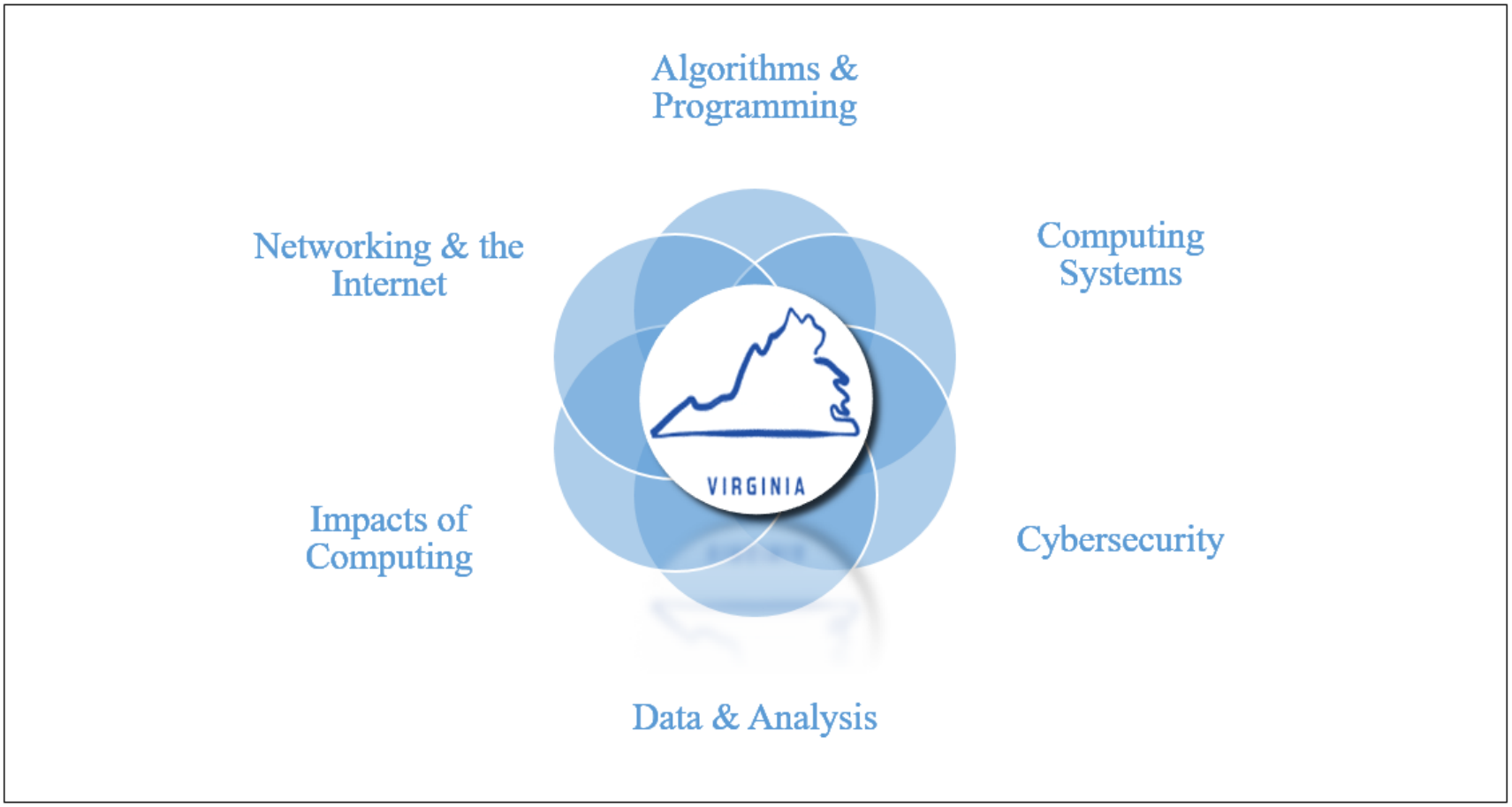
# Computer Science Standards of Learning

Curriculum Framework




Board of Education
Commonwealth of Virginia

**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).

# 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 should use these tools for exploring and creating computer programs, facilitating reasoning and problem solving, and verifying solutions.

**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.<br><br>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 *demonstrate* these skills:<br><br>• Define the components in a typical computing system.<br>• Design a questionnaire for a client to ascertain what type of system they would need.<br>• Illustrate and diagram the inside of a computing system (laptop or desktop).<br>• Design a computer system created for a specific need or client. (i.e. gaming computer, college student). | Students should *investigate* these concepts:<br><br>• How may quantum computing change the typical computer systems we currently use?<br>• 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 *apply* these terms in context:<br><br>• Central Processing Unit (CPU)<br>• Components<br>• Computer Systems<br>• Graphical Processing Unit (GPU)<br>• Hard Drive<br>• Hardware<br>• Motherboard<br>• Random Access Memory (RAM)<br>• 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.<br><br>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.<br><br>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 can then be 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 *demonstrate* these skills:<br><br>• Diagram a program running a simple mathematical operation using the IPOS model.<br>• Describe the roles of individual components in the processing system. | Students should *investigate* these concepts:<br><br>• How do different types of storage enable the processing system to perform complex tasks?<br>• What types of computing systems might not have traditional input and output mechanisms like a keyboard or screen? | Students should *apply* these terms in context:<br><br>• Component<br>• Computing System<br>• Input<br>• Logic<br>• Output<br>• Processing<br>• 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 process 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.<br><br>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 *demonstrate* these skills:<br><br>• Explain how computers send messages to one another over the Internet.<br>• Describe how the Internet is a network of networks built with many layers of Internet technology. | Students should *investigate* these concepts:<br><br>• How does the initiating computer locate and display webpages from a URL request?<br>• What happens if part of the physical network is broken due to a cut line or other damage? | Students should *apply* these terms in context:<br><br>• DNS<br>• IP<br>• Protocol<br>• Network<br>• Router |

| | | |
|---|---|---|
| | • How does the physical Internet span long distances, like across oceans? | • Server<br>• 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.<br><br>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.<br><br>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 *demonstrate* these skills:<br><br>• Describe the underlying concepts that support an ever-growing system of networks called the Internet.<br>• Explain how the Internet is designed to be reliable for fast communication across the globe.<br>• Define the purpose and make-up of the IETF. | Students should *investigate* these concepts:<br><br>• What would be the advantages and disadvantages of a specific country or entity determining the protocols and architecture of the Internet?<br>• How might the experience of the Internet be different if every country had their own way of communicating over the Internet? | Students should *apply* these terms in context:<br><br>• IETF<br>• IP Address<br>• Protocol<br>• Redundancy<br>• Reliability<br>• 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.<br><br>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.<br><br>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 *demonstrate* these skills:<br><br>• Compare and contrast the characteristics of symmetrical and asymmetrical encryption methods.<br>• Explain the importance of encryption for messages sent using the Internet. | Students should *investigate* these concepts:<br><br>• Why do new encryption methods need to be increasingly more complex?<br>• What personal identifying information could someone find about you based on your use of the Internet? | Students should *apply* these terms in context:<br><br>• Asymmetrical Encryption<br>• Encryption<br>• Personal Identifying Information<br>• Private Key<br>• Public Key<br>• 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 |

| **Context of the Standard** |
|---|
| 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.<br><br>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 *demonstrate* these skills:<br><br>• Explore how large data sets are collected and analyzed.<br>• Explain the tradeoffs experienced by users, programmers, companies, and communities when companies collect large amounts of data. | Students should *investigate* these concepts:<br><br>• Who is responsible for ensuring that companies use data responsibly?<br>• Are there kinds of information companies should not be able to gather from users?<br>• Who benefits from advances in machine learning? Who is harmed? | Students should *apply* these terms in context:<br><br>• Algorithm<br>• Cookie<br>• 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 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.<br><br>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 application programming interface (API), 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 *demonstrate* these skills:<br><br>• Create an information hypothesis from a set of data that they have created/generated.<br>• Design a relational database that is connected to show how input data from an app might be stored. | Students should *investigate* these concepts:<br><br>• What are some ways software programs create data?<br>• Who owns data generated by users as they interact with a software program? | Students should *apply* these terms in context:<br><br>• Relational database<br>• SQL<br>• 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 application programming interface (API), a series of specialized commands which allow programs to access a database.<br><br>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 *demonstrate* these skills:<br><br>• Discuss how data representations can aid in the interpretation of collected data.<br>• Examine the tradeoffs created by distilling information through a | Students should *investigate* these concepts:<br><br>• How do websites use various data representation methods to promote their bias?<br>• Are there methods that can be used to enable unbiased interpretation of data? | Students should *apply* these terms in context:<br><br>• API<br>• Data Representation<br>• Raw Data<br>• 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

a. compound conditional execution; and
b. 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 might involve using nested loops to iterate through a multi-dimensional table. |

| Essential Skills | Essential Questions | Essential Vocabulary |
|---|---|---|
| Students should *demonstrate* these skills:<br>• Create a program that makes use of compound conditional execution and complex iteration. | Students should *investigate* these concepts:<br>• What is an example of a decision-making process that might be expressed using compound conditional statements | Students should *apply* these terms in context:<br>• Conditional<br>• Boolean<br>• Iteration |

| Essential Skills | Essential Questions | Essential Vocabulary |
|---|---|---|
| • Predict the result of complex Boolean and/or logical expressions/conditions.<br>• Trace the values of named variables over the course of a program that uses nested loops. | (e.g., college entrance criteria, athletic team tryouts)?<br>• Why do programmers use loops and iteration rather than writing out long lists of repetitive code? | • Input<br>• 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.<br><br>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 |
|---|---|---|
| Students should *demonstrate* these skills:<br><br>• Create a program that uses existing functions or libraries in a given programming language/framework.<br>• Research, download, install, and use a third-party open-source library or framework.<br>• Create a program that makes use of defined functions. | Students should *investigate* these concepts:<br><br>• Why do programmers use libraries?<br>• 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 *apply* these terms in context:<br><br>• Function<br>• Library<br>• Parameter<br>• 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.<br><br>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 *demonstrate* these skills:<br><br>• Create programs that use data structures.<br>• Create a program that uses an iterator.<br>• Generate a program that creates and processes data structures given pseudocode examples. | Students should *investigate* these concepts:<br><br>• Why do programmers store data in lists or tables instead of individual variables?<br>• Why do programmers use iterators rather than writing everything out?<br>• What are some challenges that arise when debugging iterators? | Students should *apply* these terms in context:<br><br>• Data<br>• Iterator<br>• 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 |
|---|---|---|
| Students should *demonstrate* these skills:<br><br>• Debug a given software program.<br>• Create a software program that includes mechanisms to assist in debugging.<br>• Generate procedures for testing and debugging software. | Students should *investigate* these concepts:<br><br>• What are some ways to help avoid a situation where you don't know how to debug your software?<br>• What are some ways to identify and validate the assumptions you make about how your program works as you work on debugging? | Students should *apply* these terms in context:<br><br>• Code tracing<br>• Debug<br>• 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.<br><br>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.<br><br>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 *demonstrate* these skills:<br><br>• Explain how computing innovations impact our global society in both negative and positive ways.<br>• Explore the ways in which collaboration is supported by new technologies. | Students should *investigate* these concepts:<br><br>• 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?<br>• What effects does collaboration have on the development of computing products? | Students should *apply* these terms in context:<br><br>• Collaboration<br>• Crowdsourcing<br>• Data<br>• Digital Economy<br>• 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.<br><br>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.<br><br>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 *demonstrate* these skills:<br><br>• Examine the global distribution of computing resources and evaluate its | Students should *investigate* these concepts:<br><br>• Who made the devices you use every day? What is their job like? | Students should *apply* these terms in context:<br><br>• Cloud Computing<br>• Data |

| Essential Skills | Essential Questions | Essential Vocabulary |
|---|---|---|
| impact on equity, access, and influence on individual locations.<br>• Explore the tradeoffs of cloud computing for collaboration in the context of access and equity. | • Who made the software you use every day? What is their job like? | • Digital Divide<br>• Manufacturing<br>• Software Development<br>• 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 *demonstrate* these skills:<br><br>• Explain copyright law as it applies to software.<br>• Define "open source."<br>• Identify different methods of protecting intellectual property as they relate to computing (e.g., license, copyright, patent). | Students should *investigate* these concepts:<br><br>• Why do programmers create open-source software?<br>• What are some of the reasons a programmer might decide to make software "closed-source?"<br>• Is it ethical for programmers to create open-source alternatives to paid software (e.g., Inkscape and Corel)? | Students should *apply* these terms in context:<br><br>• Intellectual Property<br>• Open Source<br>• License<br>• Patent<br>• 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.<br><br>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.<br><br>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 *demonstrate* these skills:<br><br>• Discuss how data is used in economic and social contexts.<br>• Identify impacts of computing on political, social, and economic contexts.<br>• Discuss how the data economy impacts individuals. | Students should *investigate* these concepts:<br><br>• Is privacy a right?<br>• Who owns user data generated through interaction with software applications?<br>• Who is responsible for people who misuse programs and exploit users? | Students should *apply* these terms in context:<br><br>• Data Economy<br>• Machine Learning Algorithm<br>• Algorithmic Bias |