” or “1”). Two-dimensional data structures, like matrices, are created by storing lists at each position in a larger list. When data is stored in a list, it is very easy to access those values and manipulate the values stored therein using a programming structure called an “iterator.” Iterators take an array or list as an input, and perform the same process using each value in the list one at a time. The most common iterator is the “for” loop, but there are many other techniques. In all cases, storing data in arrays rather than in variables allows programmers to create abstractions, reusing code for each table value rather than writing code for each

Context of the Standard
variable one at a time.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Create a program that makes use of 1D and/or 2D data structures. • Create a program that makes use of an iterator. • Generate a program that creates and processes 1D or 2D data structures given pseudocode examples.	Students should <i>investigate</i> these concepts: • Why do programmers store data in lists or tables instead of individual variables? • Why do programmers use iterators rather than writing everything out? • What are some challenges that arise when debugging iterators?	Students should <i>apply</i> these terms in context: • Data • Iterator • List • Array

PRG.17 The student will adapt classic algorithms for use in a particular context and analyze them for effectiveness and efficiency.

Context of the Standard
Algorithms, in addition to being a sequence of instructions in code, are also logical procedures for processing data according to rules. Because algorithms are not programming-language-specific or domain-specific, many algorithms are used across domains to solve different problems. Such algorithms include (but are not limited to): insertion sort, selection sort, bubble sort, merge sort, quick sort, binary search, A*, minimax, and many more. Each algorithm has different use cases across different domains. For example, minimax can be used to create an unbeatable tic-tac-toe game, and A* is used to find the shortest path between two given points in a field of obstacles.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Generate a program designed around a commonly used algorithm given pseudocode. • Describe the effectiveness and efficiency of different methods from similar types of algorithmic solutions (e.g., sorting, searching).	Students should <i>investigate</i> these concepts: • Why are algorithms often represented in pseudocode? • How might you evaluate the effectiveness or efficiency of an algorithm?	Students should <i>apply</i> these terms in context: • Algorithm • Effectiveness • Efficiency • Pseudocode

PRG.18 The student will develop and use a series of test cases to verify that a program performs according to its design specifications, including edge cases and all branches.

Context of the Standard
When creating a computer program that performs a task for users, it is important that programmers consider all possible use cases. If a program performs well under narrow circumstances, but crashes if users misuse or misunderstand the program, it is likely the program will not perform according to its design specifications. When testing programs, designers will set up a number of scenarios that testers will work through, testing how the program performs in many different use cases. “Edge cases” refer to use cases where the program may not be designed to accommodate the case, or a use case at the minimum or maximum operating parameter. For example, a speaker distorts output audio at high volumes. Often times, the device will notify the user that the operating parameter (i.e., volume) is set too high and the user should correct the problem. Branches refer to the different paths users can take as they accomplish their goals: the series of pages or sections of the application they see as they go about looking for information or completing a task. User experience design is the field that investigates how to create programs that are easy to use and intuitive for users. User experience designers conduct tests where users attempt to complete goals, and analyze data from those sessions to make recommendations to programmers as to how to improve the software in future iterations.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Generate a set of criteria by which to measure the success of a given program. • Create a set of criteria to measure the success of an original program. • Conduct user testing with student-created software.	Students should <i>investigate</i> these concepts: • Why do programmers and designers conduct user testing? • When in the development process should programmers start testing software with users? Why?	Students should <i>apply</i> these terms in context: • Branches • Edge case • Testing • Use case

PRG.19 The student will, through the process of code review, evaluate a program's correctness, readability, usability, and other factors.

Context of the Standard
Code review is the process of reading through a program to understand the processes it goes through as it completes tasks, and how efficiently it solves problems. It is important that programs are correct (performs according to design specifications), readable (code is logically written and easy to understand), and usable (it is easy to understand what the program does, and easy to make the program perform as expected). There are many different ways to evaluate a computer program, and the methods designers use to evaluate programs will vary depending on the intended user profile, the complexity of the task the program should perform, and security or privacy issues embedded in the use case. Some techniques for improving code readability include creating debugging messages that prompt programmers to add required inputs, call functions with correct parameters, or otherwise correct misuse. Sometimes, improving code readability can be as simple as creating variable or function names that show the reader what the purpose of the variable or function is (a variable that stores the amount of time that has passed since the program began should be called 'timer' or 'clock').

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Trace the execution of student-created programs. • Generate criteria for readability and usability given good and bad examples. • Evaluate student-created programs based on original criteria.	Students should <i>investigate</i> these concepts: • What are some things code reviewers should look for when evaluating code? • Who should be responsible for software that violates users privacy, or that creates unjust or toxic social impact?	Students should <i>apply</i> these terms in context: • Code review • Design specifications • Readability • Usability

PRG.20 The student will use a systematic approach and debugging tools to independently debug a program.

Context of the Standard
Debugging is an essential part of the software development cycle. When programmers debug a software program, they look at the code and use a variety of techniques (e.g., adding print statements, code tracing) to validate their assumptions about how the program is operating. If they can confirm that the program is working as expected, then the programmer can move on to add new features. If the program produces an unexpected output, the programmer will correct the mistake and add new lines of code to help get more detailed debugging information until the program produces the expected output value. Debugging is, fundamentally, the process of making hypotheses and then validating those hypotheses using output from the software program. Code tracing (PRG.8) and using tracing tables are other important aspects of debugging that helps programmers focus in on potentially buggy or difficult to understand code snippets.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: • Debug a given software program. • Create a software program that includes mechanisms to assist in debugging. • Generate procedures for debugging software.	Students should <i>investigate</i> these concepts: • What are some ways to help avoid a situation where you don't know how to debug your software? • What are some ways to identify and validate the assumptions you make about how your program works as you work on debugging?	Students should <i>apply</i> these terms in context: • Code tracing • Debug • Software Development Cycle • Tracing Tables

Impacts of Computing

PRG.21 The student will identify some of the practical, business, and ethical impacts of open source and free software and the widespread access they provide.

Context of the Standard

Open source software is an important aspect of software development communities. Programmers create software, and then release their work with annotations and documentation so that other programmers can learn from their work and create new things based on the open source program.

When software is “open source,” it means that the developers have relinquished all or part of their exclusive intellectual property rights and have made their work publicly available for other programmers to use, alter, or reuse in different contexts. Software developers also contribute to open source projects as an act of community-building. Open source software is an excellent resource for people who are learning to code, and it makes it easy for people to create tools that solve problems for users or other software developers. When software developers release open-source code, they will often specify a license which explains the terms under which the open source software may be used. The MIT license is a popular example.

Essential Skills	Essential Questions	Essential Vocabulary
Students should <i>demonstrate</i> these skills: - Describe the meaning of a program being “open source.” - Identify examples of open-source software for software development. - Explain the reasons for open-source software and its impact on software development.	Students should <i>investigate</i> these concepts: - Why do programmers create open-source software? - What are some of the reasons a programmer might decide to make software “closed-source”? - Is it ethical for programmers to create open-source alternatives to paid software (e.g., Inkscape and Corel)?	Students should <i>apply</i> these terms in context: - Intellectual Property Rights - License - Open-source - MIT License

Appendix A

Grade K

Term	Definition
Algorithm	A list of steps to finish a task
Appropriate	Suitable use
Computing Device	An electronic device that can store and receive information
Data	Individual facts and information
Design document	Written description of a program or story
Desktop computer	A stationary computing device
Digital Citizenship	Responsible behavior with technology
Digital Safety	Protecting yourself while using devices
Email	Program used to communicate online
Graphic organizer	Visual display of the relationship between terms, objects, or ideas
Hardware	The physical parts of a computer
Internet	A network of interconnected networks
Keyboard	An input device used to enter text on a screen
Laptop computer	A mobile computing device

Term	Definition
Mouse	An input device used to move items on the screen and navigate
Object graph	Physically organizing objects according to a category
Pattern Matching	Finding similarities between things
Persistence	Trying again and again, even when something is very hard
Personal information	Specific information about a student
Picture graph	Pictorial display of data with symbols, icons, and pictures to represent different quantities
Printer	An output device that displays the result on paper
Program	An algorithm that has been coded into something that can be run by a machine
Protect	To keep safe
Reboot	To turn off the device and turn it back on
Sequence	An ordered set of instructions
Software	The programs that run on the computer
Storyboard	Visual organization of sequential pictures that outline a program or story

Term	Definition
Table	Information (such as numbers and descriptions) arranged in rows and columns.
Tablet	A handheld computing device that primarily uses touch input
Trackpad	An input device used to move items on the screen and navigate
Troubleshoot	Actions taken to solve a problem with hardware or software
WiFi	Technology that uses radio waves to provide Internet activity

Grade 1

Term	Definition
Appropriate	Suitable use
Artifact	An object made by a person
Attribute	Physical description of an object (e.g., color, shape, size)
Author	The creator of a book, image, song, or object
Block-based programming language	Environment to create a program by fitting together command blocks in a sequence

Term	Definition
Computing device	An electronic device that can store and receive information
Data	Individual facts and information
Debug	Find and fix problems in a program
Desktop computer	A stationary computing device
Digital citizenship	Responsible behavior with technology
Digital safety	Protecting yourself while using devices
Email	Program used to communicate online
Illustrator	The creator of an image or other visual piece
Keyboard	An input device used to enter text on a screen
Laptop computer	A mobile computing device
Loop	A set of actions repeated until a condition is met
Mobile phone	A small handheld computing device that primarily uses touch input to
Mouse	An input device used to move items on the screen and navigate
Pattern matching	Finding similarities between things
Personal information	Specific information about a student
Prediction	Making a guess of what will happen based on current facts

Term	Definition
Printer	An output device that displays the result on paper
Program	An algorithm that has been coded into something that can be run by a machine
Protect	To keep safe
Repeat	To perform an action or set of actions multiple times in a row
Sequence	An ordered set of instructions
Social media	Applications that allow people to communicate and share content with each other
Tablet	A handheld computing device that primarily uses touch input
Touch screen	An input device used to move items on the screen and navigate

Grade 2

Term	Definition
Appropriate	Suitable use
Attribute	Physical description of an object (e.g., color, shape, size)
Author	The creator of a book, image, song, or object

Term	Definition
Block-based programming language	Environment to create a program by fitting together command blocks in a sequence
Blog	An informal website that is regularly updated by an individual or group
Computing device	An electronic device that can store and receive information
Cyberbullying	The use of electronic communication to bully a person
Data	Individual facts and information
Debug	Find and fix problems in a program
Digital artifact	An object that is made or stored on a computer
Email	Program used to communicate online
Event	Something that causes a portion of a program to run (e.g., a mouse click)
Graphic organizer	A visual display of terms, facts, and ideas
Hardware	Physical components of a computing system
Illustrator	Creator of a visual artifact (e.g., image or painting)
Inference	A conclusion reached on the basis of evidence and reasoning
Input	Data that is taken in by a computer for processing
Internet	A global computer network that allows people to communicate, create, and share content
Loop	A set of actions repeated until a condition is met

Term	Definition
Model	Creating a representation of an idea, object, or a process
Output	Data that is produced by a computer as a result of a program
Password	A secret word or phrase that must be used to gain admission to something
Pattern matching	Finding similarities between things
Prediction	Making a guess of what will happen based on current facts
Program	An algorithm that has been coded into something that can be run by a machine
Reboot	Turn off a computer and then turn it on again
Repeat	To perform an action or set of actions multiple times in a row
Sequence	An ordered set of instructions
Social media	Applications that allow people to communicate and share content with each other
Software	The programs used by a computing system
Storyboard	A sequence of drawings that represent the order of a program happening
Troubleshoot	Identify and correct faults in a computing system
Video conferencing	Communicating with someone on the Internet with both audio and video
Website	A location on the Internet referenced by a WWW address

Grade 3

Term	Definition
Algorithm	Sequence of steps that completes a task
Attribute	Physical description of an object (e.g., color, shape, size)
Author	The creator of a book, image, song, or object
Blog	An informal website that is regularly updated by an individual or group
Bug	An error or flaw in a program that causes it to give the wrong answer or crash
Composer	The creator of an audio artifact (e.g., song)
Computing device	An electronic device that can store and receive information
Computing system	Interconnected computers or computer components that work to accomplish large tasks
Cyberbullying	The use of electronic communication to bully a person
Data	Individual facts and information
Debug	Find and fix problems in a program
Decompose	Breaking a complex problem into parts that are easier to understand and solve
Email	Program used to communicate online
Event	Something that causes a portion of a program to run (e.g., a mouse click)
Graphic organizer	A visual display of facts, terms, and ideas
Illustrator	Creator of a visual artifact (e.g., image or painting)

Term	Definition
Input	Data that is taken in by a computer for processing
Internet	A global computer network that allows people to communicate, create, and share content
Loop	A set of actions repeated until a condition is met
Model	Creating a representation of an idea, object, or a process
Network	A group of computers that can communicate directly with each other
Output	Data that is produced by a computer as a result of a program
Pair Programming	
Password	A secret word or phrase that must be used to gain admission to something
Planning tool	A document or other resource to help organize thoughts in the creation of a product
Prediction	Making a guess of what will happen based on current facts
Program	An algorithm that has been coded into something that can be run by a machine
Reboot	Turn off a computer and then turn it on again
Repeat	To perform an action or set of actions multiple times in a row
Storyboard	A sequence of drawings that represent the order of a program happening
Subset	A group within a group
Text message	Electronic communication usually sent between mobile phones
Troubleshoot	Identify and correct faults in a computing system

Term	Definition
Video conferencing	Communicating with someone on the Internet with both audio and video

Grade 4

Term	Definition
Algorithm	Sequence of steps that completes a task
ASCII	A conversion chart for representing different characters in numeric form
Attribute	Physical description of an object (e.g., color, shape, size)
Author	The creator of a book, image, song, or object
Binary	The number system used by computers to represent all messages and commands
Blog	An informal website that is regularly updated by an individual or group
Bug	An error or flaw in a program that causes it to give the wrong answer or crash
Composer	The creator of an audio artifact (e.g., song)
Cyberbullying	The use of electronic communication to bully a person
Data	Individual facts and information
Debug	Find and fix problems in a program
Decompose	Breaking a complex problem into parts that are easier to understand and solve
Email	Program used to communicate online

Term	Definition
Graphic organizer	A visual display of facts, terms, and ideas
Illustrator	Creator of a visual artifact (e.g., image or painting)
Input	Data that is taken in by a computer for processing
Internet	A global computer network that allows people to communicate, create, and share content
Loop	A set of actions repeated until a condition is met
Model	Creating a representation of an idea, object, or a process
Network	A group of computers that can communicate directly with each other
Output	Data that is produced by a computer as a result of a program
Password	A secret word or phrase that must be used to gain admission to something
Pixel	Small colored dots that make up an image
Planning tool	A document or other resource to help organize thoughts in the creation of a product
Prediction	Making a guess of what will happen based on current facts
Processor	Computing component that performs the manipulation to change input into output
Reboot	Turn off a computer and then turn it on again
Sensor	Computing component that collects data that would otherwise be difficult to collect by hand
Storyboard	A sequence of drawings that represent the order of a program happening
Subset	A group within a group

Term	Definition
Text message	Electronic communication usually sent between mobile phones
Troubleshoot	Identify and correct faults in a computing system
Variable	Programming element that can hold a value
Video conferencing	Communicating with someone on the Internet with both audio and video

Grade 5

Term	Definition
Algorithm	Sequence of steps that completes a task
ASCII	A conversion chart for representing different characters in numeric form
Author	The creator of a book, image, song, or object
Binary	The number system used by computers to represent all messages and commands
Bug	An error or flaw in a program that causes it to give the wrong answer or crash
Composer	The creator of an audio artifact (e.g., song)
Conditional	A set of actions that only runs if a condition is met
Cyberbullying	The use of electronic communication to bully a person
Data	Individual facts and information
Debug	Find and fix problems in a program

Term	Definition
Decompose	Breaking a complex problem into parts that are easier to understand and solve
Graphic organizer	A visual display of facts, terms, and ideas
Illustrator	Creator of a visual artifact (e.g., image or painting)
Input	Data that is taken in by a computer for processing
Internet	A global computer network that allows people to communicate, create, and share content
Local network	A group of computers that are physically located near each other and can communicate directly
Loop	A set of actions repeated until a condition is met
Model	Creating a representation of an idea, object, or a process
Network	A group of computers that can communicate directly with each other
Output	Data that is produced by a computer as a result of a program
Password	A secret word or phrase that must be used to gain admission to something
Pixel	Small colored dots that make up an image
Planning tool	A document or other resource to help organize thoughts in the creation of a product
Prediction	Making a guess of what will happen based on current facts
Processor	Computing component that performs the manipulation to change input into output
Sensor	Computing component that collects data that would otherwise be difficult to collect by hand
Storage	Computing component that can hold data to be used at a later time

Term	Definition
Storyboard	A sequence of drawings that represent the order of a program happening
Troubleshoot	Identify and correct faults in a computing system
Variable	Programming element that can hold a value
Worldwide network	A group of computers that are spread far apart and can communicate with each other

Grade 6

Term	Definition
Binary	The number system used by computers to represent all messages and commands
Biometrics	The use of physical human characteristics to facilitate security authentication
Bug	An error or flaw in a program that causes it to give the wrong answer or crash
Cite	To mention the author of a piece of text or media
Control structure	A block of programming that analyses variables and chooses a direction in which to go based on given parameters
Creative Commons	A set of intellectual property laws that defines how a piece of text or media can be used by others.
Data	Facts and statistics collected together for reference or analysis.
Data cleaning	Detecting and correcting corrupt and inaccurate records from a set of data
Debug	Find and fix problems in a program

Term	Definition
End-user	The person who uses a particular program
Firewall	A part of a computer system or network which is designed to block unauthorized access while permitting outward communication
If statement	A programming conditional statement that, if proved true, performs a function or displays information
Incomplete data	A data set that has values or sections missing that could make analysis difficult
Loop	A control structure that repeats a section of code until a condition is met
Model	Creating a representation of an idea, object, or a process
Password	A secret word or phrase that must be used to gain admission to something
Plagiarism	The act of passing off someone else's work or ideas as one's own
Program tracing	Reading a program and predicting how variables change and what will be output
Sensors	Hardware components that collect environmental data such as temperature or sound
Simulation	The use of a model to replicate or imitate a situation or phenomenon
Source	The original text or media from which content is used and cited

Grade 7

Term	Definition
Automation	The use of largely automatic equipment in a system of manufacturing or other production process

Term	Definition
Career Pathway	A series of connected education programs and courses that prepare students for a particular career
Command	An instruction telling a computer program to do something
Compound conditionals	The use of multiple conditions joined by AND or OR in a control structure
Computational thinking	A set of problem-solving methods that express problems and their solutions in a way that a computer could execute
Computer network	A series of interconnected computers and devices that share resources and exchange data with each other
Constraints	A limit or restriction on a program or situation
Culture	The customs, arts, social institutions, and achievements of a particular nation, people, or other social group
Cybersecurity	The study and practice of protecting computers and programs from unwanted access and theft of data
Data	Facts and statistics collected together for reference or analysis
Data efficiency	The speed and resource usage involved in collecting, manipulating, and analyzing data
Data integrity	The preservation of accuracy and consistency of a data set over the course of its analysis
Data transmission	The process of transferring data between two or more digital devices

Term	Definition
Debugging	Systematically finding the cause of an error in a program and fixing it
Decompose	System of program design that involves breaking problem down into smaller tasks
Documentation	Written text or illustrations that describe a program to its users or other programmers
Firewall	A part of a computer system or network which is designed to block unauthorized access while permitting outward communication
Funnel marketing	The process of directing consumers from becoming aware of a product to the eventual sale
Global economy	The international exchange of goods and services
Goods and services	The products and actions exchanged for money in an economy
Identity Theft	The deliberate use of someone's personal data for financial gain or to harm their reputation
If statements	A programming conditional statement that, if proved true, performs a function or displays information
Input	Data that is fed into a program to be processed
Internet	A global computer network consisting of multiple interconnected networks
Internet speed	The rate at which data is communicated from the Internet to a computer and vice versa

Term	Definition
IP address	A numerical label assigned to each computing device on a network
Model	Creating a representation of an idea, object, or a process
Network	A group of interconnected computers and other devices
Occupational Outlook Handbook	A publication of the US Department of Labor's statistics about various aspects of work in the United States
Output	Data that is produced by a program for the user or by another program
Parameter	A special value that is used to further define the action of a function
Password	A secret word or phrase that must be used to gain admission to something
Pay rate	The amount of money workers are paid by hour, week, etc.
Protocol	The official procedure of a particular computing process
Reliability	The ability of computers and the Internet to consistently perform to the expectations of their designed function
Salary	The amount of money works are paid as described by an employment contract
Simulation	The use of a model to replicate or imitate a situation or phenomenon

Term	Definition
Subprogram	A section of code designed to complete a task that is used multiple times within a program
Technology	Computing devices and skills used in production or investigation
Variable	Programming element that can hold a numeric or non-numeric value
Workplace readiness skill	The basic academic, critical thinking and personal skills necessary to maintain employment

Grade 8

Term	Definition
Command	An instruction telling a computer program to do something
Cybersecurity	The study and practice of protecting computers and programs from unwanted access and theft of data
Debugging	Systematically finding the cause of an error in a program and fixing it
Denial-of-Service attack	A cyber-attack where a machine is flooded with simple requests making it unable to respond to more meaningful requests
Digital footprint	The collection of data that is associated with your actions and communication on the Internet

Term	Definition
Digital forensics	The investigation and recovery of material found in digital devices
Encryption	Encoding a message so that only the intended parties can read it
Flowchart	A type of diagram that represents the path and logic through a program
Hacker	A person who uses computers to gain unauthorized access to data.
Hacking	Gaining access to a website, program, or other resource you are not supposed to
Identity theft	The deliberate use of someone's personal data for financial gain or to harm their reputation
If-else statement	A programming conditional statement that defines two actions to be run depending on the result of a particular condition
Internet	A global computer network consisting of multiple interconnected networks
IP address	A numerical label assigned to each computing device on a network
Loop	A control structure that repeats a section of code until a condition is met
Malware	Software that is intentionally designed to cause damage to a computer, server, or network
Nesting	Organizing control structures such as if statements or loops within other control structures

Term	Definition
Phishing	The fraudulent practice of sending emails purporting to be from reputable companies in order to induce individuals to reveal personal information, such as passwords and credit card numbers.
Privacy	Protecting data or actions performed on a computing system
Private sector	The part of the economy that is not under government control
Public domain	All the creative works which have no intellectual property rights applied
Trojan horse	A form of malware that appears to be harmless software but performs more malicious actions
URL	Uniform Resource Locator, or more commonly known as a web address
Virus	A piece of malware that is capable of replicating itself and embedding in various aspects of a computer