Appendix A

ISTE Standards for Students

1. Empowered Learner

Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences. Students:

a. articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes.
b. build networks and customize their learning environments in ways that support the learning process.
c. use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.
d. understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2. Digital Citizen

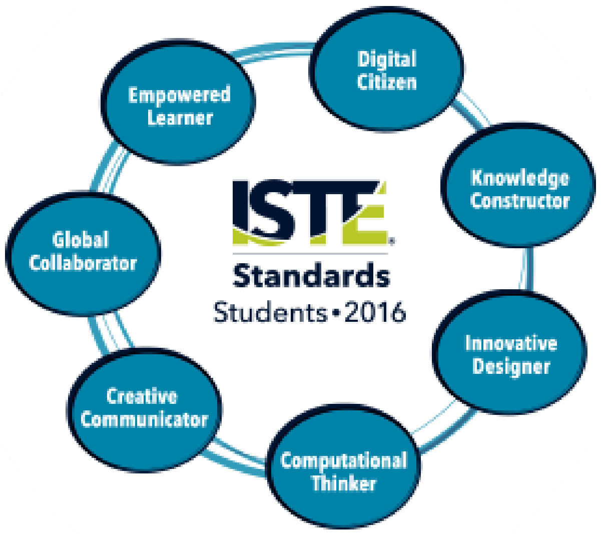
Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical. Students:

a. cultivate and manage their digital identity and reputation and are aware of the permanence of their actions in the digital world.
b. engage in positive, safe, legal and ethical behavior when using technology, including social interactions online or when using networked devices.
c. demonstrate an understanding of and respect for the rights and obligations of using and sharing intellectual property.
d. manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online.

3. Knowledge Constructor

Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others. Students:

a. plan and employ effective research strategies to locate information and other resources for their intellectual or creative pursuits.
b. evaluate the accuracy, perspective, credibility and relevance of information, media, data or other resources.
c. curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions.
d. build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.



4. Innovative Designer

Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions. Students:

a. know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.
b. select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.
c. develop, test and refine prototypes as part of a cyclical design process.
d. exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.

5. Computational Thinker

Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions. Students:

a. formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.
b. collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.
c. break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
d. understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

6. Creative Communicator

Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals. Students:

a. choose the appropriate platforms and tools for meeting the desired objectives of their creation or communication.
b. create original works or responsibly repurpose or remix digital resources into new creations.
c. communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.
d. publish or present content that customizes the message and medium for their intended audiences.

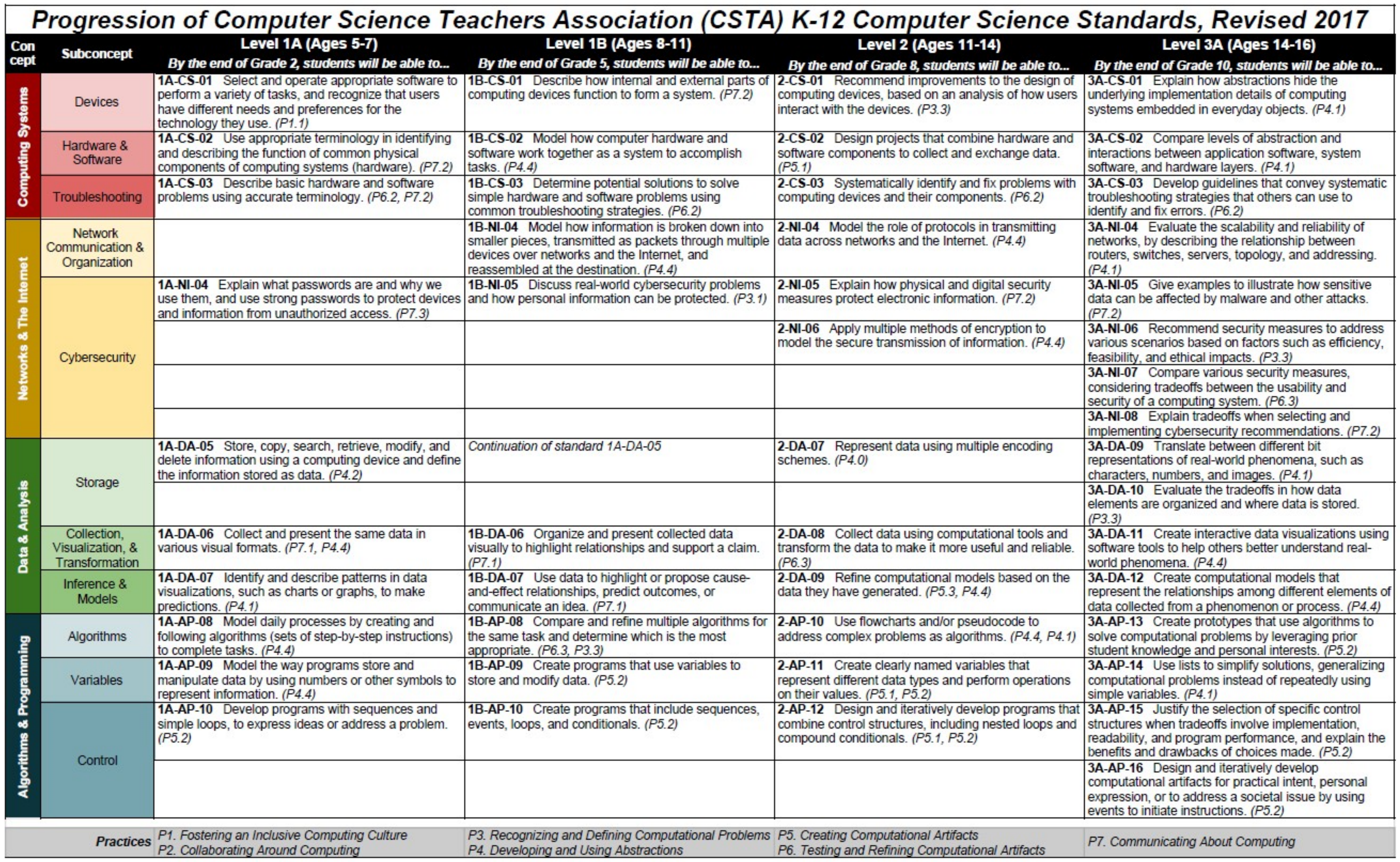
7. Global Collaborator

Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally. Students:

a. use digital tools to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning.
b. use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.
c. contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.
d. explore local and global issues and use collaborative technologies to work with others to investigate solutions

Appendix B

Computer Science Teacher Association (CSTA) K-12 Computer Science Standards



Progression of Computer Science Teachers Association (CSTA) K-12 Computer Science Standards, Revised 2017

Concept	Subconcept	Level 1A (Ages 5-7) *By the end of Grade 2, students will be able to…*	Level 1B (Ages 8-11) *By the end of Grade 5, students will be able to…*	Level 2 (Ages 11-14) *By the end of Grade 8, students will be able to…*	Level 3A (Ages 14-16) *By the end of Grade 10, students will be able to…*
Computing Systems	Devices	**1A-CS-01** Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use. *(P1.1)*	**1B-CS-01** Describe how internal and external parts of computing devices function to form a system. *(P7.2)*	**2-CS-01** Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. *(P3.3)*	**3A-CS-01** Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. *(P4.1)*
	Hardware & Software	**1A-CS-02** Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). *(P7.2)*	**1B-CS-02** Model how computer hardware and software work together as a system to accomplish tasks. *(P4.4)*	**2-CS-02** Design projects that combine hardware and software components to collect and exchange data. *(P5.1)*	**3A-CS-02** Compare levels of abstraction and interactions between application software, system software, and hardware layers. *(P4.1)*
	Troubleshooting	**1A-CS-03** Describe basic hardware and software problems using accurate terminology. *(P6.2, P7.2)*	**1B-CS-03** Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. *(P6.2)*	**2-CS-03** Systematically identify and fix problems with computing devices and their components. *(P6.2)*	**3A-CS-03** Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. *(P6.2)*
Networks & The Internet	Network Communication & Organization		**1B-NI-04** Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. *(P4.4)*	**2-NI-04** Model the role of protocols in transmitting data across networks and the Internet. *(P4.4)*	**3A-NI-04** Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. *(P4.1)*
	Cybersecurity	**1A-NI-04** Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access. *(P7.3)*	**1B-NI-05** Discuss real-world cybersecurity problems and how personal information can be protected. *(P3.1)*	**2-NI-05** Explain how physical and digital security measures protect electronic information. *(P7.2)*	**3A-NI-05** Give examples to illustrate how sensitive data can be affected by malware and other attacks. *(P7.2)*
				2-NI-06 Apply multiple methods of encryption to model the secure transmission of information. *(P4.4)*	**3A-NI-06** Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. *(P3.3)*
					3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system. *(P6.3)*
					3A-NI-08 Explain tradeoffs when selecting and implementing cybersecurity recommendations. *(P7.2)*
Data & Analysis	Storage	**1A-DA-05** Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. *(P4.2)*	*Continuation of standard 1A-DA-05*	**2-DA-07** Represent data using multiple encoding schemes. *(P4.0)*	**3A-DA-09** Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. *(P4.1)*
					3A-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored. *(P3.3)*
	Collection, Visualization, & Transformation	**1A-DA-06** Collect and present the same data in various visual formats. *(P7.1, P4.4)*	**1B-DA-06** Organize and present collected data visually to highlight relationships and support a claim. *(P7.1)*	**2-DA-08** Collect data using computational tools and transform the data to make it more useful and reliable. *(P6.3)*	**3A-DA-11** Create interactive data visualizations using software tools to help others better understand real-world phenomena. *(P4.4)*
	Inference & Models	**1A-DA-07** Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. *(P4.1)*	**1B-DA-07** Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea. *(P7.1)*	**2-DA-09** Refine computational models based on the data they have generated. *(P5.3, P4.4)*	**3A-DA-12** Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. *(P4.4)*
Algorithms & Programming	Algorithms	**1A-AP-08** Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. *(P4.4)*	**1B-AP-08** Compare and refine multiple algorithms for the same task and determine which is the most appropriate. *(P6.3, P3.3)*	**2-AP-10** Use flowcharts and/or pseudocode to address complex problems as algorithms. *(P4.4, P4.1)*	**3A-AP-13** Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. *(P5.2)*
	Variables	**1A-AP-09** Model the way programs store and manipulate data by using numbers or other symbols to represent information. *(P4.4)*	**1B-AP-09** Create programs that use variables to store and modify data. *(P5.2)*	**2-AP-11** Create clearly named variables that represent different data types and perform operations on their values. *(P5.1, P5.2)*	**3A-AP-14** Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. *(P4.1)*
	Control	**1A-AP-10** Develop programs with sequences and simple loops, to express ideas or address a problem. *(P5.2)*	**1B-AP-10** Create programs that include sequences, events, loops, and conditionals. *(P5.2)*	**2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. *(P5.1, P5.2)*	**3A-AP-15** Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. *(P5.2)*
					3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. *(P5.2)*
	Practices	*P1. Fostering an Inclusive Computing Culture* *P2. Collaborating Around Computing*	*P3. Recognizing and Defining Computational Problems* *P4. Developing and Using Abstractions*	*P5. Creating Computational Artifacts* *P6. Testing and Refining Computational Artifacts*	*P7. Communicating About Computing*

Progression of Computer Science Teachers Association (CSTA) K-12 Computer Science Standards, Revised 2017

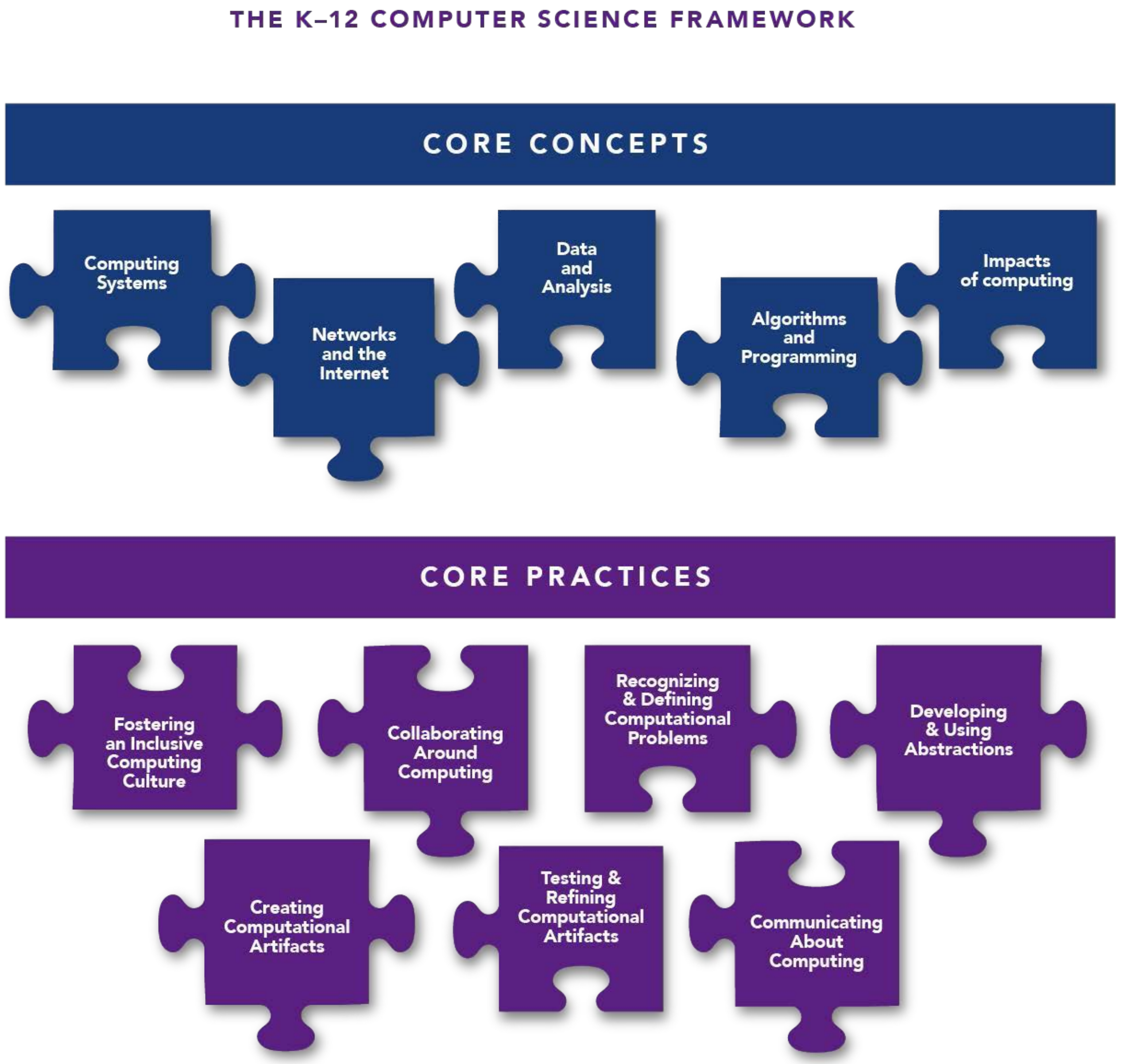
Concept	Subconcept	Level 1A (Ages 5-7) *By the end of Grade 2, students will be able to…*	Level 1B (Ages 8-11) *By the end of Grade 5, students will be able to…*	Level 2 (Ages 11-14) *By the end of Grade 8, students will be able to…*	Level 3A (Ages 14-16) *By the end of Grade 10, students will be able to…*
Algorithms & Programming *(continued)*	Modularity	**1A-AP-11** Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. *(P3.2)*	**1B-AP-11** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. *(P3.2)*	**2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. *(P3.2)*	**3A-AP-17** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. *(P3.2)*
			1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. *(P5.3)*	**2-AP-14** Create procedures with parameters to organize code and make it easier to reuse. *(P4.1, P4.3)*	**3A-AP-18** Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. *(P5.2)*
	Program Development	**1A-AP-12** Develop plans that describe a program's sequence of events, goals, and expected outcomes. *(P5.1, P7.2)*	**1B-AP-13** Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. *(P1.1, P5.1)*	**2-AP-15** Seek and incorporate feedback from team members and users to refine a solution that meets user needs. *(P2.3, P1.1)*	**3A-AP-19** Systematically design and develop programs for broad audiences by incorporating feedback from users. *(P5.1)*
		1A-AP-13 Give attribution when using the ideas and creations of others while developing programs. *(P7.3)*	**1B-AP-14** Observe intellectual property rights and give appropriate attribution when creating or remixing programs. *(P7.3)*	**2-AP-16** Incorporate existing code, media, and libraries into original programs, and give attribution. *(P4.2, P5.2, P7.3)*	**3A-AP-20** Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. *(P7.3)*
		1A-AP-14 Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops. *(P6.2)*	**1B-AP-15** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. *(P6.1, P6.2)*	**2-AP-17** Systematically test and refine programs using a range of test cases. *(P6.1)*	**3A-AP-21** Evaluate and refine computational artifacts to make them more usable and accessible. *(P6.3)*
			1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. *(P2.2)*	**2-AP-18** Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. *(P2.2)*	**3A-AP-22** Design and develop computational artifacts working in team roles using collaborative tools. *(P2.4)*
		1A-AP-15 Using correct terminology, describe steps taken and choices made during the iterative process of program development. *(P7.2)*	**1B-AP-17** Describe choices made during program development using code comments, presentations, and demonstrations. *(P7.2)*	**2-AP-19** Document programs in order to make them easier to follow, test, and debug. *(P7.2)*	**3A-AP-23** Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. *(P7.2)*
Impacts of Computing	Culture	**1A-IC-16** Compare how people live and work before and after the implementation or adoption of new computing technology. *(P7.0)*	**1B-IC-18** Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices. *(P7.1)*	**2-IC-20** Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. *(P7.2)*	**3A-IC-24** Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. *(P1.2)*
			1B-IC-19 Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. *(P1.2)*	**2-IC-21** Discuss issues of bias and accessibility in the design of existing technologies. *(P1.2)*	**3A-IC-25** Test and refine computational artifacts to reduce bias and equity deficits. *(P1.2)*
					3A-IC-26 Demonstrate ways a given algorithm applies to problems across disciplines. *(P3.1)*
	Social Interactions	**1A-IC-17** Work respectfully and responsibly with others online. *(P2.1)*	**1B-IC-20** Seek diverse perspectives for the purpose of improving computational artifacts. *(P1.1)*	**2-IC-22** Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact. *(P2.4, P5.2)*	**3A-IC-27** Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. *(P2.4)*
	Safety, Law, & Ethics		**1B-IC-21** Use public domain or creative commons media, and refrain from copying or using material created by others without permission. *(P7.3)*		**3A-IC-28** Explain the beneficial and harmful effects that intellectual property laws can have on innovation. *(P7.3)*
		1A-IC-18 Keep login information private, and log off of devices appropriately. *(P7.3)*		**2-IC-23** Describe tradeoffs between allowing information to be public and keeping information private and secure. *(P7.2)*	**3A-IC-29** Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. *(P7.2)*
					3A-IC-30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics. *(P7.3)*
	Practices	*P1. Fostering an Inclusive Computing Culture* *P2. Collaborating Around Computing*	*P3. Recognizing and Defining Computational Problems* *P4. Developing and Using Abstractions*	*P5. Creating Computational Artifacts* *P6. Testing and Refining Computational Artifacts*	*P7. Communicating About Computing*

Suggested Citation: Computer Science Teachers Association (2017). CSTA K-12 Computer Science Standards, Revised 2017. Retrieved from http://www.csteachers.org/standards.

The K–12 Computer Science Framework, led by the Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative in partnership with states and districts, informed the development of this work. View the framework at http://k12cs.org.

K12 Computer Science Framework

The framework is a high-level guide describing Computer Science (CS) for all students. The framework will empower students to be informed citizens who can critically engage in CS-related discussions; develop as learners, users, and creators of CS knowledge and artifacts; better understand the role of computing in the world around them; and learn, perform, and express themselves in other subjects and interests. At its center are core concepts and practices. View the framework at https://k12cs.org/navigating-the-practices/.


THE K–12 COMPUTER SCIENCE FRAMEWORK
CORE CONCEPTS
Computing Systems
Networks and the Internet
Data and Analysis
Algorithms and Programming
Impacts of computing
CORE PRACTICES
Fostering an Inclusive Computing Culture
Collaborating Around Computing
Recognizing & Defining Computational Problems
Developing & Using Abstractions
Creating Computational Artifacts
Testing & Refining Computational Artifacts
Communicating About Computing

LITERACY STANDARDS FOR GRADES 6-12:

HISTORY/SOCIAL STUDIES, SCIENCE, AND TECHNICAL SUBJECTS

College and Career Readiness Anchor Standards for Reading

The Grades 6-12 standards on the following pages define what students should understand and be able to do by the end of each grade span. They correspond to the College and Career Readiness (CCR) anchor standards below by number. The CCR and grade-specific standards are necessary complements—the former providing broad standards, the latter providing additional specificity—that together define the skills and understandings that all students must demonstrate.

Key Ideas and Details

1. Read closely to determine what the text says explicitly and to make logical inferences from it; cite specific textual evidence when writing or speaking to support conclusions drawn from the text.
2. Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.
3. Analyze how and why individuals, events, or ideas develop and interact over the course of a text.

Craft and Structure

4. Interpret words and phrases as they are used in a text, including determining technical, connotative, and figurative meanings, and analyze how specific word choices shape meaning or tone.
5. Analyze the structure of texts, including how specific sentences, paragraphs, and larger portions of the text (e.g., a section, chapter, scene, or stanza) relate to each other and the whole.
6. Assess how point of view or purpose shapes the content and style of a text.

Integration of Knowledge and Ideas

7. Integrate and evaluate content presented in diverse formats and media, including visually and quantitatively, as well as in words. *
8. Delineate and evaluate the argument and specific claims in a text, including the validity of the reasoning as well as the relevance and sufficiency of the evidence.
9. Analyze how two or more texts address similar themes or topics in order to build knowledge or to compare the approaches the authors take.

Range of Reading and Level of Text Complexity

10. Read and comprehend complex literary and informational texts independently and proficiently.

*See College and Career Readiness Anchor Standards for Writing, “Research to Build and Present Knowledge,” on page 60 for additional standards relevant to gathering, assessing, and applying information from print and digital sources.

College and Career Readiness Anchor Standards for Writing

The Grades 6-12 standards on the following pages define what students should understand and be able to do by the end of each grade span. They correspond to the College and Career Readiness (CCR) anchor standards below by number. The CCR and grade-specific standards are necessary complements—the former providing broad standards, the latter providing additional specificity—that together define the skills and understandings that all students must demonstrate.

Text Types and Purposes*

1. Write arguments to support claims in an analysis of substantive topics or texts using valid reasoning and relevant and sufficient evidence.
2. Write informative/explanatory texts to examine and convey complex ideas and information clearly and accurately through the effective selection, organization, and analysis of content.
3. Write narratives to develop real or imagined experiences or events using effective technique, well-chosen details, and well-structured event sequences.

Production and Distribution of Writing

4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
5. Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach.
6. Use technology, including the Internet, to produce and publish writing and to interact and collaborate with others.

Research to Build and Present Knowledge

7. Conduct short as well as more sustained research projects based on focused questions, demonstrating understanding of the subject under investigation.
8. Gather relevant information from multiple print and digital sources, assess the credibility and accuracy of each source, and integrate the information while avoiding plagiarism.
9. Draw evidence from literary or informational texts to support analysis, reflection, and research.

Range of Writing

10. Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of tasks, purposes, and audiences.

*These broad types of writing include many subgenres

Bibliography

Alabama Course of Study: Digital Literacy and Computer Science. Alabama State Department of Education, 2018. www.alsde.edu/sec/sct/COS/Final%202018%20Digital%20Literacy%20and%20Computer%20 Science%20COS[5206].pdf

"CS Discoveries 2018: Vocab." *Code.org*, 2017, curriculum.code.org/csd/vocab/. Accessed 3 January 2019.

"CSTA K-12 Computer Science Standards." *Computer Science Teachers Association*, 2017, www.csteachers.org/standards. Accessed 4 January 2019.

"Glossary: CS Fundamentals." *Code.org*, 2017, code.org/curriculum/docs/k- 5/glossary. Accessed 4 January 2019.*Interactive Design Foundation.* The Interactive Design Foundation, 2016, www.interaction-design.org/literature/article/5-stages- in-the-design-thinking-process. Accessed 3 January 2019.

"ISTE Standards for Students." *ISTE (International Society for Technology in Education)*, 2016, www.iste.org/standards/for-students. Accessed 4 January 2019.

K-12 Computer Science Framework. Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative, k12cs.org. Accessed 4 January 2019.

Merriam-Webster. Merriam Webster, 2019. www.merriam-webster.com/. Accessed 3 January 2019.

"Pseudocode." *Whatis.com*, 2019, whatis.techtarget.com/search/query?q=pseudocode. Accessed 4 January 2019.

Glossary

Abstraction – The process of withdrawing or removing details to highlight essential properties (e.g., the color of a pixel represented at one level as three bytes (red, green, and blue), or at a lower level as a sequence of bits). The act of representing essential features without including the background details or explanations.

Acceptable Use/Usage Policy (AUP) – A document defining constraints and practices that a user must agree to for access to an organization's network and/or the Internet. Many organizations require that employees or students sign an acceptable use policy before being granted a access to the network.

Addressing – Unique identification of components on a network.

Algorithm – A list of steps to finish a task; a set of instructions that can be performed with or without a computer.

Application Programming Interface (API) – Code that allows two software programs communicate with one another; an external software library that provides a collection of features (implemented as functions or methods) that offer reusable functionality in a program (e.g., OpenGL is an external library that provides an API that can be used for programming of computer graphics).

Array – An indexable collection of values (e.g., a row, a column, or a collection of integers representing student grades).

Artificial Intelligence (AI) – The capability of a machine to imitate intelligent human behavior.

ASCII (American Standard Code for Information Interchange) – A character set for representing numbers, punctuation, and letters in the English alphabet (e.g., the number 65 corresponds to the letter "A").

Assistive technologies – Any item, piece of equipment, or product system that is used to increase, maintain, or improve functional capabilities of a person with a disability.

Attribution – The ascribing of a work (as of literature or art) to a particular author or artist.

Automation – The use of a combination of mechanical and computer-based control systems to perform actions without the need of human oversight.

Bandwidth – The bit-rate measure of the transmission capacity over a network communication system; or, the carrying capacity of a channel or the data transfer speed of that channel.

Binary – A discrete numbering system that can represent information using only two options (0 or 1, on or off, yes or no, true or false) to allow for digital processing.

Biometric – The process by which a person's unique physical and other traits are detected and recorded by an electronic device or system as a means of confirming identity.

Bit – A contraction of "**B**inary dig**IT**." A bit is the single unit of information in a computer, typically represented as a 0 or 1.

Block-based programming language – Any predefined code that lets users create programs by manipulating "blocks" or graphical programming elements, rather than writing code using text with specific syntax rules, sometimes called visual coding, drag-and-drop programming, or graphical programming blocks.

Boolean – A variable data type or expression that can be set to either true or false.

Bug – An error in a program that prevents the program from running as expected.

Byte – A contraction of "**B**inar**Y** **TE**rm." A group of bits, usually eight, processed as a single unit of data.

Censorship – The act of examining media for the purpose of suppressing parts deemed objectionable on moral, political, military, or other grounds.

Cipher – A method of transforming a text in order to conceal its meaning; or, a coded message that requires a key to decode. The cipher cannot be decoded without the key.

Citation – The attribution of a reference to a book, paper, author, or other item in fixed, tangible form, especially in a scholarly work.

Client-server computing – A distributed network architecture with shared responsibilities between servers (computers that provides resources and services) and clients (those that request the resources/services).

Cloud-based computing – Applications, services, or resources made available to users on demand via the Internet from a remote computing provider's server.

Code – One or more command(s) or algorithm(s) designed to be carried out by a computer using a programming language. See also: **Program**

Comment – A note in the source code of a computer program that helps explain the code to those who read it.

Command – An instruction given by a user telling a computer to do something, such as run a single program or a group of linked programs.

Compression – The process of reducing the size of a computational artifact (e.g., a file) using a digital tool which implements an algorithm that recognizes repetitive data and removes redundancy across the artifact.

Computer science – The study of computers and algorithmic processes including their principles, hardware and software designs, applications, networks, and impact on society.

Computer system – A collection of one or more computers or computing devices, together with their hardware and software, integrated for the purpose of accomplishing shared tasks. Although a computer system can be limited to a single computer or computing device, it more commonly refers to a collection of multiple, connected computers, computing devices, and hardware.

Computational artifact – Something created by a human using a computer which can be, but is not limited to, a program, an image, an audio clip, a video, a presentation, or a Web page file. The computational artifact could solve a problem, show creative expression, or provide a viewer with new insight or knowledge.

Computational thinking – a problem-solving process that includes (but is not limited to) the following characteristics: Formulating problems in a way that enables us to use a computer and other tools to help solve them; Logically organizing and analyzing data; Representing data through abstractions such as models and simulations; Automating solutions through algorithmic thinking (a series of ordered steps); Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources; Generalizing and transferring this problem solving process to a wide variety of problems.

Conditionals/Compound conditionals ("If" statements) – Statements that run only when certain conditions exist. Often called a selection or "if" statement in a programming language, represented as an expression that evaluates to a Boolean value.

Control – The power to direct the course of actions. In programming, the use of elements of programming code to direct which actions take place and the order in which they take place. A programming (code) structure that implements control. Selection ("if" statements) and loops are examples of control structures.

Cookies – A small file or part of a file stored on a user's computer, created and subsequently read by a website server, and containing personal information (such as a user identification code, customized preferences, or a record of pages visited).

Copyright – A legal right created by the law of a country that grants the creator of an original work exclusive right for its use and distribution.

Creative Commons – A collection of public copyright licenses that enable the free distribution of an otherwise copyrighted work, used when an author wants to give people the right to share, use, and build upon a work that they have created.

Cyberbullying – Using electronic communication to intimidate, humiliate, or threaten another person.

Cybersecurity – The protection against access to, or alteration of, computing resources, through the use of technology, processes, and training.

Data – Raw facts that are collected and used for reference or analysis. Data can be digital or nondigital and can be in many forms, including numbers, text, show of hands, images, sounds, or video. Information used as a basis for reasoning, discussion, or calculation.

Data mining – The process of traversing through large data sets to identify patterns and establish relationships in order to solve problems through data analysis.

Database – A collection of data organized for search and retrieval.

Debug – To find and fix errors in an algorithm or program.

Decomposition (Decomposing) – The process of separating a whole into related parts or elements.

Decryption – The process of taking encoded or encrypted text or other data and converting it back into text (often called plaintext) that a human or computer can read and understand.

Design thinking – A methodology used to solve complex problems and find desirable solutions using logic, imagination, intuition, and systemic reasoning to explore possibilities of what could be and to create desired outcomes that benefit the end user.

Digital identity – Information on an entity used by computer systems to represent an external person, organization, application, or device.
Digital footprint – The collected information about an individual across multiple digital sources.
Digital globalization – The unrestricted flow of electronic data and products.
Digital literacy – The ability to use information and communication technologies to find, evaluate, create, and communicate information. Digital Literacy requires both cognitive and technical skills.
Digital permanence – The history and development of digital storage techniques, specifically quantifying the expected lifetime of data stored on various digital media and the factors which influence the perpetuity of digital data.
Domain Name System (DNS) server – An Internet service that translates a domain name to the correct IP address (e.g., amazon.com to 72.21.215.90).
Emerging technology – A new technology that is currently being developed, or will be developed in the near future.
Encryption – The process of converting information or data into a text (often called ciphertext) that is not readable by a human.
Equitable access – Robust and reliable access to current and emerging technologies and digital resources, with connectivity for all.
Extraction – The process of retrieving relevant information from data sources (like a database) in a specific pattern.
File Transfer Protocol (FTP) – A network protocol that provides the capability to upload and download data within a network.
Flaming – The use of digital emotional abuse to invoke certain emotions and responses such as rage, sadness, humiliation, self-doubt and more.
Flowchart – A diagram of the sequence of movements or actions of people or things involved in a complex system or activity.
Function – A named piece of code that can be called over and over again, sometimes called *procedures* or *methods*; a segment of code that includes the steps performed in a specified process.
Geolocation – The process or technique of identifying the physical location of a person or device by means of digital information processed via the Internet.
Hacking – Using a computer to create or explore some new idea; more commonly, to gain illegal access to a computer.
Hexadecimal – A number system with a base of 16, compared to the decimal number system (base 10) or binary number system (base 2), which uses the letters A through F to represent 10 through 15 in decimal (e.g., hexadecimal 10 is equal to 16 in decimal).
Hierarchical classification – A classification system where entries are arranged based on some order of rank structure.
Hyperlink – A link from an HTML file to another location or file, typically activated by clicking on a highlighted word or image on the screen.
Hypertext Transfer Protocol (HTTP) – An Internet protocol that controls the transfer of web data over the Internet.
Hypertext Transfer Protocol Secure (HTTPS) – An Internet protocol used to transmit web data securely via encryption.
Information – Data that has been processed into a useful form.
Ideation – The capacity for forming ideas.
Infographic – A visual image or group of images combined with simple text to represent complex data in a simplified way.
Initialize – To set something (such as a computer program counter) to a starting position, value, or configuration.
Input – A device or component that allows information to be received by a computer.
Intellectual property – A work or invention that is the result of creativity, such as a piece of writing or a design, which one owns and for which one may apply for a patent, copyright, or trademark.

Internet of Things (IoT) – The ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between objects and other Internet-enabled devices and systems.

Internet Protocol (IP) address – A collection of numbers (often four sets of numbers separated by a period, such as 72.21.215.90) that is used to uniquely identify each device connected to a computer network, such as the Internet.

Iteration (Loop) – A repetitive action or command typically created with programming loops. Loop is the action of doing something over and over again.

Loop - See Iteration; Control

Keyword – Main or significant term used to search the Internet for content; also used to represent the words that comprise a computer programming language.

Malware – Software designed to negatively impact a computer's normal functioning.

Metadata – Information that describes other information, such as descriptive data, organizational descriptions, and procedural information regarding the creation and technical specifications of the information. For example, an image file may contain metadata regarding the time and place of the picture creation as well as technical specifications about the camera used.

Multimedia – Using, involving, or encompassing several media (e.g., images, sound, video).

Netiquette – (Inter**NET** et**IQUETTE**). The correct or acceptable way of communicating on the Internet.

Net neutrality – The principle that Internet service providers should enable access to all content and applications regardless of the sources, and without favoring or blocking particular products or websites.

Network – An interconnected system of computers, peripherals, terminals, and databases connected by communication lines.

Operating system – Software that controls the operation of a computer and directs the processing of programs.

Output – Any device or component that transmits information from a computer.

Packet – A unit of data routed between an origin and a destination on a network.

Packet sniffing – Intercepting and reading the information within network packets that are sent as plaintext (not encrypted).

Parameter – a numerical or other measurable factor forming one of a set that defines a system or sets the conditions of its operation.

Password – A string of characters used for authentication to prove identity in order to approve access.

Peer-to-peer computing – Two or more computers with similar access privileges and responsibilities that communicate directly in order to share resources and services, rather than going through an intermediate server.

Prototype –A first full-scale and usually functional form of a new design.

Perseverance – Continued effort to do or achieve something despite difficulties, failure, or opposition.

Personal security – Actions which reduce the risk of threats and protect the user from hurtful disruptions in the patterns of daily life.

Phishing – The fraudulent practice of sending emails purporting to be from reputable companies in order to induce individuals to reveal personal information, such as passwords and credit card numbers.

Pixel – A contraction of "**PIC**ture **EL**ement." Any of the small, discrete elements that together constitute an image (as on a television or computer screen).

Portability – The ability of a user to export data, information, or software entered into or created by a software application or computing platform so it may be used in other applications or platforms.

Predictive modeling – A process that uses data mining and probability to forecast outcomes.

Problem domain – The area of expertise or application that is being explored to the exclusion of all other topics.

Problem-solving process – Problem definition > Problem Analysis > Generate possible solutions > Analyze solutions > Selecting appropriate solution(s) > Evaluate solution.

Program – An algorithm that has been coded into a form that can be run by a machine.

Programming (Coding) – The art of envisioning, designing, and implementing a computer program using some computational language.

Programming language – A vocabulary and set of syntax rules for instructing a computer or computing device to perform specific tasks.

Pseudocode – A detailed yet readable description of what a computer program or algorithm must do, expressed in a formally-styled natural language rather than programming language. (See Conditionals)

Queue – A data structure that consists of a list of records arranged so that records are added at one end and removed from the other. Sometimes described as First In First Out (FIFO).

Ransomware – A type of malware designed to block access to data until a specific demand (usually financial) is met.

Recurring standards – Key practices or concepts that appear at grade levels along the K - 12 continuum with progressive complexity. Rather than repeating these standards at multiple grade levels in this document, they are included on the Digital Literacy and Computer Science Course of Study Recurring Standards pages.

Router – Device or software that determines the path that data packets travel from source to destination.

Scalability – The ability of a process to maintain functionality and integrity as the scope and size of the process increases.

Selection – Using a Boolean condition to determine which of two parts of an algorithm is used.

Sequencing – Performing tasks in logical order.

Server – Computers on a network that provide access to resources and/or services to clients.

Simple Mail Transfer Protocol (SMTP) – An Internet standard that defines the protocol for sending electronic mail.

Simulation –The production of a digital model of something, especially for the purpose of study.

Social engineering – The management and manipulation of human beings in accordance with their place and function in society.

Software piracy – The illegal copying, distribution, or use of software.

Spoofing – To deceive or hoax; often used to indicate identity deception of email or network addresses.

Stack – A memory or a section of memory in a computer for temporary storage in which the last item stored is the first retrieved. Sometimes described as Last In First Out (LIFO).

Switch – A high-speed device that receives incoming data packets and redirects them to their destination on a local area network (LAN).

Syntax – An arrangement of items that abides by a prescribed set of rules.

System – A set of connected things or parts forming a complex whole, in particular; a set of principles or procedures according to which something is done; an organized scheme or method systems.

Table – A collection of arrays, which create rows and columns for ready reference; a database is represented by a collection of tables that are used to store information.

Topology – A description of the manner in which various network entities (computers, routers, printers, etc.) are arranged and connected.

Two-factor authentication – A security mechanism that requires two types of credentials for authentication and is designed to provide an additional layer of validation, minimizing security breaches (e.g., a password, and a text message confirmation).

Unplugged activity – An activity that teaches the fundamentals of computer science without digital tools, using physical implements such as card activities, strings, crayons, or puzzles.

Variable – An element, feature, or factor that is liable to change; in a programming language, a symbolic representation of some state or property of the program.

Version control – The consistent management of historical changes made to a digital artifact by collaborators.

Virus – Programming code that has been written to cause corruption of data on a computer; often attached to an executable file that spreads from one file to another once the program is executed.

Website – A collection of interlinked web pages on the World Wide Web.