Colorado Academic Standards

Computer Science

COLORADO
Department of Education

ALL STUDENTS • ALL STANDARDS

Computer Science Standards Committee

Chairperson

Axel Reitzig
Robotics and Computer Science Coordinator
St. Vrain Valley School District

Members

Bobbie Bastian
Computer Science Teacher and TSA Advisor
Adams 12 Five Star Schools

Kristina Brown
Computer Science and Colorado Technical Education Teacher
Poudre School District

Sharon Combs
Instructional Technology Designer
Douglas County School District

William Heldman
Computer Science and Cybersecurity Teacher
Jefferson County School District

Brandon Petersen
Professional Development Strategist
Share Fair Nation/Mind Spark Learning

Chuck Powell
Computer Science Teacher
President, Computer Science Teachers Association, Front Range Chapter
Jefferson County School District

David Webb
Associate Professor
University of Colorado Boulder

Andrea Young
President and CEO
Colorado Technology Association

State Board of Education and Colorado Department of Education

Colorado State Board of Education

Angelika Schroeder (D, Chair)
2nd Congressional District
Boulder

Joyce Rankin (R, Vice Chair)
3rd Congressional District
Carbondale

Steve Durham (R)
5th Congressional District
Colorado Springs

Valentina (Val) Flores (D)
1st Congressional District
Denver

Jane Goff (D)
7th Congressional District
Arvada

Rebecca McClellan (D)
6th Congressional District
Centennial

Debora Scheffel (R)
4th Congressional District
Parker

Colorado Department of Education

Katy Anthes, Ph.D.
Commissioner of Education
Secretary to the Board of Education

Melissa Colsman, Ph.D.
Associate Commissioner of Education
Student Learning Division

Floyd Cobb, Ph.D.
Executive Director
Teaching and Learning Unit

CDE Standards and Instructional Support Office

Karol Gates
Director

Carla Aguilar, Ph.D.
Music Content Specialist

Ariana Antonio
Standards Project Manager

Joanna Bruno, Ph.D.
Science Content Specialist

Lourdes (Lulu) Buck
World Languages Content Specialist

Donna Goodwin, Ph.D.
Visual Arts Content Specialist

Stephanie Hartman, Ph.D.
Social Studies Content Specialist

Judi Hofmeister
Dance Content Specialist
Drama and Theatre Arts Content Specialist

Jamie Hurley, Ph.D.
Comprehensive Health Content Specialist
Physical Education Content Specialist

Raymond Johnson
Mathematics Content Specialist

Christine Liebe
Computer Science Content Specialist

Vince Puzick
Reading, Writing, and Communicating Content Specialist

Purpose of Computer Science

"We want students to understand what a computer can do, what a human can do, and why that's different." - Mark Guzdial, Professor of Computing, Georgia Tech

Colorado's economic vitality would greatly benefit from the implementation of comprehensive K-12 computer science education. There are literally thousands of computer science jobs in Colorado and only hundreds of Colorado college graduates to fill those positions. Additionally, Frey and Osborne (2013) estimate that 47% of current employment in all sectors of the economy will be replaced by technology in 10 to 20 years. Professionals in all disciplines will be more successful with knowledge and skills in computer science. Citizens will make more informed choices with foundational understanding of computer science. Students have been learning how to use computers for many years (digital literacy), computer science is a discipline in which students explore foundational concepts related to creating hardware, software, programming, and user interfaces. We owe it to students to prepare them adequately for future employment.

The creation of high school computer science standards in response to House Bill 16-1198 began in April 2017. After listening to focus groups around the state of Colorado and examining a professional review of national computer science standards, a volunteer citizen committee made up of professional computer science teachers, higher education professors, and private sector professionals designed the voluntary Colorado high school computer science standards. The committee used national standards from the Computer Science Teachers Association, the K12 Computer Science Framework, and other state computer science standards as references.

As practicing computer science educators, the committee sought to minimize the bulk of standards by concentrating them into three major areas. Colorado Essential Skills, replacing the 21st Century Skills, were used to augment and concentrate the computer science standards. Collaboration and communication are essential in the computer science classroom and in the private sector. However, the committee was careful to write standards that guided content and not instruction.

Even though technology presents educators with a rapidly changing landscape, long-lasting themes in computer science education have persisted and provide a robust foundation of learning. Computational thinking, computing systems and networks, and computer programming provide the bulk of content knowledge in computer science. Additional topics, providing students with opportunities to examine the impact technology has on privacy, communication, and society and exploring creative innovation are embedded in the standards.

The Association for Computing Machinery, or ACM, the premier computer science educational professional organization in the United States, details an entire code of ethics for computer science professionals. The committee has embedded computer science ethics into the three primary grade level expectations. While other national and state standards specifically name impacts on society as a standard, the committee decided that ethical considerations could be applied to curriculum in every standard.

Creativity is an area of human endeavor that is difficult to define, instruct, and assess. The committee recognized that computer science requires aspects of creativity, but was reticent to specifically add creativity as a standard. AP Computer Science Principles lists creativity as one of its seven big ideas. It is the committee's opinion that the inclusion of teaching the design process, among other aspects of computer science instruction, inherently fosters creativity.

Carefully crafted with responsive inclusion from public feedback, the committee presents Colorado's first Computer Science high school standards.

Statute

House Bill 16-1198 requires the State Board of Education to adopt secondary computer science standards that identify the knowledge and skills secondary students should acquire related to computer science, including computer coding, in one or more courses that qualify as a graduation requirement in either mathematics or science. Local education providers may choose to implement the standards adopted.

The voluntary nature of the Computer Science standards is different from all other academic standards that are required to be implemented.

Prepared Graduates in Computer Science

1. Develop, utilize and evaluate algorithms, to model and solve problems.
2. Systematically analyze a problem using decomposition and abstraction to formulate a solution.
3. Represent and analyze data in order to generate new knowledge and capability.
4. Use systems thinking to describe networks and common software and hardware components.
5. Develop systems solutions from a set of specifications to complete a design process.
6. Recognize and analyze security concepts.
7. Design and create programs, individually and collaboratively, for a variety of disciplines.
8. Create computational artifacts that consider security from tampering, malicious or otherwise.

Standards in Computer Science

Standards are the topical organization of an academic content area. The three standards of computer science are:

1. Computational Thinking
Includes concepts related to the use of algorithms and data in different ways to generate new knowledge and articulate solutions to real world problems.

2. Computing Systems and Networks
Includes concepts related to the development and communication between software and hardware, and systems thinking around data protection and recovery.

3. Computer Programming
Includes concepts related to creating computer programs and applications, working collaboratively to engage in client-based problem solving, and internet security.

Prepared Graduates:

1. Develop, utilize and evaluate algorithms, to model and solve problems.

Grade Level Expectation:

1. Computational thinking is used to create algorithmic solutions to real-world problems.

Evidence Outcomes

Students Can:

a. Identify and create different types of algorithms (sort, search, etc.).
b. Predict the outcome of different types of algorithms.
c. Create or adapt algorithms to solve problems for multiple purposes (e.g., personal interests, client needs).
d. Use an algorithm that involves mathematical operations and functions to solve problems.
e. Use an iterative approach to utilizing and/or developing an algorithm.
f. Recognize problems that cannot be solved computationally.
g. Identify and describe algorithms that exist within their personal lives.

Academic Context and Connections

Colorado Essential Skills:

1. Examine ways computers could make human activities easier and more efficient. (Entrepreneurial Skills: Critical thinking/problem solving; Inquiry and analysis)
2. Evaluate attempts to create a working algorithm. (Personal Skills: Self-awareness; Initiative/self direction)
3. Identify how algorithms can be used to solve social problems. (Civic/Interpersonal Skills: Communication; Global/cultural awareness).
4. Compare and contrast solvable and unsolvable computational problems. (Professional Skills: Information literacy; Use information/communications technologies; Perseverance/resilience).

Elaboration on the GLE:

1. Central to computational thinking are the processes of generalization and decomposition, with an eye toward the technology that will be used to solve the problem. This planning and abstraction process should also include students decomposing complex problems into manageable sub-problems that could potentially be solved with programs or procedures that already exist. As students develop algorithms, they should identify procedures and/or functions that are used multiple times within a program to repeat groups of instructions (CSTA 3A-AP-17 & 2-AP-14).

Computer Science Practices:

1. Fostering an Inclusive Computing Culture
2. Recognizing and Defining Computational Problems
3. Creating Computational Artifacts

Prepared Graduates:

1. Develop, utilize and evaluate algorithms, to model and solve problems.

Grade Level Expectation:

2. Algorithms can be represented and used in different ways (e.g., languages, diagrams, pseudocode).

Evidence Outcomes

Students Can:

a. Identify and compare different algorithms that can be used to solve the same problem.
b. Illustrate the flow of execution of an iterative algorithm (e.g., recursion).
c. Explain the value of heuristic algorithms to model ways to solve problems.
d. Adapt algorithms used in one problem to solve a related or different problem.
e. Use multiple methods to represent an algorithm (e.g., diagram, programming language, unplugged).

Academic Context and Connections

Colorado Essential Skills:

1. Create an algorithm to solve a client's needs. (Entrepreneurial Skills: Critical thinking/problem solving; Creativity/innovation; Inquiry and analysis)
2. Use pseudocode to represent an algorithm. (Personal Skills: Self-awareness; Initiative/self-direction)
3. Collaboratively develop an algorithm that could solve a social communication problem. (Civic/Interpersonal Skills: Communication; Global/cultural awareness)
4. Present an algorithmic solution using multiple methods. (Professional Skills: Information literacy; Use information/communications technologies; Productivity/accountability; Leadership)

Elaboration on the GLE:

1. Students should use pseudocode, diagrams and/or flowcharts to organize and sequence an algorithm that addresses a problem. Representing algorithms in alternative forms supports the planning phase of the design process and helps students see various ways to structure an algorithm (CSTA 2-AP-10).

Computer Science Practices:

1. Recognizing and Defining Computational Problems
2. Communicating about Computing

Prepared Graduates:

1. Develop, utilize and evaluate algorithms, to model and solve problems.

Grade Level Expectation:

3. Algorithm development and use is an ongoing process that involves adapting, critiquing and troubleshooting programs and/or processes.

Evidence Outcomes

Students Can:

a. Describe pros and cons of the performance of algorithms for the same task.
b. Use an iterative approach to developing an algorithm.
c. Test and troubleshoot so that algorithms produce reasonable results.

Academic Context and Connections

Colorado Essential Skills:

1. Use client feedback to develop an initial algorithm. (Entrepreneurial Skills: Critical thinking/problem solving; Creativity/innovation; Inquiry and analysis)
2. Develop a plan for using client feedback to improve an algorithm. (Personal Skills: Self-awareness; Initiative/self-direction)
3. Produce a progress report detailing algorithm development. (Civic/Interpersonal Skills: Communication; Global/cultural awareness)
4. Demonstrate debugging an algorithm. (Professional Skills: Perseverance/resilience)

Elaboration on the GLE:

1. Testing and refinement is the deliberate and iterative process of improving a computational artifact. This process includes debugging (identifying and fixing errors) and comparing actual outcomes to intended outcomes. Students should respond to the changing needs and expectations of end users and improve the performance, reliability, usability and accessibility of artifacts. For example, students could incorporate feedback from a variety of end users to help guide the size and placement of menus and buttons in a user interface (CSTA 3A-AP-21).

Computer Science Practices:

1. Testing and Refining Computational Artifacts
2. Creating Computational Artifacts
3. Recognizing and Defining Computational Problems

Prepared Graduates:

2. Systematically analyze a problem using decomposition and abstraction to formulate a solution.

Grade Level Expectation:

4. Large, complex problems can be broken down into smaller, more manageable components.

Evidence Outcomes

Students Can:

a. Demonstrate how the process of decomposition is iterative and used to solve problems.
b. Formulate possible solutions based on the decomposition of a problem.

Academic Context and Connections

Colorado Essential Skills:

1. Break down problems into smaller problems identifying patterns in each level. (Personal Skills: Self-awareness; Initiative/self-direction)
2. Propose a logical sequence to fix the problem. (Professional Skills: Information literacy; Use information/communications technologies; Perseverance/resilience; Productivity/accountability; Leadership)

Elaboration on the GLE:

1. At this level, students should decompose complex problems into manageable sub-problems that could potentially be solved with programs or procedures that already exist. For example, students could create an app to solve a community problem by connecting to an online database through an application programming interface (API) (CSTA 3A-AP-17).

Computer Science Practices:

1. Communicating about Computing
2. Recognizing and Defining Computational Problems

Prepared Graduates:

2. Systematically analyze a problem using decomposition and abstraction to formulate a solution.

Grade Level Expectation:

5. Abstraction is used to reduce complexity of larger problems by focusing on main ideas.

Evidence Outcomes

Students Can:

a. Describe how abstraction is central to computational thinking.
b. Identify and prioritize the most relevant parts of a problem while filtering out extraneous details.
c. Demonstrate different ways to represent key problem components.

Academic Context and Connections

Colorado Essential Skills:

1. Design a game with efficient use of code. (Entrepreneurial Skills: Critical thinking/problem solving; Creativity/innovation; Inquiry and analysis)
2. Develop a work plan and match essential activities with goals. (Personal Skills: Self-awareness; Initiative/self-direction)
3. Sort data using keywords and look for patterns to represent the essential nature of the data. (Civic/Interpersonal Skills: Communication; Global/cultural awareness)
4. Describe a model of a cat that has all the essential features of a cat without using the word cat to see if another student can guess the description.(Professional Skills: Information literacy; Use information/communications technologies)

Elaboration on the GLE:

1. Abstraction is a necessary part of modeling, problem solving and computational thinking; it requires the identification of key aspects of a given context to formulate and solve a problem of interest. Students might select an embedded device such as a car stereo, identify the types of data (radio station presets, volume level) and procedures (increase volume, store/recall saved station, mute) it includes, and explain how the implementation details are hidden from the user (CSTA 3A-CS-01).

Computer Science Practices:

1. Developing and Using Abstractions
2. Communicating about Computing
3. Testing and Refining Computational Artifacts

Prepared Graduates:

3. Represent and analyze data in order to generate new knowledge and capability.

Grade Level Expectation:

6. Data can be represented in different ways for storage and exchange.

Evidence Outcomes

Students Can:

a. Identify different types of data that are exchanged and produced by computers (e.g., protocols).
b. Represent data using multiple encoding schemes (e.g., RGB, Hex, HSB, ASCII, Unicode).
c. Evaluate the trade-offs for how data elements are organized and where data are stored (e.g., PNG/GIF, structured/unstructured).
d. Compare and contrast various data structures/techniques for storing and processing data (e.g., arrays, lists, tables).

Academic Context and Connections

Colorado Essential Skills:

1. Propose an app that utilizes a database.(Entrepreneurial Skills: Critical thinking/problem solving; Creativity/innovation; Inquiry and analysis)
2. Compare and contrast text or visual data coding schemes for student lockers.(Personal Skills: Self-awareness; Initiative/self-direction)
3. Examine data types in Thorn Spotlight software that helps fight human trafficking. (Civic/Interpersonal Skills: Communication; Global/cultural awareness)
4. Contribute to a group outcome regarding data storage for a project. (Professional Skills: Leadership)

Elaboration on the GLE:

1. People make choices about how data elements are organized and where data is stored (e.g., convert hexadecimal color codes to decimal percentages, ASCII/Unicode representation, and logic gates (CSTA 3A-DA-09)). These choices affect cost, speed, reliability, accessibility, privacy and integrity. Students should evaluate whether a chosen solution is most appropriate for a particular problem. Students might consider the cost, speed, reliability, accessibility, privacy and integrity tradeoffs between storing photo data on a mobile device versus in the cloud (CSTA 3A-DA-10).

Computer Science Practices:

1. Communicating about Computing
2. Creating Computational Artifacts
3. Testing and Refining Computational Artifacts

Prepared Graduates:

3. Represent and analyze data in order to generate new knowledge and capability.

Grade Level Expectation:

7. Many problems appropriate for solving with a computer are organized around patterns.

Evidence Outcomes

Students Can:

a. Analyze computer programs to identify patterns within the program.
b. Provide multiple versions of data visualization in order to deepen problem analysis.
c. Interpret and analyze data to make informed decisions.

Academic Context and Connections

Colorado Essential Skills:

1. Analyze Snapchat and Instagram for popular program patterns.(Entrepreneurial Skills: Critical thinking/problem solving; Creativity/innovation; Inquiry and analysis)
2. Represent personal grades in Tableau and SQL.(Personal Skills: Self-awareness; Initiative/self-direction)
3. Evaluate crime rates in Colorado from a variety of data sources.(Civic/Interpersonal Skills: Communication; Global/cultural awareness)
4. Provide suggestions for reducing crime based on the data evaluation and give a formal presentation or report.(Professional Skills: Information literacy; Use information/communications technologies; Perseverance/resilience; Productivity/accountability; Leadership)

Elaboration on the GLE:

1. One of the most powerful features of computational thinking is using technological tools to make sense of natural and social phenomena. Coding and analytic techniques can be used to identify and visualize patterns in complex data. For example, students could be asked to identify trends in a data set representing social media interactions, movie reviews or shopping patterns (CSTA 3B-DA-05).

Computer Science Practices:

1. Recognizing and Defining Computational Problems
2. Testing and Refining Computational Artifacts

Prepared Graduates:

3. Represent and analyze data in order to generate new knowledge and capability.

Grade Level Expectation:

8. Data from a computer program can be visually presented to better understand and articulate solutions to a problem.

Evidence Outcomes

Students Can:

a. Analyze computer output in different forms (e.g., plain text, CSV, graphs, images).
b. Design visualizations using the appropriate tool(s) with the end user in mind.
c. Provide multiple versions of data visualization in order to deepen problem analysis.

Academic Context and Connections

Colorado Essential Skills:

1. Use organization and visualization tools and techniques to identify patterns in data (Entrepreneurial Skills: Critical thinking/problem solving; Creativity/innovation; Inquiry and analysis; Risk taking)
2. Pose questions that can be explored with a given data set (Personal Skills: Initiative/self-direction)
3. Discuss how the intended audience for output might influence how to represent data (Civic/Interpersonal Skills: Global/cultural awareness)
4. Create appropriate visual representations to identify patterns and relationships in data (Professional Skills: Information literacy; Use information/communications technologies)

Elaboration on the GLE:

1. People transform, generalize, simplify and present large data sets in different ways to influence how other people interpret and understand the underlying information. Examples include visualization, aggregation, rearrangement and application of mathematical operations. People use software tools or programming to create powerful, interactive data visualizations and perform a range of mathematical operations to transform and analyze data. Students should model phenomena as systems, with rules governing the interactions within the system and evaluate these models against real-world observations. For example, flocking behaviors, queueing or life cycles (CSTA 3A-DA-11).

Computer Science Practices:

1. Recognizing and Defining Computation Problems
2. Creating Computational Artifacts
3. Fostering an Inclusive Computing Culture

Prepared Graduates:

4. Use systems thinking to describe networks and common software and hardware components.

Grade Level Expectation:

1. Communication between computers (and over the internet) can be configured in many different ways and consist of several hardware and software components.

Evidence Outcomes

Students Can:

a. Describe key protocols and underlying processes of internet-based services, (e.g., https) and discuss impact of technology change on communication protocols.
b. Illustrate and describe the basic components and various network types and topologies (e.g., personal, local, metropolitan, and wide).
c. Explain the difference between decimal, hexadecimal, octal and binary number formats and how they are used in computing environments.

Academic Context and Connections

Colorado Essential Skills:

1. Using note cards, demonstrate how a message can be sent and received using UDP, and then using TCP. (Entrepreneurial Skills: Critical Thinking/problem solving; Inquiry analysis; Risk taking)
2. Draw the star and bus topologies and explain the difference between the two. (Civic/Interpersonal Skills: Communication)
3. Use hexidecimal numbering to determine the color of paint on a wall in a picture in a website. (Professional Skills: Information Literacy; Use Information and Communications Technologies)

Elaboration on the GLE:

1. Computing is at its most powerful when devices are connected via a network. Networks are comprised of various hardware and software components that have specific functions within the network. For example, individual devices are assigned an address that uniquely identifies it on the network; routers function by comparing IP addresses to determine the pathways packets should take to reach their destination; and switches function by comparing MAC addresses to determine which computers or network segments will receive frames (CSTA, 3A-NI-04). Each device is assigned an address that uniquely identifies it on the network. Routers function by comparing IP addresses to determine the pathways packets should take to reach their destination. Switches function by comparing MAC addresses to determine which computers or network segments will receive frames. Students could use online network simulators to experiment with these factors (CSTA, 3A-NI-04).

Computer Science Practices:

1. Developing and Using Abstractions

Prepared Graduates:

4. Use systems thinking to describe networks and common software and hardware components.

Grade Level Expectation:

2. Computer hardware, the lowest level of a computer system, consists of many different parts, each providing a specialized function.

Evidence Outcomes

Students Can:

a. Explain the difference between memory and disk storage, internal and external storage, Random Access Memory (RAM), flash, cloud.
b. List and explain the common working parts of a computing device.
c. Explain how to maintain safety when working on PCs, e.g., electromagnetic precautions.
d. Describe how computing devices are engineered for fault tolerance and reliability, and identify potential sources of weakness (e.g., redundant power supplies, RAID, SAN/NAS connections).

Academic Context and Connections

Colorado Essential Skills:

1. Evaluate your computer for possible sources of failure. (Entrepreneurial Skills: Critical Thinking/problem-solving; Inquiry analysis; Risk taking)
2. Develop a poster describing personal safety when working with computers. (Personal Skills: Adaptability/flexibility; Perseverance/adaptability)
3. Brainstorm ways to improve the performance of an older computer using hardware upgrades. (Civic/Interpersonal Skills: Communication)
4. Research professional certifications and identify one that could be completed over the summer. (Professional Skills: Information Literacy; Use Information and Communications Technologies)

Elaboration on the GLE:

1. At its most basic level, a computer is composed of physical hardware and electrical impulses. A computing system is composed of components such as the central processor (executes commands), memory (for temporary storage of data), hard disk (stores data), mainboard (provides communication between components and peripherals) and network interface (communicates with other devices) and power supply (CSTA 3A-CS-02).

Computer Science Practices:

1. Developing and Using Abstractions

Prepared Graduates:

4. Use systems thinking to describe networks and common software and hardware components.

Grade Level Expectation:

3. Computer software is written for specific purposes.

Evidence Outcomes

Students Can:

a. Identify and differentiate between different kinds of software (e.g., operating systems vs. applications) and their purposes.
b. Explain what an operating system is, and why it is important for a computer or computing device (e.g., Linux, Windows, iOS).
c. Describe how software interacts with hardware to complete tasks.

Academic Context and Connections

Colorado Essential Skills:

1. Compare and contrast Linux, Macintosh, and Microsoft operating systems through a cost-benefit analysis.(Entrepreneurial Skills: Critical Thinking/problem solving; Inquiry analysis; Risk taking)
2. Compare and contrast user interfaces based on user beliefs and expectations. (Personal Skills: Adaptability/flexibility; Perseverance/adaptability)
3. Discuss the pros and cons for society of open source versus proprietary commercial software. (Civic/Interpersonal Skills: Communication)
4. Write a business plan for promoting open source or commercial software. (Professional Skills: Information Literacy; Use Information and Communications Technologies)

Elaboration on the GLE:

1. System software manages a computing device’s resources (CSTA 3A-CS-02). Students should recognize that there is a variety of software user interfaces, and that different software exists for different purposes (e.g., operating system vs. application).

Computer Science Practices:

1. Communicating about Computing

Prepared Graduates:

4. Use systems thinking to describe networks and common software and hardware components.

Grade Level Expectation:

4. Systems thinking is a way of holistically examining the various components and use cases that go into a given design.

Evidence Outcomes

Students Can:

a. Explain the integration of hardware, software and network communications components to create a networked system.
b. Summarize security approaches using a systems approach perspective.

Academic Context and Connections

Colorado Essential Skills:

1. Using Raspberry Pis, create a network system that will display a software, hardware, and network integrated system. (Entrepreneurial Skills: Critical Thinking/problem solving; Inquiry analysis; Risk taking)
2. Create a video demonstrating 10 basic network security habits. (Personal Skills: Adaptability/flexibility; Perseverance/adaptability)
3. Present and suggest the minimum network every household and business should have. (Civic/Interpersonal Skills: Communication)
4. Create a website that teaches other students how to practice helpful security habits for a computer system when using the internet. (Professional Skills: Information literacy; Use Information and Communications Technologies)

Elaboration on the GLE:

1. By itself, a computer is just a dumb piece of hardware. It is not until an operating system is loaded on to it that the computer becomes useful. The OS handles the operation of the hardware in conjunction with the software applications a user has loaded. System software manages a computing device’s resources so that software can interact with hardware. (CSTA 3A-CS-02) Systems thinking utilizes concepts and tools that helps people to understand the makeup of large systems, like computer networks, to meet user needs/requirements, and to make sure computer systems are secure.

Computer Science Practices:

1. Developing and Using Abstractions.
2. Recognizing and Defining Computational Problems
3. Communicating about Computing

Prepared Graduates:

5. Develop systems solutions from a set of specifications to complete a design process.

Grade Level Expectation:

5. Client considerations drive system design.

Evidence Outcomes

Students Can:

a. Identify client's problems/needs.
b. Articulate design requirements back to client.
c. Illustrate options for considerations and develop conceptual model.
d. Perform system analysis based on client considerations.

Academic Context and Connections

Colorado Essential Skills:

1. Interview a computer science project manager to devise an approach to interviewing prospective clients utilizing the diverse efforts of each student in class.(Entrepreneurial Skills: Critical Thinking/problem solving; Inquiry analysis; Risk taking)
2. Create a Gantt chart or other schedule for computing project completion. (Personal Skills: Adaptability/flexibility; Perseverance/adaptability; Initiative/self-direction)
3. Evaluate group progress on a computing project and provide constructive criticism. (Civic/Interpersonal Skills: Communication)
4. Evaluate any piece of software that you think might be ending its Software Development Life cycle (SDLC) and suggest changes.(Professional Skills: Information literacy; Use Information and Communications Technologies)

Elaboration on the GLE:

1. Software engineers plan and develop programs for broad audiences using a software life cycle process (CSTA 3B-AP-17). Similarly, systems architects use, plan and develop networks to meet specific client needs.

Computer Science Practices:

1. Recognizing and Defining Computational Problems
2. Communicating about Computing
3. Creating Computational Artifacts
4. Testing and Refining Computational Artifacts

Prepared Graduates:

6. Recognize and analyze security concepts.

Grade Level Expectation:

6. Robust computing systems require multiple methods of recovery.

Evidence Outcomes

Students Can:

a. Identify different ways that systems might lose data or functionality.
b. Describe elements of an effective backup system.
c. Compare backup systems for computer users, or a network.
d. List the various backup methodologies (e.g., full, differential), and why one would pick one over the other, or use all.
e. Explain the ways an organization would continue to operate in light of a systems failure.

Academic Context and Connections

Colorado Essential Skills:

1. Analyze the possible sources of a hypothetical system crash. (Entrepreneurial Skills: Critical Thinking/problem solving; Inquiry analysis; Risk taking)
2. Evaluate your personal data backup sources. (Personal Skills: Adaptability/flexibility; Perseverance/adaptability)
3. Denote data privacy measures citizens in Colorado can adopt. (Civic/Interpersonal Skills: Communication)
4. Explain how to prevent your devices from being hacked and offer advice about how to restore data if they have been hacked.(Professional Skills: Information literacy; Use Information and Communications Technologies)

Elaboration on the GLE:

1. The timely and reliable access to data and information services by authorized users, referred to as availability, is ensured through adequate bandwidth, backups and other measures (CSTA 3A-NI-06). Students should understand that an “interruption of service” can come about through disasters, hacking and other deliberate exploitations, power issues and other identifiable problems (e.g., hurricanes). The process of identifying interruptions in services is an important skill for those wanting to work in Information Technology (IT). Backing up a system means that you denote a process in which your computer copies certain data to another safe spot (e.g., another drive, the cloud). Backups are also used in Information Technology (IT) shops in various companies, governmental agencies and educational institutions.

Computer Science Practices:

1. Recognizing and Defining Computational Problems
2. Communicating about Computing
3. Collaborating about Computing

Prepared Graduates:

6. Recognize and analyze security concepts.

Grade Level Expectation:

7. Robust computing systems require data protection.

Evidence Outcomes

Students Can:

a. Identify examples of threats to systems and data.
b. Describe the process by which intruders gain entry into a production system (e.g., reconnaissance).
c. Describe and compare methods to test/validate how well systems and data are protected.
d. Investigate different career pathways relating to systems security.

Academic Context and Connections

Colorado Essential Skills:

1. Explain the importance of penetration testing, and for what purpose a company would employ a “pen-tester.” (Entrepreneurial Skills: Critical Thinking/problem solving; Inquiry analysis; Risk taking)
2. Explain “hardening” of software and data. (Personal Skills: Adaptability/flexibility; Perseverance/adaptability)
3. Describe how hackers use social engineering to gain access to a company’s network. (Civic/Interpersonal Skills: Communication; Character; Global/cultural awareness)
4. Explain what a “SQL injection” is.(Professional Skills: Information literacy; Use Information and Communications Technologies; Career awareness)

Elaboration on the GLE:

1. Security measures may include physical security tokens, two-factor authentication and biometric verification. Potential security problems, such as denial-of-service attacks, ransomware, viruses, worms, spyware and phishing, exemplify why sensitive data should be securely stored and transmitted. Students should systematically evaluate the feasibility of using computational tools to solve given problems or sub-problems, such as long, complex passwords (CSTA 3A-NI-06). See also CSTA 3B-NI-04 and 3B-AP-18). There are numerous, high-paying jobs in the area of system security. For students interested in pursuing such a career, they should understand that there is a high degree of technical understanding required to be successful.

Computer Science Practices:

1. Recognizing and Defining Computational Problems
2. Communicating about Computing
3. Collaborating about Computing

Prepared Graduates:

7. Design and create programs, individually and collaboratively, for a variety of disciplines.

Grade Level Expectation:

1. The creation of a computer program requires a design process.

Evidence Outcomes

Students Can:

a. Analyze and apply a design methodology to identify constraints and requirements of an identified problem.
b. Utilize tools and resources such as pseudocode, flowcharts, wireframes, etc., as part of the design process.
c. Determine and use graphical or text-based languages.
d. Understand and apply core programming concepts.

Academic Context and Connections

Colorado Essential Skills:

1. Apply the design process needed to change a computational artifact over several versions. (Entrepreneurial Skills: Critical thinking/problem solving; Inquiry/analysis)
2. Choose from tools and resources to implement the design process.(Personal Skills: Adaptability/flexibility)
3. Use pseudocode and flowcharts to communicate with a client design options. (Civic/Interpersonal Skills: Communication)
4. Use design resource to effectively manage tasks and be productive. (Professional Skills: Task/time management; Productivity/accountability; Use information/communications technologies)

Elaboration on the GLE:

1. Computer programming requires selection of a design methodology (e.g., engineering, software, human-centered) to identify user needs and requirements. Methodologies provide tools for making important design decisions and help programmers manage the iterative process of software design (CSTA 3A-AP-13).

Computer Science Practices:

1. Recognizing and Defining Computational Problems
2. Developing and Using Abstractions
3. Creating Computational Artifacts
4. Communicating about Computing

Prepared Graduates:

7. Design and create programs, individually and collaboratively, for a variety of disciplines.

Grade Level Expectation:

2. The process of programming involves solving computational problems.

Evidence Outcomes

Students Can:

a. Write code per selected design.
b. Create code comments to communicate to other developers and ensure documentation of code.
c. Use various troubleshooting and debugging techniques to improve code.
d. Create appropriate variables to store and retrieve data.

Academic Context and Connections

Colorado Essential Skills:

1. Create original code that meets specified design requirements. (Entrepreneurial Skills: Creativity/innovation; Inquiry/analysis)
2. Fix code that is not operational. (Personal Skills: Adaptability/flexibility; Perseverance/resilience)
3. Collaborate with others through Pair Programming, commenting code, etc. (no specific language). (Civic/Interpersonal Skills: Communication)

Elaboration on the GLE:

1. Software design is a universal approach that can be used irrespective of programming tools (such as a specific language). Effective design utilizes practices such as commenting to record rationale for specific design decisions (CSTA 3A-AP-21).

Computer Science Practices:

1. Developing and Using Abstractions
2. Creating Computational Artifacts
3. Testing and Refining Computational Artifacts
4. Communicating about Computing

Prepared Graduates:

7. Design and create programs, individually and collaboratively, for a variety of disciplines.

Grade Level Expectation:

3. Collaborative tools, methods and strategies can be used to design, develop and update computational artifacts.

Evidence Outcomes

Students Can:

a. Integrate collaborative strategies to improve programming outputs.
b. Identify and analyze a variety of collaborative tools (e.g., commenting, development repositories) in order to determine the appropriateness for intended use.
c. Identify strategies such as peer reviews to test and refine artifacts in development.
d. Determine when to use standard software tools like APIs, libraries, version control repositories, etc.

Academic Context and Connections

Colorado Essential Skills:

1. Work on a team (product manager, scrum master, analyst, developer, etc.) to improve a computational artifact. (Entrepreneurial Skills: Creativity/innovation; Inquiry/analysis: Personal Skills: Adaptability/flexibility; Perseverance/resilience)
2. Suggest an app for your town/city government that would make your town/neighborhood a nicer place to live. (Civic/Interpersonal Skills: Collaboration/teamwork; Communication; Global/cultural awareness)
3. Use positive constructive feedback to help improve a peer's program. (Professional Skills: Task/time management; Use information/communications technologies; Productivity/accountability)

Elaboration on the GLE:

1. Collaborative strategies such as peer programming and feedback protocols have students optimally revise computational artifacts (e.g., graphical interfaces, program performance, errors) and help foster an inclusive computing culture which produces artifacts that meet the needs of a broad audience (CSTA 3A-AP-22).

Computer Science Practices:

1. Creating Computational Artifacts
2. Fostering an Inclusive Computing Culture
3. Testing and Refining Computational Artifacts
4. Collaborating around Computing
5. Communicating around Computing

Prepared Graduates:

7. Design and create programs, individually and collaboratively, for a variety of disciplines.

Grade Level Expectation:

4. Client-based design requirements and feedback are essential to a quality computational product or service.

Evidence Outcomes

Students Can:

a. Understand and apply principles of client-based design.
b. Guide/advise clients on strategies and solutions best suited for their problem (i.e., type of platform).
c. Construct effective methods for gathering feedback from client.
d. Respond to feedback from clients to improve computing solutions.
e. Create and share product support documentation for potential users.
f. Articulate lessons learned as a result of the design and creation process.

Academic Context and Connections

Colorado Essential Skills:

1. Provide examples of computational artifacts that exemplify client-based and nonclient-based design. (Entrepreneurial Skills: Critical thinking/problem solving; Inquiry/analysis; Risk taking)
2. Ask a friend to give you feedback on your communication skills. Be sure to identify specific ways to improve your communication for your friend. (Personal Skills: Initiative/self-direction; Personal responsibility; Adaptability/flexibility)
3. Document technical information about the software you have produced. (Civic/Interpersonal Skills: Communication; Global/cultural awareness; Character)
4. Produce a computational artifact in accordance with a client's timeline. (Professional Skills: Task/time management; Career awareness; Use of information and communication technologies; Productivity/accountability; Leadership)

Elaboration on the GLE:

1. By allowing students the opportunity to develop programs at the request of a client or identified real-world situation, students are able to have a more authentic learning experience. Students will pursue learning opportunities that are very similar in nature to experiences they will have in a future computer science career. It is important that students follow protocols and frameworks that they would see in the modern workplace to identify problems, develop a programming solution, and bring their artifact to life for review by outside clients (CSTA 3A-AP-19 & 3A-IC-27).

Computer Science Practices:

1. Recognizing and Defining Computational Problems
2. Develop and Using Abstractions
3. Testing and Refining Computational Artifacts
4. Communicating about Computing
5. Collaborating about Computing

Prepared Graduates:

8. Create computational artifacts that consider security from tampering, malicious or otherwise.

Grade Level Expectation:

5. Computing solutions can have impacts (personal, ethical, social, economic and cultural) based on their use.

Evidence Outcomes

Students Can:

a. Investigate and understand privacy, security and protection laws.
b. Articulate the importance of securing personal data information on encrypted storage systems.
c. Identify and analyze current events to ensure the safety, security and well-being of all potential clients and end users.
d. Identify influential computing innovations, and identify the beneficial and harmful effects they have had, or could have, on society, economy and culture.
e. Discuss and explain how diversity of design and issues of accessibility impact a wide-range of users.
f. Demonstrate ways to improve the accessibility of computational technologies and artifacts.

Academic Context and Connections

Colorado Essential Skills:

1. Incorporate security protocols when developing a computational artifact.(Entrepreneurial Skills: Critical thinking/problem solving; Inquiry/analysis; Risk taking)
2. Suggest ways that social media such as Instagram could be made ADA compliant. (Personal Skills: Self-awareness; Adaptability; Initiative; Personal responsibility; Perseverance/resilience)
3. Consider how personal data is vulnerable in both storage and transmission.(Civic/Interpersonal Skills: Collaboration; Communication; Global/cultural awareness; Civic engagement; Character)

Elaboration on the GLE:

1. As students engage in computer programming, it is important for them to be highly aware of the many aspects of cyber and information security. Students need to be aware not only of security loopholes that open their programs up to hacking but also to accidental programming errors or choices that can lead to other security issues as well. Students should do their best to be proactive in their programming but be aware they will need to update code and patch as needed when security vulnerabilities arise. Students should understand the importance of keeping their devices and programs up to date through additional updates and patches but that those as well can lead to other problems. Students want to ensure security is included in their feedback cycle for developed solutions (CSTA 3A-NI-06).

Computer Science Practices:

1. Recognizing and Defining Computational Problems
2. Testing and Refining Computational Artifacts
3. Communicating about Computing

COMPUTER SCIENCE
High School, Standard 3. Computer Programming

COLORADO Department of Education

Prepared Graduates:

8. Create computational artifacts that consider security from tampering, malicious or otherwise.

Grade Level Expectation:

6. Security and software licensing can present constraints and restrictions in computational design and development.

Evidence Outcomes

Students Can:

a. Describe how software licensing influences program development.
b. Investigate and develop solutions that discourage online software piracy.
c. Explore and integrate security measures such as encryption, authentication and verification strategies to secure developed computer programs.
d. Research and abide by intellectual property laws and patents.

Academic Context and Connections

Colorado Essential Skills:

1. Learn about the steps required for protecting intellectual rights of your computational artifact. (Entrepreneurial Skills: Critical thinking/problem solving; Inquiry and analysis; Risk taking)
2. Analyze licensing agreements from a software vendor. (Civic/Interpersonal Skills: Collaboration; Communication; Global/cultural awareness; Civic engagement; Character)
3. Evaluate the benefits of open-source and proprietary software to the developer. (Professional Skills: Career awareness; Information Literacy; Use information/communications technologies; Self-advocacy)

Elaboration on the GLE:

1. After finishing a computer program, students should consider how they would potentially distribute their product. Whether they determine to sell it at a price on an app store or distribute it for free, a license of some sort is required and a process for which consumers can access the program. Alternatively, students need to be mindful that pirating occurs and should think about ways they can secure their programs to not be unlawfully distributed such as licensing codes, attachment to connected services, distribute software, etc. Students need to be aware of laws and patents that govern/protect intellectual property CSTA 3A-AP-20 & 3A-IC-28).

Computer Science Practices:

1. Recognizing and Defining Computational Problems
2. Testing and Refining a Computational Artifact
3. Communicating about Computing