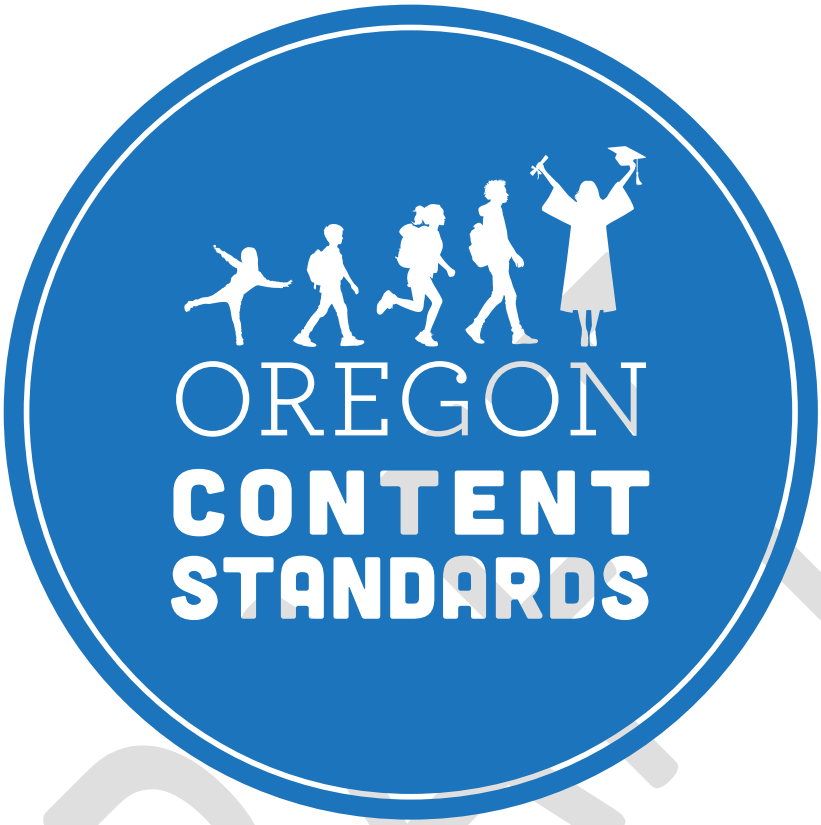

Oregon's Optional Computer Science Standards

Draft for Public Standards Review

VERSION 1.0

Table of Contents

Introduction

Oregon's Optional Computer Science Domains, Standards and Key Concepts

Background

The computer science standards have been developed as a result of the 2023 Oregon Science Education Statewide Implementation Plan. The implementation plan identifies the following goals:

- Computer science education is made available to public school students on an equitable basis.
- Computer science education is based on a framework that guides students from computer users to computer literate creators who are proficient in the concepts and practices of computer science, as informed by national frameworks and standards.

Specifically, these standards respond to Strategy 4.2 in the plan which calls for K-12 standards that courses.

Implementation of Standards

To those ends, the standards are based on the work of the Computer Science Teachers Association (CSTA) and incorporate feedback from Oregon educators and other national partners. These Oregon standards are optional and provide guidance to K-12 educators who wish to identify important ideas in computer science taught at any grade level. Each teacher, school, or district decides how to apply these standards to enhance computer science education using a model that works best in their specific setting.

The Oregon Computer Science Standards are divided into grade bands to recognize the intellectual development of students as well as the implementation capacity of each school or classroom. The number of standards at each grade band reflects the following assumptions:

- Elementary classrooms are heavily focused on building foundational content skills that are taught by teachers whose training may not have included computer science. The standards are chosen so they can be taught in classrooms that may have limited technological resources. They have also been chosen for their possible connection to other content such as math and science.
- Middle schools are more likely to have technology specialists who are teaching technology classes. This opens more time to focus on computer science as a discipline. With a potential concentration of technology within specialty classes, students have more opportunity to use that technology to explore computer science.
- Many high schools have included one or more computer science classes or class sections in the schedule. These are frequently taught by Career and Technical Education (CTE) teachers who have significant work and educational experience related to computer science. The standards were chosen to identify the content of a high school foundational computer science class.
- Some high schools offer classes in computer science that address more specific topics such as cybersecurity, artificial intelligence robotics, or game design. Many of those courses are part of CTE Programs of Study. The standards for more advanced or specialized courses are not included in this document. Educators seeking guidance on content in advanced courses may wish to refer to the Oregon Statewide Program of Study Framework: Information and Communication Technology Career Cluster or the national CSTA Reimagining CS Computer Sciences Standards.

The inclusion of key concepts provides an additional resource for teachers. These are not additional standards or instruction. The key concepts are intended to help teachers clarify the scope and language of a standard. They also can provide suggestions for lesson objectives or instructional content.

Integrating Computer Science

Computer science standards often intersect with other disciplines particularly in the domains of Algorithms and Data and Analysis. Intentional connections between science and math content with computer science can occur with very little change in lessons even from early grades. Math, science, and computer science standards include standards related to developing questions that can be answered with data as well as manipulating and displaying that data. The general concept of an algorithm appears in science and math as a mechanism to communicate investigations or strategies for solving problems. The ODE STEAM Toolkit provides significant information and resources related to integrating content across all disciplines. CSTA provides some additional examples of Integrating CS into Other Subject Areas.

Domains and Progression

All standards are grouped within 5 domains that mirror those used by CSTA.

1. **Algorithms (ALGM)** - Algorithms are step-by-step processes to complete a task or solve a problem. They are a fundamental part of computer science and understanding them is foundational for further work in computing. In this domain, students are exposed to high-level concepts related to algorithms. The algorithm standards in early grades simple everyday tasks and progress to more complex and abstract tasks.
2. **Programming (PRG)** - Programming is construed broadly to describe a variety of ways of generating computational artifacts. Programming, in the context of essential content for high school, is likely to include block-based and/or text-based programming languages. It may also include other computational artifacts, such as simulations, visualizations, robotic systems, or digital animations. Programming starts with standards that can be taught without computing technology to knowledge and skills that would require more access to the technology. There is a strong connection between algorithms and programming throughout.
3. **Data and Analysis (DAA)** - Data and analysis involve understanding how computing systems collect, store, and process data and how people can use this data to make inferences and predictions. The increasing importance of data science and artificial intelligence points to the increasing need for understanding the basic elements of data and its analysis. The standards progress from general concepts related to data that are also fundamental in math and science instruction to applications of data concepts in programming environments.
4. **Computing Systems and Security (CSS)** - Computing systems and security include the broad categories of hardware, software, troubleshooting, networks, and cybersecurity, as well as the idea that systems have multiple levels or layers that impact each other. The increased interconnectedness of large systems and their impact on safety and security underscore the importance of this domain. Standards related to security are introduced early since students are exposed to those ideas even before attending school. The inclusion of other aspects of computing systems reflects the broader student exposure in school.

5. **Computing and Society (CAS)** – Computing and society brings together two threads: 1) the student's own future, specifically pathways and careers that involve computing in some respect and 2) emerging technologies, including their societal implications and ethical issues. These standards change with a changing student experience with early grades focused on examples that are closer to home and later grades focused on possible futures.

Standards

Grade K-2 Computer Science Standards

Algorithms (K-2.ALGM)

CS.K-2. ALGM.01 Model daily processes by creating and following algorithms that include **sequencing (proceeding left-to-right and top-to-bottom), events, and iteration** to complete tasks.

CS.K-2. ALGM.02 Modify algorithms with repeating patterns to use iteration instead of repeated instructions.

CS.K-2. ALGM.03 Compare how different algorithms for solving the same problem may affect people differently.

Programming (K-2.PRG)

CS.K-2.PRG.01 Create code from an algorithm that includes sequence, events, and iteration to express ideas or complete a task.

Data and Analysis (K-2.DAA)

CS.K-2.DAA.01 Collect numeric and non-numeric data, using multiple methods, including observation, measurement, and survey.

CS.K-2.DAA.02 Investigate questions that can be answered by manually collecting data in students' everyday environments.

CS.K-2.DAA.03 Investigate a variety of data questions that address the needs of a person or community.

CS.K-2.DAA.04 Recognize that data may represent some people or perspectives more than others and discuss how missing or incomplete information can affect conclusions.

Computing Systems and Security (K-2.CSS)

CS.K-2.CSS.01 Examine the use of tools to accomplish tasks and/or solve problems.

CS.K-2.CSS.02 Describe the purposes of basic hardware components in a computing system, using accurate terminology.

CS.K-2.CSS.03 Apply practices for keeping personal data secure.

CS.K-2.CSS.04 Evaluate how sharing information online might reveal personally identifiable information (PII) and other details to people other than the intended recipients.

CS.K-2.CSS.05 Evaluate an individual's role in responsibly using computing tools.

CS.K-2.CSS.06 Create a simple, strong password using a combination of letters, numbers, and symbols (with teacher guidance).

CS.K-2.CSS.07 Explain why passwords should be kept private (except with a trusted adult).

Computing and Society (K-2.CAS)

CS.K-2.CAS.01 Compare daily life before and after the implementation or adoption of computing technology.

CS.K-2.CAS.02 Describe how computing technology, used in daily life, at home, and at school, can help people.

CS.K-2.CAS.03 Identify how people use digital devices in their daily work.

Grade 3-5 Computer Science Standards

Algorithms (3-5.ALGM)

CS.3-5. ALGM.01 Create visual representations for algorithms that include sequence, events, iteration, and selection to solve a problem or complete a task.

CS.3-5. ALGM.02 Modify and model algorithms with repeating patterns to use iteration instead of repeated instructions such as spiral requirements**.**

CS.3-5. ALGM.03 Decompose a problem or task into smaller components to develop an algorithm.

CS.3-5. ALGM.04 Assess whether an algorithm meets specific conditions and does what the user expects, using defined requirements and feedback to evaluate and improve the solution.

CS.3-5. ALGM.05 Examine different perspectives, abilities, and points of view when designing algorithms and programs.

Programming (3-5.PRG)

CS.3-5.PRG.01 Develop code from a student-created algorithm that includes sequence, events, iteration, and selection to express ideas or complete a task.

CS.3-5.PRG.02 Debug errors in a program that includes sequence, events, iteration, and selection.

CS.3-5.PRG.03 Create a unique program by modifying other programs or incorporating portions of other programs into one's own work to develop something new or add more advanced features.

CS.3-5.PRG.04 Collaborate with a team to create a program, ensuring that all team members have a role and contribute equally.

CS.3-5.PRG.05 Collaborate with peers to design, implement, document, and review a programming project that expresses an idea or solves a problem.

Data and Analysis (3-5.DAA)

CS.3-5.DAA.01 Organize data into tables where rows represent a "record" and columns represent attributes.

CS.3-5.DAA.02 Create a data visualization and a brief narrative to report the process and results of a data investigation.

CS.3-5.DAA.03 Design a data collection approach that addresses the needs of people from different backgrounds or groups.

CS.3-5.DAA.04 Explain design choices, identify limitations in the data, describe who produced the data and why, and distinguish between observed patterns and claims about cause and effect.

CS.3-5.DAA.05 Consider whose perspectives may be missing, how question wording or measurement methods may introduce bias, and how personal assumptions or confirmation bias may influence interpretation.

Computing Systems and Security (3-5.CSS)

CS.3-5.CSS.01 Articulate steps to implement common solutions for hardware and software issues using accurate terminology.

CS.3-5.CSS.02 Explain how computing devices connect to the Internet using wires or wireless signals.

CS.3-5.CSS.03 Illustrate how people access the global network of the Internet to access information and communicate with each other.

CS.3-5.CSS.04 Evaluate how authentication, different levels of access, and security measures help protect information on personal and public devices and networks.

CS.3-5.CSS.05 Justify the importance of monitoring and updating security measures to prevent unauthorized access to information and other harms.

CS.3-5.CSS.06 Describe the benefits and harms of widely used computing platforms to an individual's life and human connections.

CS.3-5.CSS.07 Recognize examples of personal information (name, address, school, birthday) that should not be shared online without adult permission.

CS.3-5.CSS.08 Describe basic password management strategies (e.g., using a password manager with adult guidance, not sharing passwords with friends).

Computing and Society (3-5.CAS)

CS.3-5.CAS.01 Examine how a computing innovation changed the way people lived, worked, or communicated over time.

CS.3-5.CAS.02 Evaluate how people make choices about the use of emerging technologies, based on their needs and the consequences.

CS.3-5.CAS.03 Explain how computing technologies and skills are used across different industries.

Grade 6-8 Computer Science Standards

Algorithms (6-8.ALGM)

CS.6-8. ALGM.01 Create visual or textual representations of algorithms that include sequence, events, iteration, selection, and variables to solve a problem or complete a task.

CS.6-8. ALGM.02 Optimize visual representations of algorithms.

CS.6-8. ALGM.03 Refine algorithms iteratively through user feedback to improve usability, accessibility, and user experience.

CS.6-8. ALGM.04 Demonstrate how an algorithm meets defined conditions and user expectations for given inputs.

CS.6-8. ALGM.05 Describe common societal impacts, ethical issues, and biases of deterministic and probabilistic algorithms.

Programming (6-8.PRG)

CS.6-8.PRG.01 Develop code from student-created algorithms that include sequence, events, iteration, selection, and variables to express ideas, complete a task, or solve a problem.

CS.6-8.PRG.02 Analyze how a segment of code works by identifying and describing the roles of key components.

CS.6-8.PRG.03 Use appropriate data types to store, update, and evaluate data within a program.

CS.6-8.PRG.04 Modify existing programs that incorporate sequence, selection, and iteration.

CS.6-8.PRG.05 Utilize reference documentation, online resources, and programming tools to assist in writing, debugging, and improving code.

CS.6-8.PRG.06 Use standard practices to test, debug, document, and peer-review code.

CS.6-8.PRG.07 Modify a program to improve usability and accessibility.

CS.6-8.PRG.08 Apply inclusive collaboration practices to support all stages of programming, from planning to testing.

CS.6-8.PRG.09 Document a program, using comments, descriptive names, and structured guides, to improve readability, enable collaboration, and explain complex logic or code intent.

Data and Analysis (6-8.DAA)

CS.6-8.DAA.01 Collect different types of data, using computational tools.

CS.6-8.DAA.02 Evaluate the impact of precision and granularity in data collection and analysis, considering how different levels of detail affect accuracy, storage, and interpretation.

CS.6-8.DAA.03 Use computational tools such as spreadsheets to collect and organize quantitative and qualitative data.

CS.6-8.DAA.04 Use computational tools to organize, filter, group, and aggregate data.

CS.6-8.DAA.05 Use computational tools to manipulate data.

CS.6-8.DAA.06 Identify errors in data.

CS.6-8.DAA.07 Use computational tools to identify relationships among variables in a dataset and make classifications or predictions.

CS.6-8.DAA.08 Create data visualizations to demonstrate how different design choices can affect a data visualization's clarity, visual appeal, accessibility, and capacity to accurately communicate insights from data investigations.

CS.6-8.DAA.09 Analyze how decisions made during data collection, data processing, data analysis, and data presentation can lead to biased data, misleading conclusions, and compromised AI models.

CS.6-8.DAA.10 Analyze the societal impacts of data-driven algorithms and computational systems, including AI.

Computing Systems and Security (6-8.CSS)

CS.6-8.CSS.01 Describe the structure and organization of file systems, including file naming conventions, directories, and file permissions, and their role in storing and managing data.

CS.6-8.CSS.02 Apply basic troubleshooting processes to identify and fix common hardware and software issues.

CS.6-8.CSS.03 Explain the key components of the Internet, their roles and functionality, and how they contribute to its resilience.

CS.6-8.CSS.04 Model how information travels securely across digital networks through the use of physical and software tools.

CS.6-8.CSS.05 Classify common types of cyberattacks, including social engineering and malware.

CS.6-8.CSS.06 Examine differences in access to computing systems, based on personal and social factors, including physical ability, geographic location, socioeconomic status, and age.

CS.6-8.CSS.07 Analyze how the elements of governing protocols, and distribution methods enable scalability, compatibility, and resilience surrounding the internet in a changing technological landscape.

Computing and Society (6-8.CAS)

CS.6-8.CAS.01 Analyze intended and unintended impacts of historical computing technologies on society and the environment.

CS.6-8.CAS.02 Evaluate how design decisions in emerging technologies influence user experiences differently across different communities.

CS.6-8.CAS.03 Contrast the features, functionality, and characteristics of emerging technologies with technologies that came before.

CS.6-8.CAS.04 Examine how changes in technology can create new jobs or change how people work.

CS.6-8.CAS.05 Illustrate how technology- driven challenges require integrating computing knowledge and skills with expertise from diverse fields.

High School Foundational Computer Science Standards

Algorithms (HSF.ALGM)

CS.HSF.ALGM.01 Develop algorithms that include variables, data, and storage, using authentic, real-world data.

CS.HSF.ALGM.02 Describe and analyze differences among deterministic algorithms and probabilistic algorithms accounting for increasing complexity.

Programming (HSF.PRG)

CS.HSF.PRG.01 Convert an algorithm written in pseudocode into a program that uses sequence, selection, iteration, procedures with parameters, and lists.

CS.HSF.PRG.02 Analyze the purpose of a segment of code.

CS.HSF.PRG.03 Assess opportunities to incorporate external code, utilizing documentation, libraries, APIs, development tools, and online resources.

CS.HSF.PRG.04 Modify a program to improve or change functionality, usability, accessibility, safety, accuracy, or inclusivity of a program.

CS.HSF.PRG.05 Argue how a program does and does not address a given problem.

CS.HSF.PRG.06 Apply correct attribution to intellectual property.

Data and Analysis (HSF.DAA)

CS.HSF.DAA.01 Describe the differences among nominal, ordinal, discrete, and continuous data and how each type of data might be generated and used in data analysis.

CS.HSF.DAA.02 Design a process that uses computational tools to collect data to answer a question, make classifications, or make predictions.

CS.HSF.DAA.03 Analyze the consequences of using data in AI/ML applications, including how biased training data can lead to biased output and reinforce societal inequalities and injustices with misinformation and disinformation.

CS.HSF.DAA.04 Evaluate the societal, environmental, and ethical implications of large- scale data collection and usage.

Computing Systems and Security (HSF.CSS)

CS.HSF.CSS.01 Apply systematic troubleshooting techniques to identify, diagnose, and resolve issues in computing systems.

CS.HSF.CSS.02 Debate the trade-offs of global access to computing systems for society in terms of societal norms, interactions, and digital engagement.

CS.HSF.CSS.03 Evaluate the rationales behind laws, policies, and best practices governing the design and use of computing systems.

CS.HSF.CSS.04 Analyze the costs of cybersecurity breaches and social engineering attacks for individuals, industries, and governments.

CS.HSF.CSS.05 Investigate the societal and environmental impacts of computing systems and the physical infrastructure that supports them.

CS.HSF.CSS.06 Distinguish among the different types of cyberattacks that affect information security for individuals and organizations.

CS.HSF.CSS.07 Diagram computing systems that integrate security protocols, incorporating user-centered design principles such as user research, prototyping, and iterative design.

Computing and Society (HSF.CAS)

CS.HSF.CAS.01 Differentiate major eras in computing history and key advancements by notable individuals and organizations.

CS.HSF.CAS.02 Evaluate policies and legislation designed to encourage ethical innovation and minimize societal risks associated with technology.

CS.HSF.CAS.03 Explain how the computing principles underlying emerging technologies are being used in innovative ways.

CS.HSF.CAS.04 Describe how an emerging technology will impact an existing project or technology.

CS.HSF.CAS.05 Evaluate how computing knowledge and skills align with personal interests and career aspirations.

CS.HSF.CAS.06 Investigate how professionals apply CS in their careers, drawing from their personal narratives.

Appendix A: Grade K-2 Key Concepts

Standard: CS.K-2.ALGM.01

Standards Statement

Model daily processes by creating and following algorithms that include sequencing (proceeding left-to-right and top-to-bottom), events, and iteration to complete tasks.

Key Concepts

- Students understand that an algorithm is step-by-step set of instructions to solve a problem or complete a task.
- Students understand that sequence is the order in which steps are executed (e.g. The sequence for brushing teeth is putting toothpaste on the brush → brushing → rinsing).
- Students understand that events are actions that trigger a response (e.g. When the alarm rings, start the morning routine).
- Students understand that iteration is repeating a set of instructions until a condition is met (e.g. repeat brushing the teeth in each part of the mouth for 30 seconds.

Standard: CS.K-2.ALGM.02

Standards Statement

Modify and algorithms with repeating patterns to use iteration instead of repeated instructions.

Key Concepts

- Students understand that iteration is repeating a set of instructions until a condition is met (e.g. repeat brushing the teeth in each part of the mouth for 30 seconds.
- Students can spot repeated actions in an algorithm.
- Students can generalize repeated actions (e.g. Instead of saying "step forward" five times, generalize by saying "repeat step forward 5 times").
- Students understand that iteration can make an algorithm easier to change.

Standard: CS. K-2. ALGM.03

Standards Statement

Compare how different algorithms for solving the same problem may affect people differently.

Key Concepts

- Students can discuss how algorithms might affect people differently by responding to prompts such as:
    - Can computers make mistakes like people do?
    - What would happen if a computer only gave answers that helped some people but not others?
    - How can I tell if a computer is being unfair or making a mistake?

Standard: CS.K-2.DAA.01

Standards Statement

Collect numeric and non-numeric data, using multiple methods, including observation, measurement, and survey.

Key Concepts

- Students understand that data is information collected to answer questions or solve problems.
- Students understand that numeric data is data using numbers.
- Students understand that non-numeric data is data using words, categories, or images.
- Students collect data to answer a question or solve a problem by observing, measuring, or conducting surveys.

Standard: CS.K-2.DAA.02

Standards Statement

Investigate questions that can be answered by manually collecting data in students' everyday environments.

Key Concepts

- Students can develop a question about the everyday environment that can be answered using data. Examples of questions might be:
    - How many students are wearing sneakers today?
    - What is the most common playground activity?
    - How long does it take to walk from the classroom to the library?
    - What time do most students go to bed?

Standard: CS.K-2.DAA.03

Standards Statement

Investigate a variety of data questions that address the needs of a person or community.

Key Concepts

- Students can identify personal needs (e.g. feeling safe, staying healthy, having fun, learning better, getting enough rest) or community needs (e.g. keeping the school clean, making the playground safe, helping others, saving energy or water, planning events).
- Students can develop a data question related to a personal or community need (e.g. What games do students play at recess? What time do most students go to bed?, What areas of the playground are used the most? What kind of field trip would most students enjoy?).
- Students collect numeric or non-numeric data that would help answer their questions.

Standard: CS.K-2.DAA.04

Standards Statement

Recognize that data may represent some people or perspectives more than others and discuss how missing or incomplete information can affect conclusions.

Key Concepts

- Some groups, experiences, or perspectives show up more often in datasets than others.
- This can happen because of who collected the data, how it was collected, or what questions were asked.
- As a result, certain voices may be amplified while others are minimized or missing entirely.

Standard: CS.K-2.CSS.01

Standards Statement

Examine the use of tools to accomplish tasks and/or solve problems.

Key Concepts

- Students understand that tools can be any object, device, or software to help complete a task or solve a problem (e.g. rulers, thermometers, calculators).
- Students understand that tools are used to make tasks easier, faster, or more accurate (e.g. using a ruler to measure instead of guessing).
- Students recognize that different tools are better for different jobs.

Standard: CS.K-2.CSS.02

Standards Statement

Describe the purposes of basic hardware components in a computing system, using accurate terminology.

Key Concepts

- Students know that hardware is the physical parts (e.g. keyboard, mouse, monitor) of a computer system that you can touch.
- Students understand that input devices (e.g. keyboard, mouse, microphone) send information into the computer.
- Students understand that output devices (e.g. monitor, speakers, printer) show or give information from the computer.

Standard: CS.K-2.CSS.03

Standards Statement

Apply practices for keeping personal data secure.

Key Concepts

- Students know that personal data is information that belongs to you and can identify you (e.g. name, birthday, address, school name, photos).
- Students know it is important to keep personal information safe and private
- Students understand practices that help keep personal data safe.
    - Passwords
    - Secure Websites
    - Safe Sharing

Standard: CS.K-2.CSS.04

Standards Statement

Evaluate how sharing information online might reveal personally identifiable information (PII) and other details to people other than the intended recipients.

Key Concepts

- Students know examples of Personally Identifiable Information (PII).
- Students understand that when posting something on the internet it can be seen by lots of people, not just their friends.
- Students understand that not everyone online is someone they know.
- Students understand that they should talk to a trusted adult before sharing something online.

Standard: CS.K-2.CSS.05

Standards Statement

Evaluate an individual's role in responsibly using computing tools.

Key Concepts

- Students understand their individual responsibilities that include:
    - Being a good digital citizen.
    - Choosing the right tool for the right task.
    - Respect the rights and feelings of others.
    - Keeping personal information private.
    - Following classroom, school, and family rules for using technology

Standard: CS.K-2.CSS.06

Standards Statement

Create a simple, strong password using a combination of letters, numbers, and symbols (with teacher guidance).

Key Concepts

- Use different kinds of characters**:** Make a password by mixing letters, numbers, and symbols so it's harder for others to guess.
- Keep it simple but strong**:** Create a password you can remember, but not something others can easily figure out.

- Get help from a trusted adult**:** Work with your teacher to make sure your password is safe and follows the rules.

Standard: CS.K-2.CSS.07

Standards Statement

Explain why passwords should be kept private (except with a trusted adult).

Key Concepts

- Passwords keep your information safe**:** If others know your password, they could get into your accounts or pretend to be you.

Only share with a trusted adult**:** Grown-ups like parents or teachers can help you stay safe, but other people should never know your password.

Standard: CS.K-2.CAS.01

Standards Statement

Compare daily life before and after the implementation or adoption of computing technology.

Key Concepts

- Students can identify how daily life tasks such as writing, shopping, learning, communicating, traveling and playing have changed with implementation of computing technology.

Standard: CS.K-2.CAS.02

Standards Statement

Describe how computing technology, used in daily life, at home, and at school, can help people.

Key Concepts

- Students can describe how computing technology helps at home including:
    - Smart devices can turn lights on and off or play music.
    - Tablets and phones can help families talk to each other from far away.
    - Computers can help with shopping, recipes, and watching shows.
- Students can describe how computing technology helps at school including:
    - Computers and tablets that help students learn by playing games, watching videos, and doing research.
    - How teachers use technology to show lessons, take attendance, and share homework.
    - How students can create art, stories, and projects using digital tools.

Standard: CS.K-2.CAS.03

Standards Statement

Identify how people use digital devices in their daily work.

Key Concepts

- Students recognize examples of digital devices such as computers, laptops, smartphones, tablets, wearables, and other specialized equipment.

- Students can identify how digital devices are used in daily work such as email, video conferencing, shared documents, and research.

Appendix B: Grade 3-5 Key Concepts

Standard: CS.3-5. ALGM.01

Standards Statement

Create visual representations for algorithms that include sequence, events, iteration, and selection to solve a problem or complete a task.

Key Concepts

- Students understand that an algorithm is step-by-step set of instructions to solve a problem or complete a task.
- Students understand that sequence is the order in which steps are executed (e.g. The sequence for brushing teeth is putting toothpaste on the brush → brushing → rinsing).
- Students understand that events are actions that trigger a response (e.g. When the alarm rings, start the morning routine).
- Students understand that iteration is repeating a set of instructions until a condition is met (e.g. repeat brushing the teeth in each part of the mouth for 30 seconds).
- Students understand that selection is making decisions based on conditions (e.g. "If it's raining, take an umbrella; else, wear sunglasses").
- Students create visual representations of an algorithm that may include storyboards, flowcharts, or decision trees.

Standard: CS.3-5. ALGM.02

Standards Statement

Modify and model algorithms with repeating patterns to use iteration instead of repeated instructions such as spiral requirements.

Key Concepts

- Students can describe the problem and:
    - Identify what a user wants or expects (needs).
    - Identify specific conditions and the solution must meet (requirements).
    - Share their algorithm with peers or teachers for feedback.

Standard: CS.3-5.ALGM.03

Standards Statement

Decompose a problem or task into smaller components to develop an algorithm.

Key Concepts

- Students can break a complex problem into smaller, manageable parts to make it easier to understand.
- Students can turn each part of the problem into a clear, actionable step.
- Students can create an algorithm based on those actionable steps.

Standard: CS.3-5.ALGM.04

Standards Statement

Assess whether an algorithm meets specific conditions and does what the user expects, using defined requirements and feedback to evaluate and improve the solution.

Key Concepts

- Students can judge the effectiveness of their algorithm by:
    - Identifying how well the algorithm solves the problem.
    - How quickly and easily the algorithm solves the problem.
    - Determining whether the algorithm consistently produces correct results.

Standard: CS.3-5.ALGM.05

Standards Statement

Examine different perspectives, abilities, and points of view when designing algorithms and programs.

Key Concepts

- Students can examine different perspectives by:
    - Considering who will use the algorithm or program.
    - Understanding the different abilities (e.g. vision, mobility, reading level) people may have when using an algorithm or program.
    - Considering different backgrounds users might have.
    - Gathering feedback from others.

Standard: CS.3-5.PRG.01

Standards Statement

Develop code from a student-created algorithm that includes sequence, events, iteration, and selection, and organize code into reusable blocks or simple procedures to reduce repetition or complete a task.

Key Concepts

- Students can implement their algorithm using a computer language or an unplugged activity such as using cards that represent each step of the algorithm and arranging them in order.

Standard: CS.3-5.PRG.02

Standards Statement

Debug errors in a program that includes sequence, events, iteration, selection, and reusable code blocks or procedures.

Key Concepts

- Students can find and fix mistakes in a program such as:
    - Steps that are out of order.
    - The program doesn't respond to a trigger.

    - An iteration or loop runs too many or too few times.
- Students use debugging strategies that may include:
    - Reading the code aloud.
    - Showing what the program is doing at each step.
    - Checking one section at a time.
    - Having others review the program.
    - Trying different inputs to see if the program runs correctly.

Standard: CS.3-5.PRG.03

Standards Statement

Create a unique program by modifying other programs or incorporating portions of other programs into one's own work to develop something new or add more advanced features.

Key Concepts

Standard: CS.3-5.PRG.04

Standards Statement

Collaborate with peers to design, implement, document, and review a programming project that uses simple abstraction (e.g., named procedures) to express an idea or solve a problem.

Key Concepts

- Students can define key roles for team members that may include planner, coder, debugger, or presenter.
- Students can use active listening, respectful feedback and compromise to make sure all members of a team can contribute.

Standard: CS.3-5.PRG.05

Standards Statement

Collaborate with peers to design, implement, document, and review a programming project that expresses an idea or solves a problem.

Key Concepts

- Students can design a program together by brainstorming ideas, identifying problems, and planning an algorithm.
- Students can implement code together by assigning roles, working in pairs or groups, developing the program using smaller steps and testing.
- Students can document the program by adding comments to code, creating a project journal, or using diagrams and screenshots to show how the program works.

Standard: CS.3-5.DAA.01

Standards Statement

Organize data into tables where rows represent a "record" and columns represent attributes.

Key Concepts

- Students recognize that a record is one complete set of data about a thing (e.g. one student, one animal, one book).
- Students recognize that an attribute is a detail or feature of one thing (e.g. name, age, color).

Standard: CS.3-5.DAA.02

Standards Statement

Create a data visualization and a brief narrative to report the process and results of a data investigation.

Key Concepts

- Students can conduct a data investigation by asking a question, collecting data, organizing and analyzing the data, and sharing what they have learned.
- Students can create data visualizations such as bar graphs, line graphs, pie charts, and pictographs.
- Students can choose and label a data visualization that is appropriate for the data investigation.

Standard: CS.3-5.DAA.03

Standards Statement

Design a data collection approach that thoughtfully considers and accommodates the diverse needs, perspectives, and experiences of individuals from a variety of backgrounds and groups.

Key Concepts

- Students understand that data should represent everyone fairly since people have different experiences, preferences, and needs.
- Students can design a data collection that:
    - Uses fair questions avoid bias or leading questions.
    - Include multiple ways of collecting data (e.g. written surveys, interviews, visual choices)
    - Sample diverse groups.

Standard: CS.3-5.DAA.04

Standards Statement

Explain design choices, identify limitations in the data, describe who produced the data and why, and distinguish between observed patterns and claims about cause and effect.

Key Concepts

- Understand how data is created**:** Explain the design choices behind the data, who produced it, and the purpose or motivation for collecting it.
- Recognize data limitations**:** Identify gaps, biases, or constraints in the data that may affect how accurately it represents a situation.

- Interpret patterns responsibly: Distinguish between patterns you can observe in the data and any claims about cause-and-effect, which require stronger evidence.

Standard: CS.3-5.DAA.05

Standards Statement

Consider whose perspectives may be missing, how question wording or measurement methods may introduce bias, and how personal assumptions or confirmation bias may influence interpretation.

Key Concepts

- Identify missing perspectives: Consider which voices, experiences, or groups may not be represented in the data and how that absence shapes understanding.
- Recognize sources of bias in data collection: Examine how question wording, measurement tools, or data-gathering methods might unintentionally favor certain responses or viewpoints.

  Reflect on personal interpretation biases: Acknowledge how your own assumptions or confirmation bias can influence how you read, interpret, or draw conclusions from the data.

Standard: CS.3-5.CSS.01

Standards Statement

Articulate steps to implement common solutions for hardware and software issues using accurate terminology.

Key Concepts

- Students can troubleshoot hardware and software issues by:
  - Identifying the problem.
  - Checking connections or settings.
  - Restarting the device.
  - Asking for help or looking up solutions.
  - Using correct terminology.

Standard: CS.3-5.CSS.02

Standards Statement

Explain how computing devices such as Bluetooth and GPS connect to the Internet using wires or wireless signals.

Key Concepts

- Students understand that the internet allows computers and devices to communicate and share information.
- Students understand that computers can be connected to the internet using wires like the Ethernet and radio waves through Wi-Fi.

Standard: CS.3-5.CSS.03

Standards Statement

Illustrate how people access the global network of the Internet to access information and communicate with each other.

Key Concepts

- Students understand that people use search engines to find answers to questions by visiting websites, watching videos, reading articles, and exploring maps.
- Students understand that people use email, messaging apps, video calls, and social media to communicate with others online.

Standard: CS.3-5.CSS.04

Standards Statement

Evaluate how authentication, different levels of access, and security measures help protect information on personal and public devices and networks.

Key Concepts

- Students understand that authentication verifies that someone is who they say they are using passwords, PINs, and biometrics to prevent unauthorized access to personal information.
- Students understand that access levels give different users different permissions to keep sensitive information safe and ensure people only do what they are allowed to.
- Students understand that security practices protect devices and networks to protect data from hackers, viruses, and accidental loss.

Standard: CS.3-5.CSS.05

Standards Statement

Justify the importance of monitoring and updating security measures to prevent unauthorized access to information and other harms.

Key Concepts

- Students understand that monitoring is important because new viruses, scams, and hacking methods appear.
- Students understand that updating fixes bugs and security holes that attackers might exploit.

Standard: CS.3-5.CSS.06

Standards Statement

Describe the benefits and harms of widely used computing platforms to an individual's life and human connections.

Key Concepts

- Students know that computing platforms include social media, messaging apps, video platforms, and learning platforms.
- Students understand that some benefits of these platforms include:

    - Staying in touch with friends and family.
    - Accessing educational materials.
    - Sharing art, music, stories and ideas.
- Students understand that some harm or risks include:
    - Create distractions.
    - Cyberbullying.
    - Privacy risks.
    - Misinformation.
    - Social pressure.

Standard: CS.3-5.CSS.07

Standards Statement

Recognize examples of personal information (name, address, school, birthday) that should not be shared online without adult permission.

Key Concepts

- Know what personal information is: Things like your name, address, school, and birthday are private and should be protected.
- Be careful what you share online: These details should not be posted or sent online without checking with a trusted adult first.
- Ask an adult for help**:** you're not sure whether something is safe to share, always ask a grownup before you do it.

Standard: CS.3-5.CSS.08

Standards Statement

Describe basic password management strategies (e.g., using a password manager with adult guidance, not sharing passwords with friends).

Key Concepts

- Keep passwords safe and organized: Use tools like a password manager *with help from a trusted adult* so you don't have to remember everything on your own.
- Never share passwords with friends: Passwords are private and should only be shared with a grownup who helps keep you safe.
- Use good habits to stay secure: Make strong passwords and store them safely so no one else can get into your accounts.

Standard: CS.3-5.CAS.01

Standards Statement

Examine how a computing innovation changed the way people lived, worked, or communicated over time.

Key Concepts

- Students can identify computing innovations such as smartphones, email, video conferencing, search engines, GPS navigation and online learning platforms.
- Students can provide examples of impacts such as faster communication, remote jobs, streaming entertainment, and fitness trackers.

Standard: CS.3-5.CAS.02

Standards Statement

Evaluate how people make choices about the use of emerging technologies, based on their needs and the consequences.

Key Concepts

- Students can identify emerging technologies such as Artificial Intelligence (AI), Virtual Reality (VR), Smart home devices, wearable tech, and self-driving cars.
- Students can identify needs that might include convenience, health, learning, and safety.
- Students can identify some positive consequences of use such as easier communication, better access to information, improved health tracking, and working more efficiently.
- Students can identify some negative consequences of use such as privacy concerns, over-reliance on technology, screen time and distraction, and job changes or loss.

Standard: CS.3-5.CAS.03

Standards Statement

Explain how computing technologies and skills are used across different industries.

Key Concepts

- Students can identify how computing technologies and skills are being used in industries such as education, healthcare, agriculture, entertainment, transportation, retail, science and research, or construction.

Appendix C: Grade 6-8 Key Concepts

Standard: CS.6-8.ALGM.01

Standards Statement

Create visual or textual representations of algorithms that include sequence, events, iteration, selection, and variables to solve a problem or complete a task.

Key Concepts

- Students understand that an algorithm is step-by-step set of instructions to solve a problem or complete a task.
- Students understand that sequence is the order in which steps are executed (e.g. recipe, set of steps to draw a shape).
- Students understand that events are actions that trigger a response (e.g. Clicking a button starts a new game).
- Students understand that iteration is repeating a set of instructions until a condition is met (e.g. "Repeat 10 times" or "While not at the end").
- Students understand that selection is making decisions based on conditions (e.g. "If it's raining, take an umbrella; else, wear sunglasses").
- Students understand that are storage for data that can change (e.g. score in a game, temperature reading).
- Students create visual representations of an algorithm (e.g. block-code or flowcharts) or textual representations (e.g. pseudocode, beginner-friendly languages).

Standard: CS.6-8.ALGM.02

Standards Statement

Optimize visual representations of algorithms.

Key Concepts

- Students can use strategies for optimizing algorithms that may include:
    - Organizing the sequence in a logical flow.
    - Eliminating redundant steps.
    - Breaking complex steps into smaller useable pieces.
    - Designing so the algorithm can grow as a task grows.
    - Adding comments or labels.

Standard: CS.6-8.ALGM.03

Standards Statement

Refine algorithms iteratively through user feedback to improve usability, accessibility, and user experience.

Key Concepts

- Students seek input from users about how well the algorithm works.
- Students make improvements in usability by methods such as:
    - Improving usability through clear instructions or logical flow.
    - Improving accessibility by making sure all users can interact with the algorithm.
    - Improving the overall feeling and satisfaction a user gets from using the algorithm.

Standard: CS.6-8.ALGM.04

Standards Statement

Demonstrate the correctness of algorithms for given inputs.

Key Concepts

- Students demonstrate correctness of an algorithm through approaches that might include:
    - Checking to make sure the algorithm has the correct outputs for a variety of inputs.
    - Stepping through the algorithm manually.
    - Checking the algorithm with multiple inputs including edge cases.
    - Explaining why the algorithm works for all valid inputs.
    - Identifying when an algorithm gives incorrect results.

Standard: CS.6-8.ALGM.05

Standards Statement

Describe common societal impacts, ethical issues, and biases of deterministic and probabilistic algorithms.

Key Concepts

- Students can identify common societal impacts of algorithms such as:
    - Replacing repetitive tasks.
    - Changing job markets.
    - Decision making in healthcare, education, and law.
    - Improved access to services.
- Students can identify common ethical issues related to algorithms such as:
    - Privacy and use of personal data.
    - Transparency on how decisions are made.
    - Accountability when algorithms make a mistake.
- Students can identify common biases of algorithms such as:
    - Learning from biased data.
    - Not working well for underrepresented groups.
    - Reinforcing bias over time.

Standard: CS.6-8.PRG.01

Standards Statement

Develop code from student-created algorithms that include sequence, events, iteration, selection, and variables to express ideas, complete a task, or solve a problem.

Key Concepts

- Students can identify a task or problem, create an efficient algorithm to complete the task or solve the problem and develop working code based on the algorithm.

Standard: CS.6-8.PRG.02

Standards Statement

Analyze how a segment of code works by identifying procedures, parameters, variables, and their roles in implementing abstraction**.**

Key Concepts

- Students should be able to identify the roles of key components that may include:
    - Organization of the code.
    - Function of each part of the code.
    - Flow of the program based on conditions and loops.
    - Movement of data throughout the code.
    - The purpose of reusable code.
    - How the code receives data and communicates results.

Standard: CS.6-8.PRG.03

Standards Statement

Use appropriate data types to store, update, and evaluate data within a program.

Key Concepts

- Students should know how to use data types that may include:
    - String.
    - Integer.
    - Float.
    - Boolean.
    - List/Array.

Standard: CS.6-8.PRG.04

Standards Statement

Modify existing programs that incorporate sequence, selection, and iteration.

Key Concepts

- Students can identify a clear goal that they wish to attain by making changes.
- Students should be able to understand what the existing code does before making changes.

- Students should be able to describe how specific changes may change the program output.
- Students should be able to identify and fix errors introduced through changes in a program.

Standard: CS.6-8.PRG.05

Standards Statement

Use reference documentation, libraries, APIs, and development tools to implement prewritten functions or services within a program.

Key Concepts

- Students understand how to read official documentation for programming languages.
- Students can use search engines effectively to find solutions, tutorials, or examples.
- Students can use built-in or external tools to identify and fix errors.
- Students should understand how to save, track, and manage changes to code.
- Students can judge whether an online resource is reliable and appropriate.

Standard: CS.6-8.PRG.06

Standards Statement

Use standard practices to test, debug by tracing, document, and peer-review code.

Key Concepts

- Students should be able to apply standard practices such as:
    - Reading error messages.
    - Inserting temporary outputs to check variable values or flow.
    - Tracing code step-by-step.
    - Checking for common mistakes.
    - Testing with multiple inputs.
    - Using built-in debugging tools.

Standard: CS.6-8.PRG.07

Standards Statement

Modify a program to improve readability and usability testing.

Key Concepts

- Students should be able to improve usability and accessibility through techniques such as:
    - Making a program easy to navigate and understand.
    - Making sure user input is appropriate and handled safely.
    - Giving users clear responses to their actions.
    - Making programs usable by people with different abilities.
    - Keeping design and behavior predictable.
    - Getting feedback from others.

Standard: CS.6-8.PRG.08

Standards Statement

Apply inclusive collaboration practices to support all stages of programming, from planning to testing.

Key Concepts

- Students can apply inclusive collaboration practices such as:
    - Shared planning and goal setting.
    - Respectful communication.
    - Equitable participation.
    - Collaborative debugging.
    - Peer review and feedback.
    - Reflection and iteration.

Standard: CS.6-8.PRG.09

Standards Statement

Document a program, using comments, descriptive names, and structured guides, to improve readability, enable collaboration, and explain complex logic or code intent.

Key Concepts

- Students use comments, descriptive naming, and consistent formatting on sections of documents in a program.

Standard: CS.6-8.DAA.01

Standards Statement

Collect and generate multiple types of data—quantitative, qualitative, spatial, audio, visual, text, and network—using computational tools.

Key Concepts

- Students can use computational tools to collect data that might include:
    - Spreadsheets.
    - Block-based programming.
    - Text-based programming.
    - Online forms and surveys.
    - Microcontrollers for sensor-based data collection.

Standard: CS.6-8.DAA.02

Standards Statement

Analyze how a segment of code works by identifying and describing the roles of sufficient and statistically significant data.

Key Concepts

- Students should know that precision refers to how close data points are to each other.

- Students should know that granularity refers to the level of detail in the data.
- Students should understand that a need for more accurate data can increase the cost of collecting and storing it.
- Students should understand that more detailed data takes up more computer memory and storage.
- Students should understand that too little data can hide patterns while too much data can overwhelm or confuse.

Standard: CS.6-8.DAA.03

Standards Statement

Use computational tools such as spreadsheets to collect and organize quantitative and qualitative data.

Key Concepts

- Students should differentiate between quantitative data (numerical values) and qualitative data (descriptive or categorical values).
- Students should be able to organize data into columns and rows in computational tools.

Standard: CS.6-8.DAA.04

Standards Statement

Use computational tools to organize, filter, group, and aggregate data.

Key Concepts

- Students can use computational tools to organize data into columns and rows so they can be filtered by specific criteria, grouped based on characteristics, and aggregated using calculations like totals, averages, counts, or percentages.

Standard: CS.6-8.DAA.05

Standards Statement

Use computational tools to manipulate data.

Key Concepts

- Students can use computational tools to change the format of data and apply formulas and functions to the data.

Standard: CS.6-8.DAA.06

Standards Statement

Identify errors in data.

Key Concepts

- Students should recognize whether data makes sense and follows a uniform format or structure.
- Students should recognize when data is missing or is duplicated.

- Students should detect whether values are unusually high or low.

Standard: CS.6-8.DAA.07

Standards Statement

Use computational tools to identify relationships among variables in a dataset and make classifications or predictions.

Key Concepts

- Students can use computational tools to sort, filter, and use graphs or scatter plots to help classify data and make predictions related to the dat.

Standard: CS.6-8.DAA.08

Standards Statement

Create data visualizations to demonstrate how different design choices can affect a data visualization's clarity, visual appeal, accessibility, and capacity to accurately communicate insights from data investigations.

Key Concepts

- Students can select specific types of charts to communicate insights from data such as:
  - Bar charts for comparison
  - Line graphs for trends over time.
  - Pie charts for proportions.
  - Scatter plots for relationships.
- Students can use visualization designs such as annotations, color highlights, callouts, and scale to emphasize key findings.

Standard: CS.6-8.DAA.09

Standards Statement

Analyze how decisions made during data collection, data processing, data analysis, and data presentation can lead to biased data, misleading conclusions, and compromised AI models.

Key Concepts

- Students understand that data bias can result from unfairness introduced when gathering data or assumptions made when cleaning or organizing data.
- Students understand that misleading conclusions can result from flawed assumptions during data analysis and using visual or narrative techniques that distort the truth.
- Students understand that biased or flawed data can lead to unfair or inaccurate AI behavior.

Standard: CS.6-8.DAA.10

Standards Statement

Analyze the societal impacts of data-driven algorithms and computational systems, including AI.

Key Concepts

- Students should be able to analyze and discuss societal issues related to computational systems including AI that may include:
    - Algorithms can reflect or amplify biases in the data they are trained on.
    - AI systems often rely on large amounts of personal data.
    - Not everyone has equal access to AI tools or benefits.
    - Algorithms are used to make or support decisions in many areas.
    - AI can be used to spread false information or manipulate opinions.
    - Large-scale AI systems require significant computing power and large supplies of energy.

Standard: CS.6-8.CSS.01

Standards Statement

Describe the structure and organization of file systems, including file naming conventions, directories, and file permissions, and their role in storing and managing data.

Key Concepts

- Students should understand organization of file systems including:
    - File systems are like digital filing cabinets.
    - Directories (also called folders) are used to group files.
    - File names should be clear and descriptive.
    - Permissions control who can do what with a file.
- Students should understand that well organized file systems:
    - Help prevent lost files.
    - Keep data organized and secure.
    - Make it easier to collaborate and share.

Standard: CS.6-8.CSS.02

Standards Statement

Apply basic troubleshooting processes to identify and fix common hardware and software issues.

Key Concepts

- Students can identify common hardware issues such as:
    - Loose cables.
    - Power problems.
    - Peripheral issues.
- Students can identify common software issues such as:
    - Frozen programs.
    - Error messages.
    - Slow performance.
    - App won't open.
- Students can use a troubleshooting process where they identify the problem, check for simple solutions first, try one fix at a time, test after each step, and ask for help if needed.

Standard: CS.6-8.CSS.03

Standards Statement

Explain **the foundational architecture of the Internet, including its core components, governing protocols, and distribution methods.**

Key Concepts

- Students should know that the internet is a global network of computers that can communicate with each other.
- Students should know that the components of the internet include:
    - Client – A device like a laptop or phone that requests information.
    - Server – A powerful computer that stores and sends information to clients.
    - Router – A device that directs data between networks.
    - Switch – A device that connects devices within the same network.
    - IP Address – A unique number for each device on the Internet.

Standard: CS.6-8.CSS.04

Standards Statement

Model how information travels securely across digital networks through the use of physical and software tools.

Key Concepts

- Students can model visually or in writing how information travels across digital networks including:
- Physical tools that help data travel such as cables, routers, modems, and Wi-Fi access points.
- Software tools that keep data secure such as encryption, firewalls, antivirus software, and Virtual Private Networks (VPNs).

Standard: CS.6-8.CSS.05

Standards Statement

Classify common types of cyberattacks, including social engineering and malware.

Key Concepts

- Students understand that a cyberattack is when someone tries to steal, damage, or access information on a computer or network without permission.
- Students understand that social engineering tricks people into giving away private information. Students understand that malware is software designed to harm or steal data.

Standard: CS.6-8.CSS.06

Standards Statement

Examine differences in access to computing systems, based on personal and social factors, including physical ability, geographic location, socioeconomic status, and age.

Key Concepts

- Students identify different ways that adaptive technology such as screen readers, voice recognition, or other input devices can help people with differing abilities access computing systems.
- Students understand how rural or remote areas may have fewer tech resources than urban areas.
- Students understand that family income may reduce access to computers, internet service, and tech support.
- Students understand that age affects comfort level, experience, and access to training related to computing systems.

Standard: CS.6-8.CSS.07

Standards Statement

Analyze how the elements of governing protocols, and distribution methods enable scalability, compatibility, and resilience surrounding the internet in a changing technological landscape.

Key Concepts

- Understand how internet protocols support growth: Analyze how rules like IP, TCP, and routing protocols allow the internet to scale as more devices, users, and services come online.
- Examine distribution methods for compatibility**:** Consider how decentralized systems, packet switching, and standardized formats ensure different networks and technologies can work together smoothly.
- Recognize how design choices build resilience**:** Explore how redundancy, fault-tolerant structures, and adaptable protocols help the internet remain reliable and functional even as technology evolves or failures occur.

Standard: CS.6-8.CAS.01

Standards Statement

Analyze intended and unintended impacts of historical computing technologies on society and the environment.

Key Concepts

- Students identify intended impacts on society that may include:
    - Improved communication.
    - Faster access to information.
    - Automation of tasks.
    - New careers and industries.

    - New education and learning tools.
- Students identify unintended impacts on society that may include:
    - Unequal access to technology.
    - Job displacement.
    - Cyberbullying.
    - Privacy concerns.
    - Screen time and mental health.

Standard: CS.6-8.CAS.02

Standards Statement

Evaluate how design decisions in emerging technologies influence user experiences differently across different communities.

Key Concepts

- Students understand that design decisions such as how a product looks and works, who it's made for, and what features are included can influence user experience.

Standard: CS.6-8.CAS.03

Standards Statement

Contrast the features, functionality, and characteristics of emerging technologies with technologies that came before.

Key Concepts

- Students identify historical features, functionality, and characteristics that contrast with emerging technologies such as:
    - Limited customization.
    - Single purpose.
    - Larger.
    - Slower.
- Students identify features, functionality, and characteristics of emerging technologies such as:
    - Advanced features like voice control and facial recognition.
    - Multi-functional.
    - Smaller.
    - More powerful.

Standard: CS.6-8.CAS.04

Standards Statement

Examine how changes in technology can create new jobs or change how people work.

Key Concepts

- Students identify new jobs created such as:
    - App Developer
    - Social Media Manager.

- Drone Operator.
- Cybersecurity Analyst.
- AI Trainer.

- Students identify ways work has changed such as:
  - Remote work.
  - Automation.
  - Collaboration tools.
  - Digital skills.

Standard: CS.6-8.CAS.05

Standards Statement

Illustrate how technology- driven challenges require integrating computing knowledge and skills with expertise from diverse fields.

Key Concepts

- Students understand how computing connects with other fields like:
  - Science (e.g., using simulations to study weather).
  - Math (e.g., algorithms and data analysis).
  - Art and Design (e.g., user-friendly interfaces).
  - Social Studies (e.g., understanding how tech affects communities).
  - Health (e.g., wearable tech for tracking fitness or medical data).

Appendix D: High School Foundational Key Concepts

Standard: CS.HSF.ALGM.01

Standards Statement

Develop algorithms that include variables, data, and storage, using authentic, real-world data.

Key Concepts

- Students use authentic datasets such as weather data, sports statistics, environmental sensor data, or census data.
- Students understand the need to clearly define input and output.
- Students can successfully sequence, select and loop a build using data.

Standard: CS.HSF.ALGM.02

Standards Statement

Describe the differences between deterministic algorithms and probabilistic algorithms.

Key Concepts

- Students understand that a deterministic algorithm always produces the same output for a given input while a probabilistic algorithm uses randomness and probability in its process so may produce different outputs or take different paths even with the same input.

Standard: CS.HSF.PRG.01

Standards Statement

Convert an algorithm written in pseudocode into a program that uses sequence, selection, iteration, procedures with parameters, and lists.

Key Concepts

- Students understand that pseudocode is a way to describe an algorithm using plain language and programming-like structure.
- Students understand that converting pseudocode into a program may involve:
    - Sequence – a series of instructions.
    - Selection – use of if/else statements to make choices.
    - Iteration – repeat actions using for or while loops.
    - Procedures with parameters – functions or procedures that take input values (parameters) and return results.
    - Lists – used to store multiple values in one variable.

Standard: CS.HSF.PRG.02

Standards Statement

Analyze a segment of code to explain how procedural abstraction, parameters, and modular design contribute to functionality and readability.

Key Concepts

- Students can read and interpret code to figure out what problem it solves, what output it produces, and how it transforms data.
- Students can communicate their interpretation in plain language.

Standard: CS.HSF.PRG.03

Standards Statement

Evaluate and integrate external libraries or APIs using documentation, assessing how abstraction enables code reuse and interoperability.

Key Concepts

- Students apply ethical and legal considerations when assessing opportunities to incorporate external code. The considerations include:
    - Open-source licenses
    - Plagiarism vs. proper attribution
    - Responsible use of AI-generated or community-sourced code.
- Students evaluate and test external code by:
    - Testing code in isolation before integrating.
    - Checking for security risks or bugs.
    - Understanding performance implications.

Standard: CS.HSF.PRG.04

Standards Statement

Refactor or redesign a program to improve functionality, usability, accessibility, safety, accuracy, or inclusivity through improved abstraction and modularization.

Key Concepts

- Students can improve a program using techniques such as:
    - Adding new features.
    - Adding helpful messages, tooltips, or instructions.
    - Including screen reader support or alt text for images.
    - Validating user input to prevent crashes or exploits.
    - Using appropriate data types and algorithms.
    - Avoiding stereotypes or biased assumptions in content or logic.

Standard: CS.HSF.PRG.05

Standards Statement

Discuss how a program does and does not address a given problem.

Key Concepts

- Students can develop an argument in plain language by using techniques such as:
    - Understanding the original problem the program intended to solve.

- Using specific examples from the program to support claims.
- Identifying missing features or poor performance.
- Using clear criteria to assess the program's effectiveness.
- Making a clear and logical argument.
- Using testing to support or challenge claims.
- Connecting the program's impact on real-world needs or users.

Standard: CS.HSF.PRG.06

Standards Statement

Apply correct attribution to intellectual property.

Key Concepts

- Students understand the concept of intellectual property such as:
  - Copyright.
  - Trademarks.
  - Patents.
  - Open-source licenses.
- Students give proper credit to the original creator of content or code by including the author's name, the source, and the license.

Standard: CS.HSF.DAA.01

Standards Statement

Describe the differences among nominal, ordinal, discrete, and continuous data and how each type of data might be generated and used in data analysis.

Key Concepts

- Students understand that nominal data is categorical with no inherent order that is often generated through user input and can be analyzed using frequency counts, mode, or pie charts and bar graphs.
- Students understand that ordinal data is categorical with a meaningful order that, but with no consistent differences between values. Ordinal data is often generated through surveys, assessments, or rankings and analyzed using median, mode, bar charts, or non-parametric statistics.
- Students understand that discrete data is numerical data that can only take specific, separate values. Discrete data is usually generated through counting events, objects, or occurrences and can be analyzed using mean, median, mode, bar graphs, dot plots, or probability distribution.
- Students understand that continuous data is numerical data that can take any value within a range and often involves measurement. Continuous data is often generated through measuring with tools or sensors and can be analyzed using mean, medial, standard deviation, histograms, line graphs, and regression analysis.

Standard: CS.HSF.DAA.02

Standards Statement

Design a process that uses computational tools to collect data to answer a question, make classifications, or make predictions.

Key Concepts

- Students define a clear question or goal that can be answered using computational tools such as:
  - Spreadsheets.
  - Programming languages.
  - Sensors.
  - Online forms.

Standard: CS.HSF.DAA.03

Standards Statement

Analyze the consequences of using data in AI/ML applications, including how biased training data can lead to biased output and reinforce societal inequalities and injustices with misinformation and disinformation.

Key Concepts

- Students understand that bias occurs when data reflects unfair or unbalanced representation of people, behaviors, or outcomes through underrepresentation of certain groups and human bias in labeling or collecting data.
- Students understand that biased output can lead to discrimination, exclusion, and misinformation.

Standard: CS.HSF.DAA.04

Standards Statement

Evaluate the societal, environmental, and ethical implications of large- scale data collection and usage.

Key Concepts

- Students understand that societal implications can include concerns about consent, discrimination against certain groups, and questions about who benefits and who is at risk.
- Students understand that environmental implications can include concerns about data center and AI training models the consume massive amounts of electricity or encourage efficient algorithms and green data centers.
- Students understand that ethical implications can include concerns about informed consent, bias and fairness, misinformation, and data ownership.

Standard: CS.HSF.CSS.01

Standards Statement

Apply systematic troubleshooting techniques to identify, diagnose, and resolve issues in computing systems.

Key Concepts

- Students apply a systematic approach to troubleshooting that includes:
  - Identifying the problem.
  - Isolating the cause.
  - Testing possible solutions.
  - Implementing a fix.
  - Verifying the solution.
  - Documenting the process.

Standard: CS.HSF.CSS.02

Standards Statement

Debate the trade-offs of global access to computing systems for society in terms of societal norms, interactions, and digital engagement.

Key Concepts

- Students understand that the benefits of global access to computing systems can include:
  - Increased connectivity.
  - Access to information and education.
  - Economic opportunities
  - Civic engagement.
- Students understand that the challenges of global access to computing systems can include:
  - Unequal access to technology.
  - Erosion of local customs and languages.
  - Surveillance and privacy.
  - Misinformation and polarization.

Standard: CS.HSF.CSS.03

Standards Statement

Evaluate the rationales behind laws, policies, and best practices governing the design and use of computing systems.

Key Concepts

- Students understand that the purpose of laws and policies in computing may include:
  - Protecting rights.
  - Ensuring equity.
  - Promoting safety.
- Students identify examples of best practices that can include:
  - Published standards.
  - Systems built with privacy and security in mind from the start.
  - Inclusive design that ensures accessibility for users with diverse needs.

Standard: CS.HSF.CSS.04

Standards Statement

Analyze the costs of cybersecurity breaches and social engineering attacks for individuals, industries, and governments.

Key Concepts

- Students understand that costs associated with cybersecurity breaches and social engineering attacks can include:
    - Financial costs.
    - Data loss and privacy.
    - Stress, anxiety, and loss of trust.
    - Operational disruptions.

Standard: CS.HSF.CSS.05

Standards Statement

Investigate the societal and environmental impacts of computing systems and the physical infrastructure that supports them.

Key Concepts

- Students recognize examples of infrastructure that include data centers, networks, and devices.
- Students identify societal impacts that may include:
    - Access and equity.
    - Cultural shifts.
    - Workforce and automation.
- Students identify environmental impacts that may include:
    - Energy consumption.
    - E-waste.
    - Resource extraction.

Standard: CS.HSF.CSS.06

Standards Statement

Distinguish among the different types of cyberattacks that affect information security for individuals and organizations.

Key Concepts

- Students differentiate between cyberattacks such as:
    - Malware.
    - Phishing.
    - Social engineering.
    - Denial of service (DoS).
    - Brute force attacks.
    - Zero-day exploits.
    - Insider threats.

Standard: CS.HSF.CSS.07

Standards Statement

Diagram computing systems that integrate security protocols, incorporating user-centered design principles such as user research, prototyping, and iterative design.

Key Concepts

- Students use diagramming skills to visually represent information that may include:
  - Use of symbols.
  - Data flow arrows.
  - Annotations.
- Students understand security protocols such as:
  - Authentication.
  - Authorization.
  - Encryption.
  - Firewalls.
  - Antivirus.
  - Secure communication.
  - Logging and monitoring.

Standard: CS.HSF.CAS.01

Standards Statement

Differentiate major eras in computing history and key advancements by notable individuals and organizations and how they have led to present day systems.

Key Concepts

- Students recognize major eras and notable individuals or organizations that may include:
  - Pre-Computer Era (Before 1940s) – Notable individuals include Ada Lovelace and Charles Babbage.
  - Early Computers Era (1940s to 1950s) – Notable individuals include Alan Turing and Grace Hopper.
  - Mainframe Era (1950s to 1970s) – Notable organizations include IBM.
  - Personal Computer Era 9 (1970s to 1990s) – Notable individuals include Steve Jobs, Steve Wozniak, and Bill Gates.
  - Internet and Web Era (1990s to 2000s) – Notable individuals include Tim Berners-Lee.
  - Mobile and Cloud Era (2000s to 2010s) – Notable organizations include Apple, Google, Amazon Web Services.
  - AI and Big Data Era (2010s to Present) – Notable organizations include OpenAI, DeepMind, NVIDIA.

Standard: CS.HSF.CAS.02

Standards Statement

Evaluate policies and legislation designed to encourage ethical innovation and minimize societal risks associated with technology and why these policies are necessary. Key Concepts

- Students recognize major policies and frameworks such as:
  - European Union AI Act.
  - Bletchley Declaration.
  - UNESCO AI Ethics Recommendations.
  - National Defense Authorization Act (NDAA).
  - U.S. Sectoral Privacy Laws:
    - Children's Online Privacy Protection Rule (COPPA).
    - Gramm-Leach-Billey Act (GLBA).

Standard: CS.HSF.CAS.03

Standards Statement

Explain how the computing principles underlying emerging technologies are being used in innovative ways.

Key Concepts

- Students understand innovative uses of emerging technologies such as:
  - Cross-disciplinary impact.
  - Human-centered design.
  - Ethical innovation.
  - Generative AI

Standard: CS.HSF.CAS.04

Standards Statement

Describe how an emerging technology will impact an existing project or technology.

Key Concepts

- Students identify emerging technologies such as:
  - Artificial Intelligence.
  - Quantum Computing.
  - Blockchain.
  - Augmented/Virtual Reality.
  - Internet of Things (IoT).
  - 5G and Edge Computing.
- Students identify potential points of impact in:
  - Functionality.
  - Efficiency.
  - User experience.

- Security and privacy.
- Ethical and social implications.

Standard: CS.HSF.CAS.05

Standards Statement

Evaluate how computing knowledge and skills align with personal interests and career aspirations.

Key Concepts

- Students identify alignment by asking questions such as:
  - What computing skills do I really have?
  - What do I enjoy doing with technology?
  - What careers use those skills in ways that match my values and goals?

Standard: CS.HSF.CAS.06

Standards Statement

Investigate how professionals apply CS in their careers, drawing from their personal narratives.

Key Concepts

- Students ask questions of professionals that might include:
  - What inspired the professional to pursue CS?
  - What challenges did they face and overcome?
  - How do they use CS in their daily work?
  - What advice to they have for students?
  - How does CS impact all career sectors?