To provide feedback for these **draft** standards go to https://www.surveymonkey.com/r/2T78756

CSFS.PCGEI: Personal, Community, Global, and Ethical Impacts

Standard	K-2 Benchmarks	3-5 Benchmarks	6-8 Benchmarks	9-12 Benchmarks
CSFS.PCGE.1 Responsible use of technology and information.	CSFS.K2.PCGE.1.1. Demonstrate proper care for electronic devices (e.g., handling devices carefully, logging off or shutting down correctly, and keeping devices away from water/food).	CSFS.35.PCGE.1.1. Identify appropriate and inappropriate uses of technology when posting to social media, sending e-mail, and browsing the Internet.	CSFS.68.PCGE.1.1. Recognize and discuss legal and ethical behaviors when using information and technology and discuss the consequences of misuse.	CSFS.912.PCGE.1.1. Compare and contrast appropriate and inappropriate social networking behaviors.
	CSFS.K2.PCGE.1.2. Describe the attributes of a good digital citizen: one who protects private information, balances time online, reports cyberbullying, and recognizes inappropriate content/contact.	CSFS.35.PCGE.1.2. Discuss responsible uses of modern communication media and devices	CSFS.68.PCGE.1.2. Describe and use safe and appropriate practices when participating in online communities (e.g., discussion groups, blogs, and social networking sites).	CSFS.912.PCGE.1.2. Describe and demonstrate ethical and responsible use of modern communication media and devices.
	CSFS.K2.PCGE.1.3 Identify safe and unsafe examples of online communications.	CSFS.35.PCGE.1.3. Explain the proper use and operation of security technologies (e.g. passwords, virus	CSFS.68.PCGE.1.3. Evaluate the proper use and operation of security technologies (e.g. passwords, virus protection	CSFS.912.PCGE.1.3. Evaluate the impacts of irresponsible use of information (e.g., plagiarism

	CSFS.K2.PCGE.1.4. Explain that a password helps protect the privacy of information.	protection software, spam filters, pop-up blockers, cookies).	software, spam filters, pop-up blockers, cookies).	and falsification of data) on collaborative projects. CSFS.912.PCGE.1.4. Explain the principles of cryptography by examining encryption, digital signatures, and authentication methods (e.g. explain why and how certificates are used with https for authentication and encryption). CSFS.912.PCGE.1.5. Implement an encryption, digital signature, or authentication method. CSFS.912.PCGE.1.6. Describe computer security vulnerabilities and methods of attack, and evaluate their social and economic impact on computer systems and people.

CSFS.PCGE.2. The impact of computing resources on personal life and global society	CSFS.K2.PCGE.2.1. Identify and describe how people use many types of technologies in their daily work and personal lives.	CSFS.35.PCGE.2.1. Explain how computers and computing devices are used to communicate with others on a daily basis.	CSFS.68.PCGE.2.1. Analyze the positive and negative impacts of computing, social networking and web technologies on human culture.	CSFS.912.PCGE.2.1. Describe how the Internet facilitates global communication.
	CSFS.K2.PCGE.2.2. Use developmentally appropriate terminology when communicating about technology.	CSFS.35.PCGE.2.2. Describe the computers and computing devices used to complete academic tasks such as completing daily assignments and doing research.	CSFS.68.PCGE.2.2. Explain the possible consequences of cyberbullying and inappropriate use of social media on personal life and society.	CSFS.912.PCGE.2.2. Understand and identify ways to use technology to support lifelong learning.
		CSFS.35.PCGE.2.3. Describe types of cyberbullying and explain what actions should be taken if students are either victims or witnesses of these behaviors.	CSFS.68.PCGE.2.3. Discuss the influence of access to information technologies over time and the effects those changes have had on education, the workplace, and the global society.	CSFS.912.PCGE.2.3. Discuss and analyze the impact of values and points of view that are presented in media messages (e.g. racial, gender, political).
		CSFS.35.PCGE.2.4. Identify the legal and	CSFS.68.PCGE.2.4. Discuss how the unequal	CSFS.912.PCGE.2.4. Analyze the positive and negative impacts of

		social consequences of cyberbullying/harassment in social computing.	distribution of computing resources in a global economy raises issues of equity, access, and power.	technology on popular culture and personal life.
		CSFS.35.PCGE.2.5. Explain how access to technology helps empower individuals and groups (e.g., gives them access to information, the ability to communicate with others around the world, and allows them to buy and sell things).	CSFS.68.PCGE.2.5. Describe ways in which adaptive technologies can assist users with special needs to function in their daily lives.	CSFS.912.PCGE.2.5. Construct strategies to combat cyberbullying or online harassment.
		CSFS.35.PCGE.2.6. Identify ways in which people with special needs access and use adaptive technology.	CSFS.68.PCGE.2.6. Identify and discuss the technology proficiencies needed in the workplace as well as ways to prepare to meet these demands.	CSFS.912.PCGE.2.6. Discuss the impact of computing on business and commerce (e.g., automated inventory processing, financial transactions, e-commerce, virtualization, and cloud computing).
		CSFS.35.PCGE.2.7. Use developmentally appropriate terminology	CSFS.68.PCGE.2.7. Interpret writings and/or communications which use	CSFS.912.PCGE.2.7. Discuss how technology has changed the way people

		when communicating about technology.	developmentally appropriate terminology.	build and manage organizations and how technology impacts personal life. CSFS.912.PCGE.2.8. Evaluate ways in which adaptive technologies may assist users with special needs. CSFS.912.PCGE.2.9. Explain how societal and economic factors are affected by access to critical information. CSFS.912.PCGE.2.10. Discuss the challenges (e.g., political, social, and economic) in providing equal access and distribution of technology in a global society. CSFS.912.PCGE.2.11. Construct writings and/or communications using

				developmentally appropriate terminology.
CSFS.PCGE.3. Evaluation of digital information resources		CSFS.35.PCGE.3.1. Identify digital information resources used to answer research questions (e.g., online library catalog, online encyclopedias, databases, and websites.) CSFS.35.PCGE.3.2. Gather, organize, and analyze information from digital resources. CSFS.35.PCGE.3.3. Compare digital resources for accuracy, relevancy, and appropriateness.	CSFS.68.PCGE.3.1. Analyze how media and technology can be used to distort, exaggerate, and misrepresent information. CSFS.68.PCGE.3.2. Describe strategies for determining the reliability of resources of information on the Internet.	CSFS.912.PCGE.3.1. Evaluate quality of digital resources for reliability (i.e., currency, relevancy, authority, accuracy, and purpose of digital information). CSFS.912.PCGE.3.2. Evaluate the accuracy, relevance, comprehensiveness, appropriateness, and bias of electronic information resources.

CSFS.PCGE.4 Security, privacy, information sharing, ownership, licensure and copyright.		CSFS.35.PCGE.4.1. Describe the difference between digital artifacts that are open or free and those that are protected by copyright.	CSFS.68.PCGE.4.1. Explain the guidelines for the fair use of downloading, sharing or modifying of digital artifacts.	CSFS.912.PCGE.4.1. Describe how different types of software licenses (e.g., open source and proprietary licenses) can be used to share and protect intellectual property.
		CSFS.35.PCGE.4.2. Understand fair use for using copyrighted materials (e.g., images, music, video, and text).	CSFS.68.PCGE.4.2. Explain how copyright law and licensing protect the owner of intellectual properties.	CSFS.912.PCGE.4.2. Explain how access to information may not include the right to distribute the information.
		CSFS.35.PCGE.4.3. Describe the purpose of copyright and the possible consequences for inappropriate use of digital artifacts that are protected by copyright.	CSFS.68.PCGE.4.3. Explain possible consequences of violating intellectual property law.	CSFS.912.PCGE.4.3. Describe differences between open source, freeware, and proprietary software licenses, and how they apply to different types of software.
		CSFS.35.PCGE.4.4. Describe the threats to safe and efficient use of devices (e.g., SPAM, spyware, phishing, and viruses) associated with	CSFS.68.PCGE.4.4. Identify threats and actions that protect devices from viruses, intrusion, vandalism, and other malicious activities.	CSFS.912.PCGE.4.4. Discuss security and privacy issues that relate to computer networks.

		various forms of technology use (e.g., downloading and executing software programs, following hyperlinks, and opening files).	CSFS.68.PCGE.4.5. Demonstrate compliance with the school's Acceptable Use Policy.	CSFS.912.PCGE.4.5. Identify computer-related laws and analyze their impact on digital privacy, security, intellectual property, network access, contracts, and harassment.
			CSFS.68.PCGE.4.6. Use digital citation tools to cite sources using a school - or district - adopted format (e.g., MLA and APA) including proper citation for all text and non-text sources (e.g., images, audio, and video).	CSFS.912.PCGE.4.6. Describe security and privacy issues that relate to computer networks including the permanency of data on the Internet, online identity, and privacy.
				CSFS.912.PCGE.4.7. Discuss the impact of

				government regulation on privacy and security.
CSFS.PCGE.5. Careers and Computers	CSFS. K2.PCGE.5.1. Recognize that people use computing technology in the workplace to perform many important tasks and functions.	CSFS. 35.PCGE.5. 1. Identify and describe how computing knowledge is essential to performing important tasks and functions.	CSFS. 68.PCGE.5.1. Identify interdisciplinary careers that are enhanced by computer science.	CSFS. 912.PCGE.5.1. Explore a variety of careers to which computing is central. CSFS.912.PCGE.5.2. Predict future careers and the technologies that may exist based on current technology trends.

CSFS.CC: Communication and Collaboration

Standard	K-2 Benchmarks	3-5 Benchmarks	6-8 Benchmarks	9-12 Benchmarks
CSFS.CC 1. Communication and collaboration	CSFS.K2.CC.1.1. Identify a variety of digital tools used for communication and collaboration (e.g., online library catalogs and databases). CSFS.K2.CC.1.2. Conduct basic keyword searches, and exchange information and feedback with teachers and other students (e.g., e-mail and text messaging). CSFS.K2.CC.1.3. Collaborate and cooperate with peers, teachers, and others using technology to solve problems.	CSFS.35.CC.1.1. Identify and utilize technology tools for individual and collaborative data collection, writing, communication, and publishing activities. CSFS.35.CC.1.2. Discuss key ideas and details while working individually or collaboratively using digital tools and media-rich resources in a way that informs, persuades, and/or entertains. CSFS.35.CC.1.3. Identify ways that teamwork and collaboration can support problem solving and innovation.	CSFS.68.CC.1.1. Demonstrate an ability to communicate appropriately through various online tools. CSFS.68.CC.1.2. Apply productivity and or multimedia tools for local and global group collaboration. CSFS.68.CC.1.3. Collaborate synchronously and asynchronously with peers, experts, and others to design, develop, and publish using a variety of	CSFS.912.CC.1.1. Evaluate modes of communication and collaboration. CSFS.912.CC.1.2. Select appropriate tools within a project environment to communicate with project team members. CSFS.912.CC.1.3. Use a variety of computing devices (e.g., probes, sensors, handheld devices, etc.) to collect, analyze, and present information for

	CSFS.K2.CC 1.4. Provide and accept constructive criticism on a collaborative project.	CSFS.35.CC.1.4. Describe how collaborating with others can be beneficial to a project. CSFS.35.CC.1.5. Understand that providing feedback to and receiving feedback from others can improve performance and outcomes.	digital tools and media-rich resources that demonstrate and communicate concepts to inform, persuade, and/or entertain. CSFS.68.CC.1.4. Utilize essential skills for collaboration: providing useful feedback, integrating feedback, understanding and accepting multiple perspectives, and socialization.	content-related problems, individually and collaboratively. CSFS.912.CC.1.4. Utilize project collaboration tools (such as version control systems and integrated development environments) while working on a collaborative software project CSFS.912.CC.1.5. Generate, evaluate, and prioritize questions that can be researched through digital resources and online tools. CSFS.912.CC.1.6. Perform advanced searches to locate information and/or design a data-collection approach to gather original data (e.g., qualitative interviews,

				surveys, prototypes, and simulations). CSFS.912.CC.1.7. Communicate and publish key ideas and details to a variety of audiences using digital tools and media-rich resources. CSFS.912.CC.1.8. Identify how collaboration influences the design and development of software artifacts. CSFS.912.CC.1.9. Evaluate program designs and implementations written by others for readability and usability.

CSFS.CCS.: Computing and Communication Systems

Standard	K-2 Benchmarks	3-5 Benchmarks	6-8 Benchmarks	9-12 Benchmarks
CSFS.CCS.1. Problem solving and Algorithms	CSFS.K2.CCS.1.1. Understand how to arrange (sort) information into useful order, such as sorting students by birth date, with or without technology. CSFS.K2.CCS.1.2. Solve age-appropriate problems (puzzles, logical thinking programs) with or without technology.	CSFS.35.CCS.1.1. Use information organized using digital graphic organizers (concept maps, venn-diagrams) to solve age-appropriate problems. CSFS.35.CCS.1.2. Demonstrate the concepts of sequences, loops, and branches (conditionals and events) at an age-appropriate level without using technology.	CSFS.68.CCS.1.1. Create, modify, and use a database (e.g., define field formats, adding new records, manipulate data) to analyze data and propose solutions for a task/problem, individually and collaboratively. CSFS.68.CCS.1.2. Perform a variety of operations such as sorting, filtering, and searching in a database to organize and display information in a variety of ways such as number formats (e.g., scientific notation, percentages, and exponents), charts, tables and graphs.	CSFS.912.CCS.1.1. Explain intractable problems and understand that problems exists that are computationally unsolvable (undecidable). (e.g. classic intractable problems include Towers of Hanoi, TSP). CSFS.912.CCS.1.2. Explain the value of heuristic algorithms to approximate solutions for intractable problems (e.g. a heuristic solution to TSP).

	CSFS.K2.CCS.1.3. Define an algorithm as a sequence of defined steps. CSFS.K2.CCS.1.4. Create simple algorithm, individually and collaboratively, without using computers to complete the task (e.g., making a sandwich, getting ready for school). CSFS.K2.CCS.1.5. Use writing tools, digital cameras, and drawing tools to illustrate thoughts, ideas, and stories in a step-by-step manner.	CSFS.35.CCS.1.3. Explain the process of arranging (sorting) information into useful order as well as the purpose for doing so. CSFS.35.CCS.1.4. Identify the parts of an algorithm as sequences, loops, and branches. CSFS.35.CCS.1.5. Understand that there are several possible algorithms for searching within a dataset (such as finding a specific word in a word list or card in a deck of cards).	CSFS.68.CCS.1.3. Decompose a problem and create a function for one of its parts at a time (e.g., video game, robot obstacle course, making dinner), individually and collaboratively. CSFS.68.CCS.1.4. Create a program that implements an algorithm to achieve a given goal, individually and collaboratively. CSFS.68.CCS.1.5. Design solutions that use repetition and two-way selection (e.g., FOR, WHILE, IF/ELSE).	CSFS.912.CCS.1.3. Describe the concept of parallel processing as a strategy to solve large problems. CSFS.912.CCS.1.4. Demonstrate concurrency by separating processes into threads of execution and dividing data into parallel streams. CSFS.912.CCS.1.5. Use predefined functions and parameters, classes, and methods to divide a complex problem into simpler parts by using the principle of abstraction to manage complexity (e.g. by using searching and sorting as abstractions).

	CSFS.K2.CCS.1.6. Develop an algorithm using tangible materials or present the algorithm in a visual medium (e.g., storyboard)	CSFS.35.CCS.1.6. Write an algorithm to solve a grade-level appropriate problem (e.g.move a character through a maze, instruct a character to draw a specific shape, have a character start, repeat or end activity as required or upon a specific event), individually or collaboratively.	CSFS.68.CCS.1.6. Recognize that boundaries need to be taken into account for an algorithm to produce correct results.	CSFS.912.CCS.1.6. Critically examine classical algorithms and implement an original algorithm.
	CSFS.K2.CCS.1.7. Gather and organize information using concept-mapping tools.	CSFS.35.CCS.1.7. Identify and correct logical errors in algorithms; written, mapped, live action, or digital.	CSFS.68.CCS.1.7. Identify simple data structures.	CSFS.912.CCS.1.7. Explain how sequence, selection, iteration, and recursion are building blocks of algorithms.
		CSFS.35.CCS.1.8. Systematically test and identify logical errors in algorithms.	CSFS.68.CCS.1.8. Recognize that more than one algorithm can solve a given problem.	CSFS.912.CCS.1.8. Decompose a problem by defining new functions and classes.
		CSFS.35.CCS.1.9. Explain how to correct logical errors in algorithms;	CSFS.68.CCS.1.9. Use logical reasoning to predict	CSFS.912.CCS.1.9. Evaluate ways to characterize how well algorithms perform and

		written, mapped, live action, or digital.	outputs while showing an understanding of inputs. CSFS.68.CCS.1.10. Select the 'best' algorithm based on a given criteria (e.g., time, resource, accessibility) to solve a problem, individually and collaboratively. CSFS.68.CCS.1.11. Use iterative development and debugging to explore the problem domain. [6-8.CT.d.7 p23] CSFS.68.CCS.1.12. Perform program tracing to predict the behavior of programs.	that two algorithms can perform differently for the same task. CSFS.912.CCS.1.10. Design and implement a simple simulation algorithm to analyze, represent and understand natural phenomena. CSFS.912.CCS.1.11. Evaluate algorithms by their efficiency, correctness, and clarity (e.g. by analyzing and comparing execution times, testing with multiple inputs or data sets, and by debugging). CSFS.912.CCS.1.12. Compare and contrast simple data structures and their uses.

				CSFS.912.CCS.1.13. Explain how automated software testing can reduce the cost of the testing effort. CSFS.912.CCS.1.14. Explain what tools are applied to provide automated testing environments.
CSFS.CCS.2. Modeling and Simulations	CSFS.K2.CCS.2.1. Define simulation and identify the concepts illustrated by a simple simulation (e.g. growth and health, butterfly life cycle.) CSFS.K2.CCS.2.2. Describe how models represent a real-life system (e.g. globe, map.)	CSFS.35.CCS.2.1. Identify the concepts illustrated by a simulation (e.g. ecosystem, predator/prey, invasive species). CSFS.35.CCS.2.2. Use data from a simulation to answer a question, individually and collaboratively. CSFS.35.CCS.2.3. Create a simple model of a system	CSFS.68.CCS.2.1. Examine connections between elements of mathematics and computer science including binary numbers, logic, sets, and functions. CSFS.68.CCS.2.2. Create/modify and use a simulation to analyze and illustrate a concept in depth (e.g., use a simulation to illustrate a genetic variation.), individually and collaboratively. CSFS.68.CCS.2.3. Evaluate what kinds of real-world	CSFS.912.CCS.2.1. Analyze data and identify patterns through modeling and simulation. [CSTAp57 3B-9] CSFS.68.CCS.2.2. Use models and simulations to help formulate, refine, and test scientific hypotheses. CSFS.68.CCS.2.3. Use data analysis to enhance the

		(e.g. cell, solar system) and explain what the model shows and does not show.	problems can be solved using modeling and simulation. CSFS.68.CCS.2.4. Interact with content-specific models and simulations to support learning, research and problem solving (e.g., immigration, international trade, invasive species).	understanding of complex natural and human systems. CSFS.68.CCS.2.4. Compare techniques for analyzing massive data collections. CSFS.68.CCS.2.5. Describe how computation shares features with art and music (by translating human intention into an artifact). CSFS.68.CCS.2.6. Use modeling and simulation to represent and understand natural phenomena.
CSFS.CCS.3. Digital tools.	CSFS.K2.CCS.3.1.	CSFS.35.CCS.3.1.	CSFS.68.CCS.3.1.	CSFS.912.CCS.3.1.

	Create a digital artifact (independently and collaboratively) that clearly expresses thoughts and ideas. CSFS.K2.CCS.3.2. Use digital tools to create, review and revise artifacts that include text, images and audio, individually or collaboratively.	Use digital tools (local and online) to manipulate and publish multimedia artifacts. CSFS.35.CCS.3.2. Create an artifact (independently and collaboratively) that answers a research question clearly communicating thoughts and ideas.	Create an artifact (independently and collaboratively) that answers a research question and communicates results and conclusions. CSFS.68.CCS.3.2. Explain why different file types exist (e.g., formats for word processing, images, music, and three-dimensional drawings). CSFS.68.CCS.3.3. Identify the kinds of content associated with different file types. CSFS.68.CCS.3.4. Integrate information from multiple file formats into a single artifact.	Discuss digital tools or resources to use for a real-world task based on their efficiency and effectiveness, individually and collaboratively. 2. Evaluate different file types for different purposes (e.g., word processing, images, music, and three-dimensional drawings).
CSFS.CCS.4. 4. Hardware and software	CSFS.K2.CCS.4.1. Identify different kinds of computing devices in the classroom and other	CSFS.35.CCS.4.1. Identify the basic components of a computer (e.g., monitor,	CSFS.68.CCS.4.1. Identify and describe the function of the main internal parts of a basic computing device	CSFS.912.CCS.4.1. Describe a software development process that is used to solve problems at different

	places (e.g., laptops, tablets, smart phones, desktops, printers). CSFS.K2.CCS.4.2. Recognize and operate different types of computers, applications and peripherals (e.g., use input/output devices such as a mouse, keyboard, or touch screen; find, navigate, launch a program). CSFS.K2.CCS.4.3 Understand that a computer program is running when a program or command is executed.	keyboard, mouse, controller, speakers). CSFS.35.CCS.4.2. Describe the function and purpose of various input/output devices and peripherals (e.g., monitor, screen, keyboard, controller, speakers). CSFS.35.CCS.4.3. Distinguish between hardware and software. CSFS.35.CCS.4.4. Identify and solve simple hardware	(e.g., motherboard, hard drive, Central Processing Unit (CPU)). CSFS.68.CCS.4.2. Describe the main functions of an operating system and explain how an operating system provides user and system services (e.g. user interface, IO device management, task management). CSFS.68.CCS.4.3. Discuss the relationships between hardware and software (e.g., BIOS, operating systems and firmware) CSFS.68.CCS.4.4. Identify and describe the use of	software development stages (e.g. design, coding, testing, and verification). CSFS.912.CCS.4.2. Describe the organization of a computer and identify its principal components by name, function, and the flow of instructions and data between components (e.g. storage devices, memory, CPU, graphics processors, IO and network ports). CSFS.912.CCS.4.3. Differentiate between multiple levels of hardware and software (such as CPU hardware, operating system, translation, and interpretation) that support program execution. CSFS.912.CCS.4.4. Evaluate various forms of input and

		and software problems that may occur during everyday use (e.g., power, connections, application window or toolbar).	sensors, actuators, and control systems in an embodied system (e.g., a robot, an e-textile, installation art, smart room). CSFS.68.CCS.4.5. Evaluate a hardware/software problem and construct the steps involved in diagnosing and solving the problem (e.g., power, connections, application window or toolbar, cables, ports, network resources, video, and sound). CSFS.68.CCS.4.6. Describe the essential characteristics of a software artifact. CSFS.68.CCS.4.7. Describe the major components and	output (e.g., IO and storage devices and digital media). CSFS.912.CCS.4.5. Develop and evaluate criteria for purchasing or upgrading computer system hardware (e.g., Wi-Fi, mobile devices, home and office machines). CSFS.912.CCS.4.6. Develop criteria for selecting appropriate hardware and software when solving a specific real-world problem (such as business, educational, personal). CSFS.912.CCS.4.7. Develop a software artifact (independently and

			functions of computer systems and networks. CSFS.68.CCS.4.8. Identify software used to support specialized forms of human-computer interaction	collaboratively) in phases (or stages) according to a common software development methodology (e.g. Waterfall or Spiral model). CSFS.912.CCS.4.8. Evaluate the basic components of computer networks. CSFS.912.CCS.4.9. Analyze historical trends in hardware and software to assess implications on computing devices for the future (e.g., upgrades for power/energy, computation capacity, speed, size, ease of use).
CSFS.CCS.5. Network systems			CSFS.68.CCS.5.1. Describe how information, both text and non-text, is translated and communicated between digital computers over a computer network. CSFS.68.CCS.5.2.	CSFS.912.CCS.5.1. Identify and select the most appropriate file format based on trade-offs (e.g. open file formats, text, proprietary and binary formats, compression and encryption formats).

			Explain the difference between physical (wired), local area wireless, and mobile networks. CSFS.68.CCS.5.3. Model the components of a network.	CSFS.912.CCS.5.2. Describe the issues that impact network functionality (e.g. latency, bandwidth, firewalls and server capability). CSFS.912.CCS.5.3 Describe common network protocols, such as IP, TCP, SMTP, HTTP, and FTP, and how these are applied by client-server and peer-to-peer networks.
CSFS.CCS.6. 6. Human – Computer interactions and Artificial Intelligence		CSFS.35.CCS.6.1. Explain how hardware applications (e.g., Global Positioning System (GPS) navigation for driving directions, text-to-speech translation, and language translation) can enable everyone to do things they could not do otherwise. CSFS.35.CCS.6.2. Compare and contrast	CSFS.68.CCS.6.1. Explain why some tasks can be accomplished more easily by computers. CSFS.68.CCS.6.2. Describe how humans and machines	CSFS.912.CCS.6.1. Describe the unique features of computers embedded in mobile devices and vehicles. CSFS.912.CCS.6.2. Describe the common physical and

		human and computer performance on similar tasks (e.g., sorting alphabetically or finding a path across a cluttered room) to understand which is best suited to the task. CSFS.35.CCS.6.3. Recognize that computers model intelligent behavior (as found in robotics, speech and language recognition, and computer animation.)	interact to accomplish tasks that cannot be accomplished by either alone. CSFS.68.CCS.6.3. Identify novel ways humans interact with computers, including probes, sensors, and handheld devices. CSFS.68.CCS.6.4. Describe ways in which computers use models of intelligent behavior (e.g., robot motion, speech and language understanding, and computer vision). CSFS.68.CCS.6.5. Identify factors that distinguish humans from machines.	cognitive challenges faced by users when learning to use software and hardware CSFS.912.CCS.6.3. Describe the process of designing software to support specialized forms of human-computer interaction. CSFS.912.CCS.6.4. Explain the notion of intelligent behavior through computer modeling and robotics. CSFS.912.CCS.6.5. Describe common measurements of machine intelligence (e.g. Turing test).

			CSFS.68.CCS.6.6. Design and demonstrate the use of a device (e.g., robot, e-textile) to accomplish a task, individually and collaboratively.	CSFS.912.CCS.6.6. Describe a few of the major branches of artificial intelligence (e.g. expert systems, natural language processing, machine perception, machine learning). CSFS.912.CCS.6.7. Describe major applications of artificial intelligence and robotics, including, but not limited to, the medical, space, and automotive fields.

CSFS.CPP: Computer Practices and Programming

Standard	K-2 Benchmarks	3-5 Benchmarks	6-8 Benchmarks	9-12 Benchmarks
CSFS.CPP1 Data Analysis	CSFS.K2.CPP1.1. Identify different kinds of data (eg. text, charts, graphs, numbers, pictures, audio, video, collections of objects.) CSFS.K2.CPP1.2. Identify, research, and collect a data set on a topic, issue, problem, or question using age-appropriate technologies.	CSFS.35.CPP1.1. Identify and describe examples of databases from everyday life (e.g., library catalogs, school records, telephone directories, and contact lists). CSFS.35.CPP1.2. Collect and manipulate data using a variety of computing methods (e.g., sorting, totaling, and averaging). CSFS.35.CPP1.3.	CSFS.68.CPP1.1. Select and use data-collection technology (e.g., probes, handheld devices, geographic mapping systems and output from multiple runs of a computer program) to gather, view, organize, analyze, and report results for content-related problems, individually and collaboratively.	CSFS.912.CPP1.1. Analyze and manipulate data collected by a variety of data collection techniques to support a hypothesis. CSFS.912.CPP1.2. Collect real-time data from sources such as simulations, scientific and robotic sensors, and device emulators, using this data to formulate strategies or algorithms to solve advanced problems.

	CSFS.K2.CPP1.3. Propose a developmentally appropriate solution to a problem or question based on a analysis of the data and critical thinking, individually and collaboratively. CSFS.K2.CPP1.4. Create data visualizations (e.g., charts and infographics), individually and collaboratively.	Utilize a database, such as a spreadsheet, to collect, organize, graph, and analyze data to answer a question.		
CSFS.CPP2 2. Computer Programming Basics	CSFS.K2.CPP2.1. Define a computer program as a set of commands created by people to do something. CSFS.K2.CPP2.2. Perform a simple task (e.g. making a sandwich, brushing teeth) breaking it into small steps. CSFS.K2.CPP2.3. Explain that computers only	CSFS.35.CPP2.1. Create, test, and modify a program in a graphical environment (e.g., block-based visual programming language), individually and collaboratively. CSFS.35.CPP2.2. Use arithmetic operators, conditionals, and repetition in programs	CSFS.68.CPP2.1. Use visual representations of problem states, structures and data. CSFS.68.CPP2.2. Evaluate the logical flow of a step-by-step program by acting it out through computer-free activities.. CSFS.68.CPP2.3. Develop problem solutions using a	CSFS.912.CPP2.1. Explain the program execution process (by an interpreter and in CPU hardware). CSFS.912.CPP2.2. Use global and local scope appropriately in program implementation. CSFS.912.CPP2.3. Use an industrial-strength

	follow the program's instructions. CSFS.K2.CPP2.4. Construct a simple program using tools that do not require a textual programming language (e.g. block-based programming language). CSFS.K2.CPP2.5. Use interactive debugging to detect and correct simple program errors.	CSFS.35.CPP2.3. Recognize that programs need known initial conditions (e.g., set initial score to zero in a game, initialize variables, or initial values set by hardware input. CSFS.35.CPP2.4. Use interactive debugging to detect and correct program errors, including those involving arithmetic operators, conditionals, and repetition.	block programming language, including all of the following: looping behavior, conditional statements, expressions, variables, and functions. CSFS.68.CPP2.4. Develop problem solutions using a programming language, including all of the following: looping behavior, conditional statements, expressions, variables, and functions.	integrated development environment to implement a program. CSFS.912.CPP2.4. Use application programming interfaces (APIs) and libraries to facilitate programming solutions. CSFS.912.CPP2.5. Explain the role of an API in the development of applications and the distinction between a programming language's syntax and the API. CSFS.912.CPP2.6. Describe a variety of commonly used programming languages.

				CSFS.912.CPP2.7. Classify programming languages by paradigm and application domain (e.g. imperative, functional, logic languages and how well suited they are for certain application domains such as web programming, symbolic processing, data/numerical processing etc.).
CSFS.CPP3 Program Product Development	CSFS.K2.CPP3.1. Create developmentally appropriate multimedia products with support from teachers, family members, or student partners. CSFS.K2.CPP3.2. Give a simple presentation of products limited to presenting basic	CSFS.35.CPP3.1. Use technology tools for individual and collaborative writing, communication, and publishing activities. CSFS.35.CPP3.2. Present created products either individually and	CSFS.68.CPP3.1. Select appropriate tools and technology resources to accomplish a variety of tasks and solve problems. CSFS.68.CPP3.2. Use advanced tools to design and create online	CSFS.912.CPP3.1. Use a development process in creating a computational artifact, individually and collaboratively, followed by reflection, analysis, and iteration (e.g., data-set analysis program for science and engineering fair, capstone project that includes a program, term research project based on program data). CSFS.912.CPP3.2. Create mobile computing applications and/or dynamic

	information (e.g., title of project, a few basic facts) about the topic.	collaboratively where a topic, concept, or skill is carefully analyzed or thoughtfully explored.	content (e.g., webpage, blog, digital portfolio, multimedia), individually and collaboratively.	web pages through the use of a variety of design and development tools, programming languages and mobile devices/emulators.