

MISSOURI STATE BOARD OF EDUCATION AGENDA ITEM:		May 2019	
CONSIDERATION OF COMPUTER SCIENCE LEARNING STANDARDS			
STATUTORY AUTHORITY:			
Section 170.018, RSMo		<input type="checkbox"/> Consent Item	<input checked="" type="checkbox"/> Action Item
		<input type="checkbox"/> Report Item	
STRATEGIC PRIORITY			
Access, Opportunity, Equity – Provide all students access to a broad range of high-quality educational opportunities from early learning into post-high school engagement.			
SUMMARY			
Section 170.018, RSMo directed the State Board of Education (SBE) to adopt and implement academic performance standards relating to computer science beginning in the 2019-20 school year. The law also specified that the Department of Elementary and Secondary Education (DESE) organize work groups to develop and recommend rigorous academic performance standards relating to computer science for students in kindergarten and in each grade not higher than twelfth grade. The work groups began meeting in November 2018 and the leaders of the work groups submitted their first draft of the Computer Science Academic Performance Standards to the SBE on February 19, 2019.			
The Department has solicited comments and feedback regarding the proposed Computer Science Academic Performance Standards from the public. The Department received many comments from these sources which have been incorporated into the work group product, where appropriate. The attached proposed Computer Science Academic Performance Standards will be presented for the SBE approval.			
PRESENTERS			
Blaine Henningsen, Assistant Commissioner and Lori Brewer, Director of Business, Marketing, and Information Technology, Office of College and Career Readiness, will assist with the presentation and discussion of this agenda item.			
RECOMMENDATION			
The Department recommends that the State Board of Education approve the Computer Science Academic Performance Standards as presented.			

State Board of Education

K-12 Computer Science Academic Performance Standards



May 2019



Update

- September 19, 2018 - House Bill 3 was passed, by the General Assembly during the 2nd Special Session
- October 30, 2018 - Governor Parson signed the legislation into law
- Information now in Section 170.018, RSMo

Section 170.018 – What’s Different?

- DESE shall convene a work group to develop and recommend rigorous K-12 academic performance standards for computer science
- State board of education shall adopt and implement academic performance standards relating to computer science beginning in the 2019-20 school year
- Statute states that “the requirements of section 160.514 shall not apply to this section.”

Work Group Members

Expert Resources

- **CISCO** – Kurtis Harrison
- **Code.Org** – Katie Hendrickson
- **GlobalHack** – Beth Casagrand
- **IBM** – Lucy LePage
- **KC Stem Alliance** – Martha McCabe
- **KC Tech Council** – Ryan Weber
- **LaunchCode** – Chris Bay
- **Microsoft** – Marilyn Allen
- **PLTW** – Andrea Holzwarth
- **Cerner** – Melissa Boyd

Educational Resources

Rusty Monhollon

DHE- Asst. Commissioner of Academic Affairs

Larry Felton

Missouri School Boards' Association

Lori Brewer

DESE - Director of Business, Marketing and IT

Matt Johnson

DESE - Director of Engineering and Technology

Chip Sharp

DESE - Director of Mathematics

Kristen McKinney

DESE - Director of Science

Lisa Sireno

DESE - Standard and Assessment Administrator

Grades K-2

Name	District
Diana Demoss	Columbia
Chantall Mason	Webster Groves
Michael Stewart	East Lynne
Anita Studdard	Strain-Japan
Stephanie Phillips – Group Leader	Ozark

Grades 3-5

Name	District
Kristin Burkemper	Columbia
Megan Frueh	Nodaway Holt
Beth McCann	Kingston
Adrienne Patton	Columbia
April Wamble	Richland
Julie Williams	West Plains
Doug Barton – Group Leader	Lindbergh

Grades 6-8

Name	District
Christina Chandler	Center
Betty Chism	Ferguson-Florissant
Monica Clayton	Hannibal
Sarah Golubski	Clever
Mike Hall	Columbia
Robert Monroe	Fulton
Bob Deneau – Group Leader	Rockwood

Grades 9-10

Name	District/College
Garret Andreasen	Meramec Valley
Charlene Atkins	University of Central Missouri
Martha Catchings	Hickman Mills
Tim DeClue	Southwest Baptist University
Glenn Hancock	Rockwood
Cheryl Ingram	Crowder College
Matt Meeks	Blair Oaks
Lynette Pierce	Kennett
Renee Turpin	South Nodaway
Patrick Sasser – Group Leader	Columbia

Grades 11-12

Name	District/College
Lisa Arment	Marion County R-2
Robert Bowers	Lathrop
Reginald Brigham	East Central College
Phil Fleming	Fort Osage
Gina Green	Bolivar
Jennifer Hegeman	Missouri Western University
Daniel Kopsas	Ozark Technical College
Robin Robertson	Republic
Mary Shepherd	Northwest Missouri State University
Mark Young	Webster Groves
Kevin Clevenger – Group Leader	Blue Springs

Process

- Work groups were organized according to state statute
- November 9, 2018 - First meeting of work groups and experts
- Other meetings and information sharing were conducted in person, conference calls, WebEx, Zoom, Google Docs, etc.

Seeking Feedback

- February 19, 2019 - Public hearing at State Board
Transcript posted on DESE website at
<https://dese.mo.gov/college-career-readiness/curriculum/computer-science-missouri-learning-standards>
- February 19 through March 11, 2019 - Solicited public comments and feedback on the proposed standards
- May 6, 2019 – Presented to the Joint Committee on Education

Response to Public Comments

- All comments have been posted on the department's website at <https://dese.mo.gov/college-career-readiness/curriculum/computer-science-missouri-learning-standards>
- Work groups reviewed comments and standards were revised as needed.

Documents

- New, revised K-12 Performance Standards
- Glossary
- All documents posted on the DESE Missouri Learning Standards website

Section 170.018 – What’s New?

- Students may fulfill one unit of academic credit in mathematics, science or practical arts with a computer science course
- Students are still required to take mathematics or science End-of-Course (EOC) exams.

Section 170.018 – Required Documentation

- If a student wishes to substitute a computer science course for a mathematics course
 - ❑ the school district shall ...
 - ❑ the parents/guardians must ...

Core Concepts

- **5 core concepts span all grade levels:**
 - ❑ Impacts of Computing
 - ❑ Networks and the Internet
 - ❑ Algorithms and Programming
 - ❑ Computing Systems
 - ❑ Data and Analysis

Examples of Subconcepts

- **Social Interactions** (Subconcept of Impact of Computing)
 - In early grades, students learn that computing can connect people and support interpersonal communication. As they progress, students learn how the social nature of computing affects institutions and careers in various sectors.

Examples of Subconcepts

- **Cybersecurity** (Subconcept of Networks and the Internet)
 - In early grades, students learn how to protect their personal information. As they progress, students learn increasingly complex ways to protect information sent across networks.

Examples of Subconcepts

- **Safety, Law and Ethics** (Subconcept of Impacts of Computing)
 - In early grades, students learn the fundamentals of digital citizenship and appropriate use of digital media. As they progress, students learn about the legal and ethical issues that shape computing practices.

Work Group Remarks

- Missouri students will
 - ❑ become better problem solvers
 - ❑ navigate the internet with greater security and privacy
 - ❑ be better prepared for high paying jobs in high demand
 - ❑ develop both soft skills and technical skills
 - ❑ understand their impact and influence as citizens

Recommendation

We recommend the State Board of Education approve the proposed Computer Science Performance Standards as presented.

Special Thanks

Work group leaders

- ❑ Grades K-2 Stephanie Phillips
- ❑ Grades 3-5 Doug Barton
- ❑ Grades 6-8 Bob Deneau
- ❑ Grades 9-10 Patrick Sasser
- ❑ Grades 11-12 Kevin Clevenger

Next Steps

Districts should begin preparing local curriculum based on the new standards for implementation in the 2019-2020 school year.

GRADES K-12

COMPUTER SCIENCE

PERFORMANCE STANDARDS

Glossary of Terms

Revised April 11, 2019

Computer Science Glossary of Terms

Source: K-12 Computer Science Framework

A	
<i>abstraction</i>	(process): The process of reducing complexity by focusing on the main idea. By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the problem. (product): A new representation of a thing, a system, or a problem that helpfully reframes a problem by hiding details irrelevant to the question at hand. [MDESE, 2016]
<i>accessibility</i>	The design of products, devices, services, or environments for people who experience disabilities. Accessibility standards that are generally accepted by professional groups include the Web Content Accessibility Guidelines (WCAG) 2.0 and Accessible Rich Internet Applications (ARIA) standards. [Wikipedia]
<i>algorithm</i>	A step-by-step process to complete a task.
<i>analog</i>	The defining characteristic of data that is represented in a continuous, physical way. Whereas digital data is a set of individual symbols, analog data is stored in physical media, such as the surface grooves on a vinyl record, the magnetic tape of a VCR cassette, or other nondigital media. [Techopedia]
<i>app</i>	A type of application software designed to run on a mobile device, such as a smartphone or tablet computer. Also known as a mobile application. [Techopedia]
<i>artifact</i>	Anything created by a human. See computational artifact for the definition used in computer science.
<i>audience</i>	Expected end users of a computational artifact or system.
<i>accessibility</i>	The design of products, devices, services, or environments for people who experience disabilities. Accessibility standards that are generally accepted by professional groups include the Web Content Accessibility Guidelines (WCAG) 2.0 and Accessible Rich Internet Applications (ARIA) standards. [Wikipedia]
<i>authentication</i>	The verification of the identity of a person or process. [FOLDOC]
<i>automate; automation</i>	automate: To link disparate systems and software so that they become self-acting or self-regulating. [Ross, 2016] automation: The process of automating.
B	
<i>Boolean</i>	A type of data or expression with two possible values: true and false. [FOLDOC]
<i>bug</i>	An error in a software program. It may cause a program to unexpectedly quit or behave in an unintended manner. [Tech Terms] The process of finding and correcting errors (bugs) is called debugging. [Wikipedia]
C	
<i>code</i>	Any set of instructions expressed in a programming language. [MDESE, 2016]
<i>comment</i>	A programmer-readable annotation in the code of a computer program added to make the code easier to understand. Comments are generally ignored by machines. [Wikipedia]
<i>complexity</i>	The minimum amount of resources, such as memory, time, or messages, needed to solve a problem or execute an algorithm. [NIST/DADS]
<i>component</i>	An element of a larger group. Usually, a component provides a particular service or group of related services. [Tech Terms, TechTarget]
<i>computational</i>	Relating to computers or computing methods.

<i>computational artifact</i>	Anything created by a human using a computational thinking process and a computing device. A computational artifact can be, but is not limited to, a program, image, audio, video, presentation, or web page file. [College Board, 2016]
<i>computational thinking</i>	The human ability to formulate problems so that their solutions can be represented as computational steps or algorithms to be executed by a computer. [Lee, 2016]
<i>computer</i>	A machine or device that performs processes, calculations, and operations based on instructions provided by a software or hardware program. [Techopedia]
<i>computer science</i>	The study of computers and algorithmic processes, including their principles, their hardware and software designs, their implementation, and their impact on society. [ACM, 2006]
<i>computing</i>	Any goal-oriented activity requiring, benefiting from, or creating algorithmic processes. [MDESE, 2016]
<i>computing device</i>	A physical device that uses hardware and software to receive, process, and output information. Computers, mobile phones, and computer chips inside appliances are all examples of computing devices.
<i>computing system</i>	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 computing 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.
<i>conditional</i>	A feature of a programming language that performs different computations or actions depending on whether a programmer-specified Boolean condition evaluates to true or false. [MDESE, 2016] (A conditional could refer to a conditional statement, conditional expression, or conditional construct.)
<i>configuration</i>	(process): Defining the options that are provided when installing or modifying hardware and software or the process of creating the configuration (product). [TechTarget] (product): The specific hardware and software details that tell exactly what the system is made up of, especially in terms of devices attached, capacity, or capability. [TechTarget]
<i>connection</i>	A physical or wireless attachment between multiple computing systems, computers, or computing devices.
<i>connectivity</i>	A program's or device's ability to link with other programs and devices. [Webopedia]
<i>control;</i> <i>control structure</i>	control: (in general) 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. control structure: A programming (code) structure that implements control. Conditionals and loops are examples of control structures.
<i>culture;</i> <i>cultural practices</i>	culture: A human institution manifested in the learned behavior of people, including their specific belief systems, language(s), social relations, technologies, institutions, organizations, and systems for using and developing resources. [NCSS, 2013] cultural practices: The displays and behaviors of a culture.
<i>cybersecurity</i>	The protection against access to, or alteration of, computing resources through the use of technology, processes, and training. [TechTarget]
D	
<i>data</i>	Information that is 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. [CAS, 2013; Tech Terms]
<i>data structure</i>	A particular way to store and organize data within a computer program to suit a specific purpose so that it can be accessed and worked with in appropriate ways. [TechTarget]
<i>data type</i>	A classification of data that is distinguished by its attributes and the types of operations that can be performed on it. Some common

	data types are integer, string, Boolean (true or false), and floating-point.
<i>debugging</i>	The process of finding and correcting errors (bugs) in programs. [MDESE, 2016]
<i>decompose;</i> <i>decomposition</i>	decompose: To break down into components. decomposition: Breaking down a problem or system into components. [MDESE, 2016]
<i>device</i>	A unit of physical hardware that provides one or more computing functions within a computing system. It can provide input to the computer, accept output, or both. [Techopedia]
<i>digital</i>	A characteristic of electronic technology that uses discrete values, generally 0 and 1, to generate, store, and process data. [Techopedia]
<i>digital citizenship</i>	The norms of appropriate, responsible behavior with regard to the use of technology. [MDESE, 2016]
E	
<i>efficiency</i>	A measure of the amount of resources an algorithm uses to find an answer. It is usually expressed in terms of the theoretical computations, the memory used, the number of messages passed, the number of disk accesses, etc. [NIST/DADS]
<i>encapsulation</i>	The technique of combining data and the procedures that act on it to create a type. [FOLDOC]
<i>encryption</i>	The conversion of electronic data into another form, called ciphertext, which cannot be easily understood by anyone except authorized parties. [TechTarget]
<i>end user (or user)</i>	A person for whom a hardware or software product is designed (as distinguished from the developers). [TechTarget]
<i>event</i>	Any identifiable occurrence that has significance for system hardware or software. User-generated events include keystrokes and mouse clicks; system-generated events include program loading and errors. [TechTarget]
<i>event handler</i>	A procedure that specifies what should happen when a specific event occurs.
<i>execute;</i> <i>execution</i>	execute: To carry out (or “run”) an instruction or set of instructions (program, app, etc.). execution: The process of executing an instruction or set of instructions. [FOLDOC]
H	
<i>hardware</i>	The physical components that make up a computing system, computer, or computing device. [MDESE, 2016]
<i>hierarchy</i>	An organizational structure in which items are ranked according to levels of importance. [TechTarget]
<i>human–computer interaction (HCI)</i>	The study of how people interact with computers and to what extent computing systems are or are not developed for successful interaction with human beings. [TechTarget]
I	
<i>identifier</i>	The user-defined, unique name of a program element (such as a variable or procedure) in code. An identifier name should indicate the meaning and usage of the element being named. [Techopedia]
<i>implementation</i>	The process of expressing the design of a solution in a programming language (code) that can be made to run on a computing device.
<i>inference</i>	A conclusion reached on the basis of evidence and reasoning. [Oxford]
<i>input</i>	The signals or instructions sent to a computer. [Techopedia]
<i>integrity</i>	The overall completeness, accuracy, and consistency of data. [Techopedia]
<i>Internet</i>	The global collection of computer networks and their connections, all using shared protocols to communicate. [CAS, 2013]
<i>iterative</i>	Involving the repeating of a process with the aim of approaching a desired goal, target, or result. [MDESE, 2016]

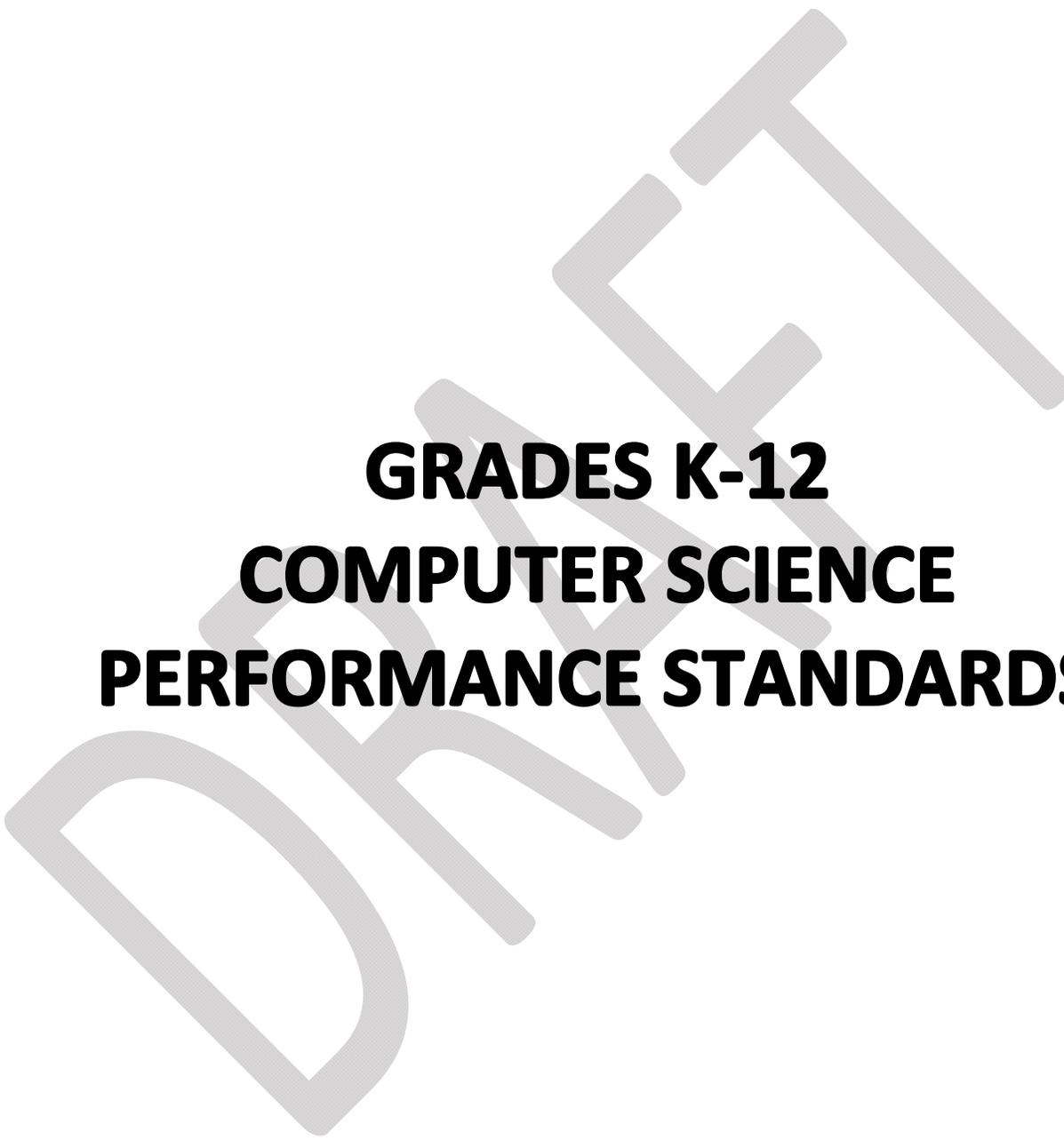
L

	<i>loop</i>	A programming structure that repeats a sequence of instructions as long as a specific condition is true. [Tech Terms]
M		
	<i>memory</i>	Temporary storage used by computing devices. [MDESE, 2016]
	<i>model</i>	A representation of some part of a problem or a system. [MDESE, 2016] Note: This definition differs from that used in science.
	<i>modularity</i>	The characteristic of a software/web application that has been divided (decomposed) into smaller modules. An application might have several procedures that are called from inside its main procedure. Existing procedures could be reused by recombining them in a new application. [Techopedia]
	<i>module</i>	A software component or part of a program that contains one or more procedures. One or more independently developed modules make up a program. [Techopedia]
N		
	<i>network</i>	A group of computing devices (personal computers, phones, servers, switches, routers, etc.) connected by cables or wireless media for the exchange of information and resources.
O		
	<i>operation</i>	An action, resulting from a single instruction that changes the state of data. [Free Dictionary]
P		
	<i>packet</i>	The unit of data sent over a network. [Tech Terms]
	<i>parameter</i>	A special kind of variable used in a procedure to refer to one of the pieces of data received as input by the procedure. [MDESE, 2016]
	<i>piracy</i>	The illegal copying, distribution, or use of software. [TechTarget]
	<i>procedure</i>	An independent code module that fulfills some concrete task and is referenced within a larger body of program code. The fundamental role of a procedure is to offer a single point of reference for some small goal or task that the developer or programmer can trigger by invoking the procedure itself. [Techopedia] In this framework, procedure is used as a general term that may refer to an actual procedure or a method, function, or module of any other name by which modules are known in other programming languages.
	<i>process</i>	A series of actions or steps taken to achieve a particular outcome. [Oxford]
	<i>program; programming</i>	program (n): A set of instructions that the computer executes to achieve a particular objective. [MDESE, 2016] program (v): To produce a program by programming. programming: The craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them. [MDESE, 2016]
	<i>protocol</i>	The special set of rules used by endpoints in a telecommunication connection when they communicate. Protocols specify interactions between the communicating entities. [TechTarget]
	<i>prototype</i>	An early approximation of a final product or information system, often built for demonstration purposes. [TechTarget, Techopedia]
R		
	<i>redundancy</i>	A system design in which a component is duplicated, so if it fails, there will be a backup. [TechTarget]
	<i>reliability</i>	An attribute of any system that consistently produces the same results, preferably meeting or exceeding its requirements. [FOLDDOC]
	<i>remix</i>	The process of creating something new from something old. Originally a process that involved music, remixing involves creating a new version of a program by recombining and modifying parts of existing programs, and often adding new pieces, to form new solutions. [Kafai & Burke, 2014]
	<i>router</i>	A device or software that determines the path that data packets travel from source to destination. [TechTarget]

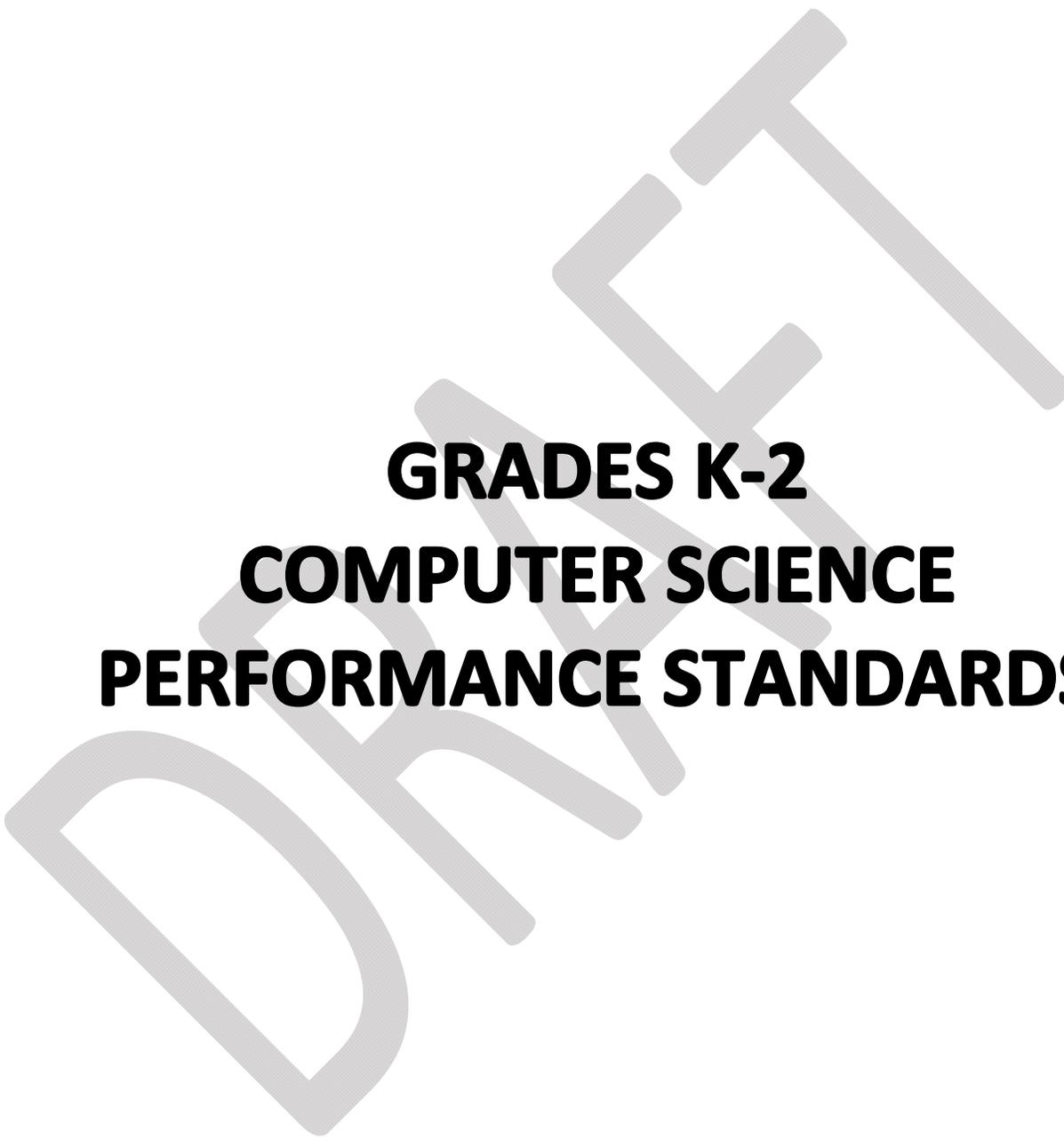
S	
<i>scalability</i>	The capability of a network to handle a growing amount of work or its potential to be enlarged to accommodate that growth. [Wikipedia]
<i>security</i>	See the definition for cybersecurity.
<i>simulate; simulation</i>	simulate: To imitate the operation of a real-world process or system. simulation: Imitation of the operation of a real-world process or system. [MDESE, 2016]
<i>software</i>	Programs that run on a computing system, computer, or other computing device.
<i>storage</i>	(place) A place, usually a device, into which data can be entered, in which the data can be held, and from which the data can be retrieved at a later time. [FOLDOC] (process) A process through which digital data is saved within a data storage device by means of computing technology. Storage is a mechanism that enables a computer to retain data, either temporarily or permanently. [Techopedia]
<i>string</i>	A sequence of letters, numbers, and/or other symbols. A string might represent, for example, a name, address, or song title. Some functions commonly associated with strings are length, concatenation, and substring. [TechTarget]
<i>structure</i>	A general term used in the framework to discuss the concept of encapsulation without specifying a particular programming methodology.
<i>switch</i>	A high-speed device that receives incoming data packets and redirects them to their destination on a local area network (LAN). [Techopedia]
<i>system</i>	A collection of elements or components that work together for a common purpose. [TechTarget] See also the definition for computing system.
T	
<i>test case</i>	A set of conditions or variables under which a tester will determine whether the system being tested satisfies requirements or works correctly. [STF]
<i>topology</i>	The physical and logical configuration of a network; the arrangement of a network, including its nodes and connecting links. A logical topology is the way devices appear connected to the user. A physical topology is the way they are actually interconnected with wires and cables. [PCMag]
<i>troubleshooting</i>	A systematic approach to problem solving that is often used to find and resolve a problem, error, or fault within software or a computing system. [Techopedia, TechTarget]
U	
<i>user</i>	See the definition for end user.
V	
<i>variable</i>	A symbolic name that is used to keep track of a value that can change while a program is running. Variables are not just used for numbers; they can also hold text, including whole sentences (strings) or logical values (true or false). A variable has a data type and is associated with a data storage location; its value is normally changed during the course of program execution. [CAS, 2013; Techopedia] <i>Note: This definition differs from that used in math.</i>

References - Some definitions came directly from these sources, while others were excerpted or adapted to include content relevant to this framework.

- **K-12 Computer Science Framework** <https://k12cs.org/glossary/>
- **A Model Curriculum for K–12 Computer Science** Tucker, A., McCowan, D., Deek, F., Stephenson, C., Jones, J., & Verno, A. (2006). *A model curriculum for K–12 computer science: Report of the ACM K–12 task force curriculum committee* (2nd ed.). New York, NY: Association for Computing Machinery.
- **Computing At School's Computing in the National Curriculum: A Guide for Primary Teachers** Computing At School. (2013). *Computing in the national curriculum: A guide for primary teachers*. Belford, UK: Newnorth Print. <https://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf>
- **College Board Advanced Placement® Computer Science Principles** College Board. (2016). *AP Computer Science Principles course and exam description*. New York, NY: College Board. <https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf>
- **Free On-Line Dictionary of Computing** Free on-line dictionary of computing. (n.d.). Retrieved from <https://foldoc.org>
- **The Free Dictionary** The free dictionary. (n.d.). Retrieved from <https://www.thefreedictionary.com>
- **Connected Code: Why Children Need to Learn Programming** Kafai, Y., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. Cambridge, MA: MIT Press.
- **Reclaiming the Roots of CT** Lee, I. (2016). *Reclaiming the roots of CT*. *CSTA Voice: The Voice of K–12 Computer Science Education and Its Educators*, 12(1), 3–4. https://www.csteachers.org/resource/resmgr/Voice/csta_voice_03_2016.pdf
- **Massachusetts Digital Literacy and Computer Science (DL&CS) Standards** Massachusetts Department of Elementary and Secondary Education. (2016, June). *2016 Massachusetts digital literacy and computer science (DLCS) curriculum framework*. Malden, MA: Author. <https://www.doe.mass.edu/frameworks/dlcs.pdf>
- **College, Career & Civic Life (C3) Framework for Social Studies State Standards** National Council for the Social Studies. (2013). *The college, career, and civic life (C3) framework for social studies state standards: Guidance for enhancing the rigor of K–12 civics, economics, geography, and history*. Silver Spring, MD: Author. <https://www.socialstudies.org/system/files/c3/C3-Framework-for-Social-Studies.pdf>
- **National Institute of Science and Technology Dictionary of Algorithms and Data Structures** Pieterse, V., & Black, P. E. (Eds.). (n.d). *Dictionary of algorithms and data structures*. Retrieved from <https://xlinux.nist.gov/dads>
- **Oxford Dictionaries** Oxford dictionaries. (n.d.). Retrieved from <https://www.oxforddictionaries.com/us>
- **PCmag.com Encyclopedia** PCmag.com encyclopedia. (n.d.). Retrieved from <https://www.pcmag.com/encyclopedia/term/46301/logical-vs-physical-topology>
- **What Is Automation** Ross, B. (2016, May 10). *What is automation and how can it improve customer service?* *Information Age*. Retrieved from <https://www.information-age.com/industry/software/123461408/what-automation-and-how-can-it-improve-customer-service>
- **Software Testing Fundamentals** Software testing fundamentals. (n.d). Retrieved from <https://softwaretestingfundamentals.com>
- **Tech Terms** Tech terms computer dictionary. (n.d.). Retrieved from <https://www.techterms.com>
- **Techopedia** Techopedia technology dictionary. (n.d.). Retrieved from <https://www.techopedia.com/dictionary>
- **TechTarget Network** TechTarget network. (n.d.). Retrieved from <https://www.techtarget.com/network>
- **Webopedia** Webopedia. (n.d.). Retrieved from <https://www.webopedia.com>
- **Wikipedia** Wikipedia: The free encyclopedia. (n.d.). Retrieved from <https://www.wikipedia.org/>



**GRADES K-12
COMPUTER SCIENCE
PERFORMANCE STANDARDS**



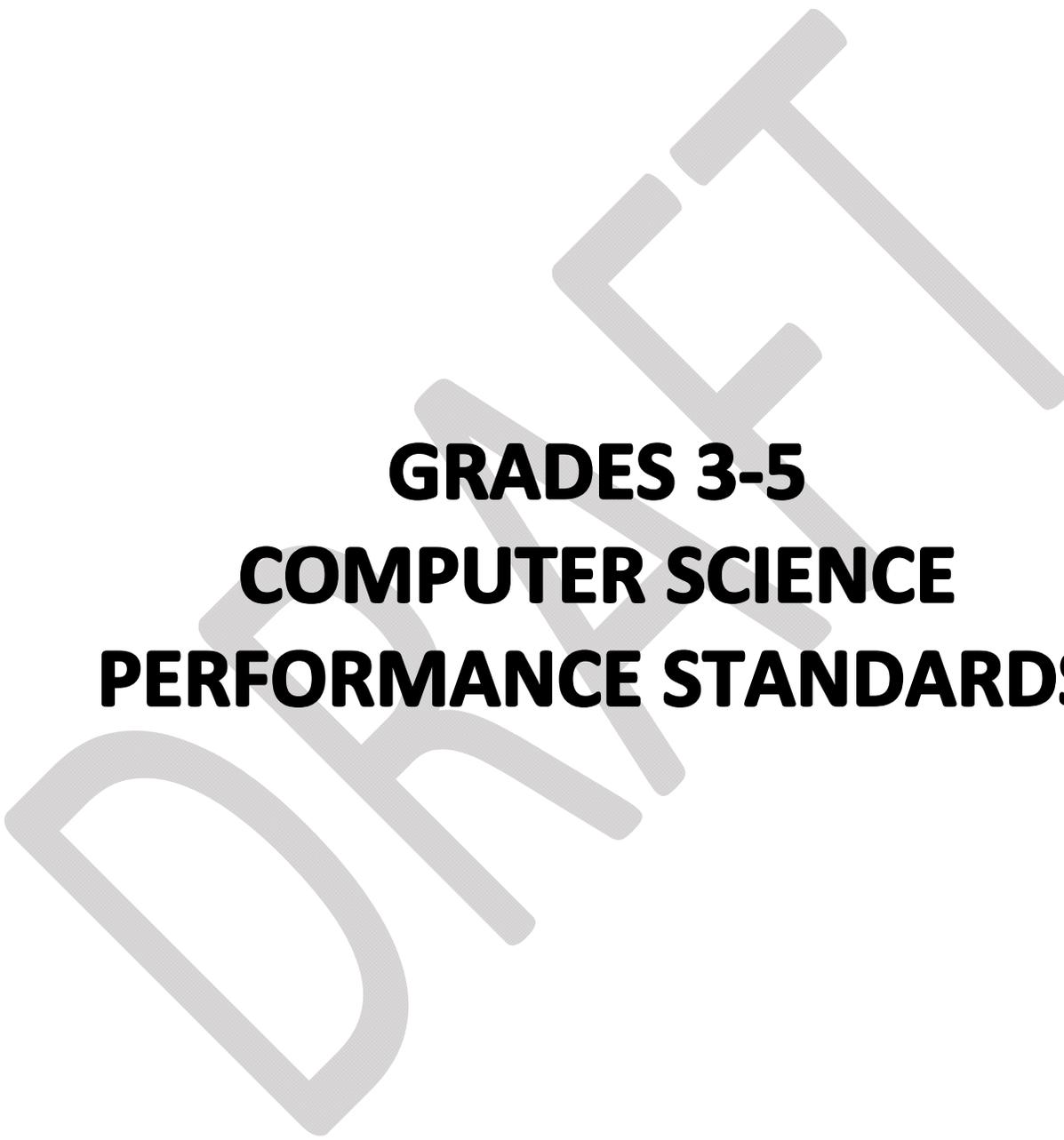
**GRADES K-2
COMPUTER SCIENCE
PERFORMANCE STANDARDS**

GRADES K-2 COMPUTER SCIENCE PERFORMANCE STANDARDS

Concept	Subconcept	Kindergarten	1st Grade	2nd Grade
Computing Systems	Devices	K.CS.D.01 With guidance, follow directions and make appropriate choices to use computing devices to perform a variety of tasks.	1.CS.D.01 With guidance, select and use a computing device to perform a variety of tasks for an intended outcome.	2.CS.D.01 Select and use a computing device to perform a variety of tasks for an intended outcome.
	Hardware & Software	K.CS.H.01 Use appropriate terminology in naming and describing the function of common computing devices and components (e.g., mouse is used to control the cursor, desktop computer, laptop computer, tablet device, monitor, keyboard, mouse, printer).	1.CS.H.01 Use appropriate terminology to locate and identify common computing devices and components, in a variety of environments (e.g., desktop computer, laptop computer, tablet device, monitor, keyboard, mouse, printer).	2.CS.H.01 Identify the components of a computer system and what the basic functions are (e.g., hard drive and memory) as well as peripherals (e.g., printers, scanners, external hard drives) and external storage features and their uses (e.g., cloud storage).
		K.CS.S.02 With guidance, choose appropriate software to perform a variety of tasks.	1.CS.S.02 With little support, choose appropriate software to perform a variety of tasks.	2.CS.S.02 Independently choose appropriate software to perform a variety of tasks.
	Troubleshooting	K.CS.T.01 Recognize that computing systems might not work as expected and learn to use accurate terminology to identify simple hardware or software problems (e.g., volume turned down on headphones, monitor turned off, keyboard not working, mouse not working).	1.CS.T.01 Identify, using accurate terminology, simple hardware and software problems that may occur during use (e.g., app or program is not working as expected, no sound is coming from the device, caps lock turned on).	2.CS.T.01 Identify using accurate terminology, simple hardware and software problems that may occur during use (e.g., app or program is not working as expected, no sound is coming from the device, caps lock turned on) and discuss problems with peers and adults.
Networks & the Internet	Network Communication & Organization	K.NI.NCO.01 Discuss that computing devices can be connected together. (e.g., printers connect to devices, phone/tablet share information).	1.NI.NCO.01 Recognize that by connecting computing devices together they can share information (e.g., remote storage, printing, the internet).	2.NI.NCO.01 Recognize that computing devices can be connected at various scales (e.g., Bluetooth, Wi-Fi, hotspot, LAN, WAN, peer-to-peer).
	Cybersecurity	K.NI.C.01 Discuss what passwords are and why we do not share them with others. With guidance, use passwords to access technological devices, apps, etc.	1.NI.C.01 Identify what passwords are and explain why they are not shared. Discuss what makes a password strong. Independently, use passwords to access technological devices, apps, etc.	2.NI.C.01 Recognize what passwords are and why we do not share them. Explain why we use them and why we use strong passwords to protect devices and information from unauthorized access.
Data & Analysis	Storage	K.DA.S.01 With guidance, locate, open, modify and save an existing file with a computing device.	1.DA.S.01 With guidance locate, open, modify and save an existing file, use appropriate file-naming conventions and recognize that the file exists within an organizational structure (e.g., drive, folder, file).	2.DA.S.01 With guidance, create, copy, locate, modify and delete a file on a computing device, use appropriate file-naming conventions and recognize that the file exists within an organizational structure (e.g., drive, folder, file) - define the information stored as data.

Concept	Subconcept	Kindergarten	1st Grade	2nd Grade
Data & Analysis	Collection, Visualization & Transformation	K.DA.CVT.01 With guidance, collect data and present it visually.	1.DA.CVT.01 With guidance, collect data and present it two different ways.	2.DA.CVT.01 With guidance, collect and present the same data in various visual formats.
	Inference & Models	K.DA.IM.01 With guidance, draw conclusions and make predictions based on picture graphs or patterns (e.g., make predictions based on weather data presented in a picture graph or complete a pattern).	1.DA.IM.01 With guidance, identify and interpret data from a chart or graph (visualization) in order to make a prediction, with or without a computing device.	2.DA.IM.01 With guidance, construct and interpret data and present it in a chart or graph (visualization) in order to make a prediction, with or without a computing device.
Algorithms & Programming	Algorithms	K.AP.A.01 With guidance, model daily processes and follow algorithms (sets of step-by-step instructions) to complete tasks verbally, kinesthetically, with robot devices or a programming language.	1.AP.A.01 With guidance, model daily processes and follow algorithms (sets of step-by-step instructions) to complete tasks verbally, kinesthetically, with robot devices or a programming language.	2.AP.A.01 With guidance, model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks verbally, kinesthetically, with robot devices or a programming language.
	Variables	K.AP.V.01 With guidance, recognize that computers represent different types of data using numbers or other symbols.	1.AP.V.01 With guidance, model the way that a program accesses stored data using a variable name.	2.AP.V.01 Model the way a computer program manipulates grade level appropriate data (e.g., print, numbers, kinesthetic movement, symbols, robot manipulatives).
	Control	K.AP.C.01 With guidance, independently or collaboratively create programs to accomplish tasks using a programming language, robot device or unplugged activity that includes sequencing (i.e., emphasizing the beginning, middle and end).	1.AP.C.01 With guidance, independently or collaboratively create programs to accomplish tasks using a programming language, robot device or unplugged activity that includes sequencing and repetition.	2.AP.C.01 With guidance, create programs using a programming language, robot device or unplugged activity that utilize sequencing and simple looping to solve a problem or express ideas both independently and collaboratively.
	Program Development	K.AP.PD.01 With guidance, create a grade-level appropriate artifact to illustrate thoughts, ideas or sequence of events (step-by-step) manner (e.g., story map, storyboard, sequential graphic organizer).	1.AP.PD.01 Independently or with guidance, create a grade level appropriate document of the plan, ideas and sequence of events (step-by- step) manner (e.g., story map, storyboard, sequential graphic organizer) to illustrate what the program will do.	2.AP.PD.01 Independently or with guidance, create a grade level appropriate document of the plan, ideas and sequence of events (step-by- step) manner (e.g., story map, storyboard, sequential graphic organizer) to illustrate what the program will do.
		K.AP.PD.02 Independently or with guidance give credit to ideas, creations and solutions of others while developing algorithms.	1.AP.PD.02 Independently or with guidance give credit to ideas, creations and solutions of others while writing and/or developing programs.	2.AP.PD.02 Give credit to ideas, information, creations and solutions of others while writing and developing programs.

Concept	Subconcept	Kindergarten	1st Grade	2nd Grade
Algorithms & Programming	Program Development	K.AP.PD.03 With guidance, independently or collaboratively debug algorithms using a programming language and/or unplugged activity that includes sequencing.	1.AP.PD.03 With guidance, independently or collaboratively debug programs using a programming language and/or unplugged activity that includes sequencing and simple loops.	2.AP.PD.03 Independently and collaboratively, debug programs, which include sequencing and simple loops, to accomplish tasks as a means of creative expression or problem solving using a programming language and/or unplugged activities.
		K.AP.PD.04 Use correct terminology (beginning, middle, end) in the development of an algorithm to solve a simple problem.	1.AP.PD.04 Use correct terminology (first, second, third) and explain the choices made in the development of an algorithm to solve a simple problem.	2.AP.PD.04 Use correct terminology (e.g., debug, program input/output, code) to explain the development of an algorithm to solve a problem in an unplugged activity, hands on manipulatives or a programming language.
Impacts of Computing	Culture	K.IC.C.01 Discuss different ways in which types of technologies are used in daily life.	1.IC.C.01 Identify how people use different types of technologies in their daily work and personal lives.	2.IC.C.01 Identify and describe how people use many types of technologies in their daily work and personal lives.
	Social Interactions	K.IC.SI.01 With guidance, identify appropriate manners while participating in an online environment and online behaviors.	1.IC.SI.01 With guidance, identify appropriate and inappropriate behavior. Act responsibly while participating in an online community and know how to report concerns of cyberbullying.	2.IC.SI.01 Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior and know how to report concerns of cyberbullying.
	Safety, Law & Ethics	K.IC.SLE.01 Exhibit good digital citizenship using technology safely, responsibly and ethically.	1.IC.SLE.01 Work respectfully and responsibly with others online. Learn what information that is put online is appropriate and can start a digital footprint.	2.IC.SLE.01 Identify safe and unsafe examples of online communications. Learn that the information put online leaves a digital footprint.



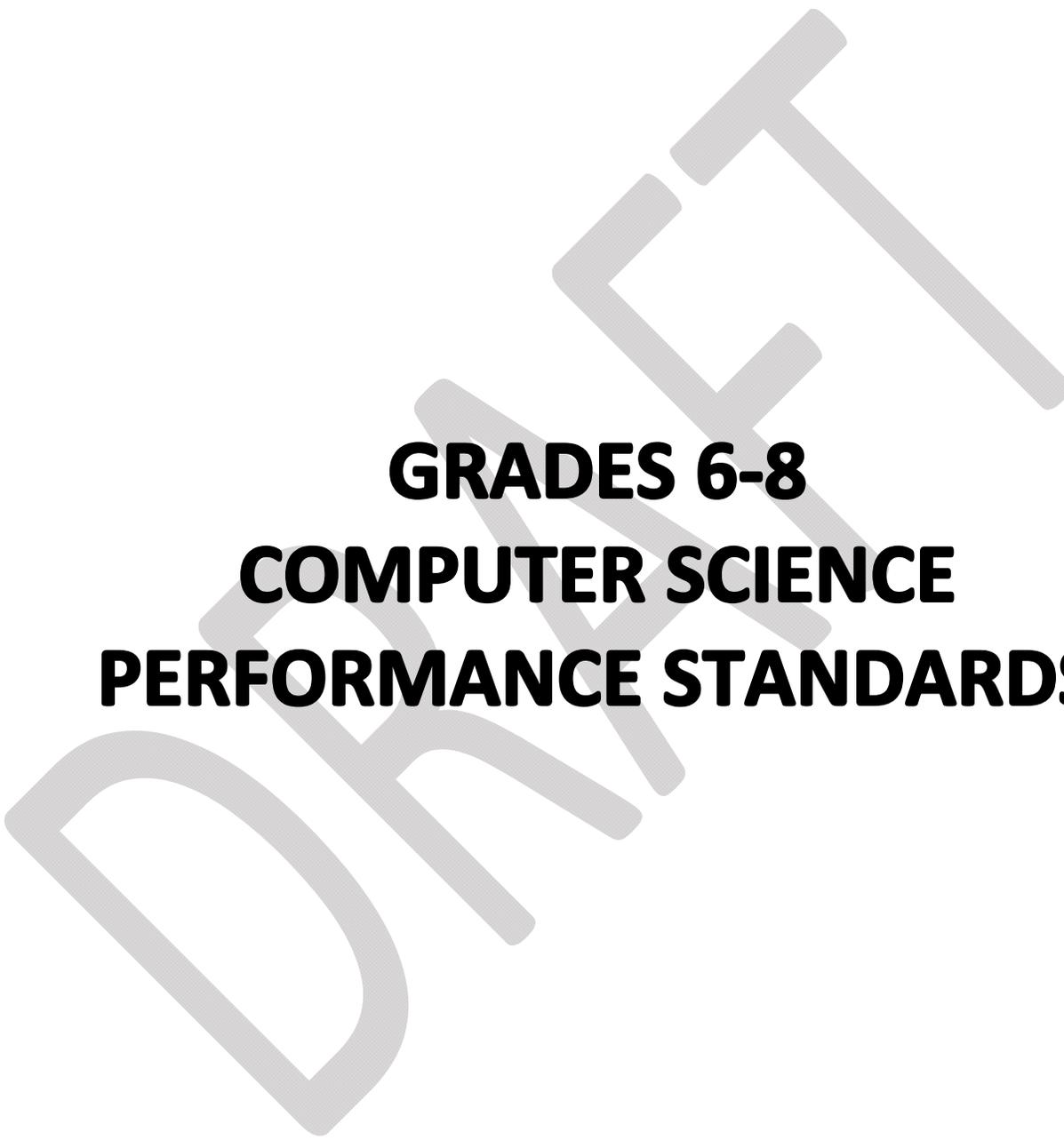
**GRADES 3-5
COMPUTER SCIENCE
PERFORMANCE STANDARDS**

GRADES 3-5 COMPUTER SCIENCE PERFORMANCE STANDARDS

Concept	Subconcept	3rd Grade	4th Grade	5th Grade
Computing Systems	Hardware & Software	3.CS.HS.01 Model how information flows through hardware and software to accomplish tasks.	4.CS.HS.01 Model that information is translated, transmitted and processed in order to flow through hardware and software.	5.CS.HS.01 Model that information is translated into bits in order to transmit and process between software to accomplish tasks.
	Troubleshooting	3.CS.T.01 Identify, using accurate terminology, simple hardware and software problems that may occur during everyday use, discuss problems with peers and adults and apply strategies for solving these problems (e.g., refresh the screen, closing and reopening an application or file, unmuting or adjusting the volume on headphones).	4.CS.T.01 Identify, using accurate terminology, simple hardware and software problems that may occur during everyday use, discuss problems with peers and adults and apply strategies for solving these problems (e.g., rebooting the computing device, checking the power, force shut down of an application).	5.CS.T.01 Identify, using accurate terminology, simple hardware and software problems that may occur during everyday use. Discuss problems with peers and adults, apply strategies for solving these problems and explain why the strategy should work.
Networks & the Internet	Network Communication & Organization	3.NI.NCO.01 Recognize how information changes when sent and received over physical or wireless paths. (Information is broken into smaller parts, sent to the destination and then reassembled into a whole.)	4.NI.NCO.01 Explain how information is broken down into packets, transmitted through multiple computing devices over networks and the internet and reassembled at the destination.	5.NI.NCO.01 Model how information is broken down into packets, transmitted through multiple computing devices over networks and the internet and reassembled at the destination.
	Cybersecurity	3.NI.C.01 Identify problems that relate to inappropriate use of computing devices and networks.	4.NI.C.01 Discuss real-world cybersecurity problems and identify strategies for how personal information can be protected.	5.NI.C.01 Analyze the credibility of digital information (e.g., comparing multiple accounts and sources, the author's point of view).
		5.NI.C.02 Discuss cybersecurity problems caused by information that is published for different reasons (e.g., inform, advertise, persuade, harm).		
Data & Analysis	Storage	3.DA.S.01 Recognize that different types of information are stored in different formats that have associated programs (e.g., documents open in a word processor) and varied storage requirements.	4.DA.S.01 Choose different storage locations (e.g., physical, shared, cloud) based on the type of file, storage requirements (e.g., file size, availability, available memory) and sharing requirements.	5.DA.S.01 Evaluate trade-offs, including availability and quality, based on the type of file, storage requirements (e.g., file size, availability, available memory) and sharing requirements.
	Collection, Visualization & Transformation	3.DA.CVT.01 Collect data using various programs and formats (e.g., surveys, forms) and organize the data in various visual formats (e.g., charts, graphs, tables).	4.DA.CVT.01 Organize and present collected data visually to highlight comparisons.	5.DA.CVT.01 Organize and present collected data to highlight comparisons and support a claim.

Concept	Subconcept	3rd Grade	4th Grade	5th Grade
Data & Analysis	Inference & Models	3.DA.IM.01 With guidance, utilize data to make predictions and discuss whether there is adequate data to be useful and to make reliable predictions.	4.DA.IM.01 Determine how the accuracy of conclusions are influenced by the amount of useful and reliable data collected.	5.DA.IM.01 Use reliable data to highlight or propose cause and effect relationships, predict outcomes or communicate an idea.
Algorithms & Programming	Algorithms	3.AP.A.01 Compare multiple algorithms (sets of step-by-step instructions) for accomplishing the same task verbally and kinesthetically, with robot devices or a programming language.	4.AP.A.01 Compare and simplify multiple algorithms (sets of step-by-step instructions) for accomplishing the same task verbally and kinesthetically, with robot devices or a programming language.	5.AP.A.01 Compare and simplify multiple algorithms (sets of step-by-step instructions) for accomplishing the same task verbally and kinesthetically, with robot devices or a programming language, then determine which is the most efficient.
	Variables	3.A.V.01 Create programs that use variables to store and modify grade level appropriate data.	4.AP.V.01 Create programs that use variables to store and modify grade level appropriate data.	5.AP.V.01 Create programs that use variables to store and modify grade level appropriate data.
	Control	3.AP.C.01 Collaboratively create a program using control structures (e.g., sequence, conditionals, interactive-looping) to make decisions within a program.	4.AP.C.01 Create a program using control structures (e.g., sequence, conditionals, interactive-looping) to solve a problem or express ideas both independently and collaboratively.	5.AP.C.01 Create a program using control structures (e.g., sequence, conditionals, interactive-looping), event handlers and variables to solve a problem or express ideas both independently and collaboratively.
	Modularity	3.AP.M.01 Decompose (break down) the steps needed to solve a problem into precise sequence of instructions.	4.AP.M.01 Decompose (break down) large problems into smaller, manageable sub problems to facilitate the program development process.	5.AP.M.01 Decompose (break down) large problems into smaller, manageable sub problems and then into a precise sequence of instructions.
		3.AP.M.02 With grade appropriate complexity, modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	4.AP.M.02 With grade appropriate complexity, modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	5.AP.M.02 With grade appropriate complexity, modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
	Program Development	3.AP.PD.01 Use an iterative and collaborative process to plan the development of a program while solving simple problems.	4.AP.PD.01 Use an iterative and collaborative process to plan the development of a program that includes user preferences while solving simple problems.	5.AP.PD.01 Use an iterative and collaborative process to plan the development of a program that includes other perspectives and user preferences while solving simple problems.
		3.AP.PD.02 Observe intellectual property rights and give appropriate credit when creating or remixing programs.	4.AP.PD.02 Observe intellectual property rights and give appropriate credit when creating or remixing programs.	5.AP.PD.02 Observe intellectual property rights and give appropriate credit when creating or remixing programs.

Concept	Subconcept	3rd Grade	4th Grade	5th Grade
Algorithms & Programming	Program Development	3.AP.PD.03 Analyze and debug a program that includes sequencing, repetition and variables in a programming language.	4.AP.PD.03 Analyze, create and debug a program that includes sequencing, repetition, conditionals and variables in a programming language.	5.AP.PD.03 Analyze, examine, create and debug a program that includes sequencing, repetition, conditionals and variables in a programming language.
		3.AP.PD.04 Communicate and explain your program development using comments, presentations and interactive demonstrations.	4.AP.PD.04 Communicate and explain your program development using comments, presentations and interactive demonstrations.	5.AP.PD.04 Communicate and explain your program development using comments, presentations and interactive demonstrations.
Impacts of Computing	Culture	3.IC.C.01 Identify computing technologies that have changed the world and express how those technologies influence, and are influenced by, cultural practices.	4.IC.C.01 Give examples of computing technologies that have changed the world and express how those technologies influence, and are influenced by, cultural practices.	5.IC.C.01 Give examples and explain how computing technologies have changed the world and express how computing technologies influence, and are influenced by, cultural practices.
		3.IC.C.02 Identify possible problems and how computing devices have built in features for increasing accessibility to all users.	4.IC.C.02 Brainstorm problems and ways to improve computing devices to increase accessibility to all users.	5.IC.C.02 Develop, test and refine digital artifacts to improve accessibility and usability.
	Social Interactions	3.IC.SI.01 Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community (e.g., responsibilities of being a good digital citizen, private and personal information, showing respect for other people's work). Identify and report inappropriate behavior and know how to report cyberbullying.	4.IC.SI.01 Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community (e.g., using strong passwords, creating a positive online community, recognizing spam and what to do about it, citing online sources). Identify and report inappropriate behavior and know how to report cyberbullying.	5.IC.SI.01 Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community (e.g., talking safely online, promoting good digital citizens, privacy settings, cyberbullying). Identify and report inappropriate behavior and know how to report cyberbullying.
		Safety, Law & Ethics	3.IC.SLE.01 Identify types of digital data that may have intellectual property rights that prevent copying or require attribution.	4.IC.SLE.01 Discuss the social impact of violating intellectual property rights.
	3.IC.SLE.02 Discuss the importance of a positive digital footprint.		4.IC.SLE.02 Discuss and understand the implications of a negative digital footprint.	5.IC.SLE.02 Continue to discuss and understand the implications of positive and negative digital footprints and that they never go away.



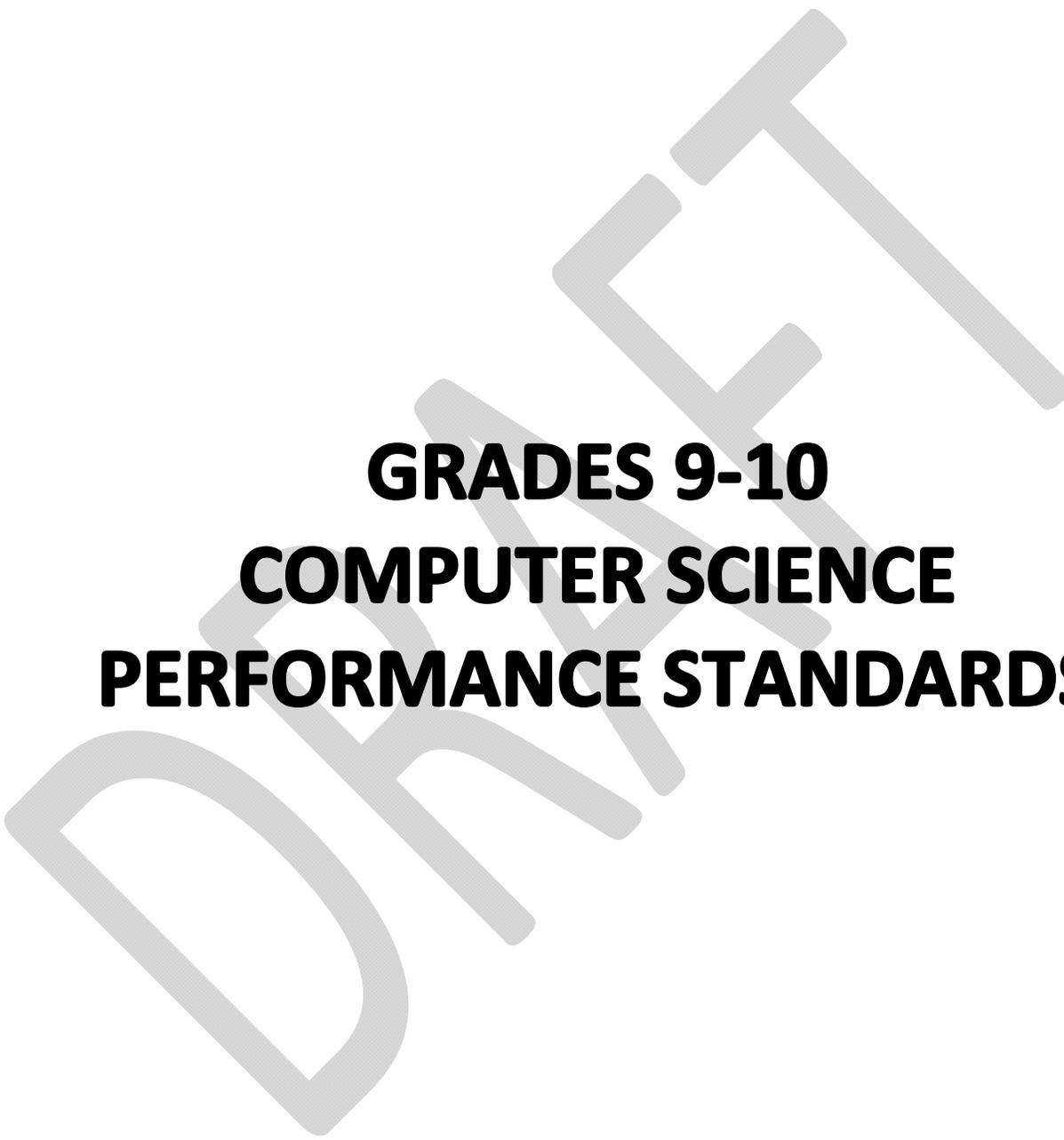
**GRADES 6-8
COMPUTER SCIENCE
PERFORMANCE STANDARDS**

GRADES 6-8 COMPUTER SCIENCE PERFORMANCE STANDARDS

Concept	Subconcept	By the End of the 8th Grade	
Computing Systems	Devices	6-8.CS.D.01 Evaluate the design of computing devices, based on the characteristics of each device and how users interact with it, to improve the overall user experience.	
	Hardware & Software	6-8.CS.HS.01 Design projects that combine hardware and software to collect and exchange data.	
	Troubleshooting	6-8.CS.T.01 Develop a systematic troubleshooting routine to identify the problem, research solutions and fix problems with computing devices, components and software.	
Networks & the Internet	Network Communication & Organization	6-8.NI.NCO.01 Model the different ways that data is transferred across a network and the protocols used to transmit the data.	
	Cybersecurity	6-8.NI.C.01 Recognize and determine computer threats and be able to identify programs and methods to protect electronic information.	6-8.NI.C.02 Demonstrate how data is transmitted through multiple methods of encryption.
Data & Analysis	Storage	6-8.DA.S.01 Represent data using multiple encoding schemes.	
	Collection, Visualization & Transformation	6-8.DA.VT.01 Collect data using computational tools and display it for the end user in an easy to understand way.	
Algorithms & Programming	Inference & Models	6-8.DA.IM.01 Analyze methods to refine computational models based on received data.	
	Algorithms	6-8.AP.A.01 Design algorithms with flow charts and/or pseudocode to show solutions to complex problems.	
	Variables	6-8.AP.V.01 Create clearly named variables to store and manipulate information.	
	Control	6-8.AP.C.01 Design and develop combinations of control structures, nested loops and compound conditionals.	
	Modularity	6-8.AP.M.01 Decompose problems and subproblems into parts to facilitate the design, implementation and review of programs.	6-8.AP.M.02 Create procedures with parameters to organize code and to make it easier to reuse.
	Program Development	6-8.AP.PD.01 Use flowcharts and/or pseudocode to solve problems using algorithms.	6-8.AP.PD.02 Use feedback from team members and users to refine solutions to meet user needs.
6-8.AP.PD.03 Give proper attribution to code, media, etc. that is used in their programs.		6-8.AP.PD.04 Test and refine programs using a range of test cases.	
6-8.AP.PD.05 Manage project tasks and timelines when collaboratively developing computational artifacts.			

Concept	Subconcept	By the End of the 8 th Grade	
Impacts of Computing	Culture	6-8.IC.C.01 Compare tradeoffs associated with computing technologies that have impacted people's activities, careers and lives when solving global problems using the power of computing.	6-8.IC.C.02 Discuss issues of bias and accessibility in the design of existing technologies.
	Social Interaction	6-8.IC.SI.01 Collaborate through strategies such as crowdsourcing or surveys when creating a computational artifact.	
	Safety, Law & Ethics	6-8.IC.SLE.01 Describe tradeoffs between allowing information to be public and keeping information private and secure.	

DRAFT

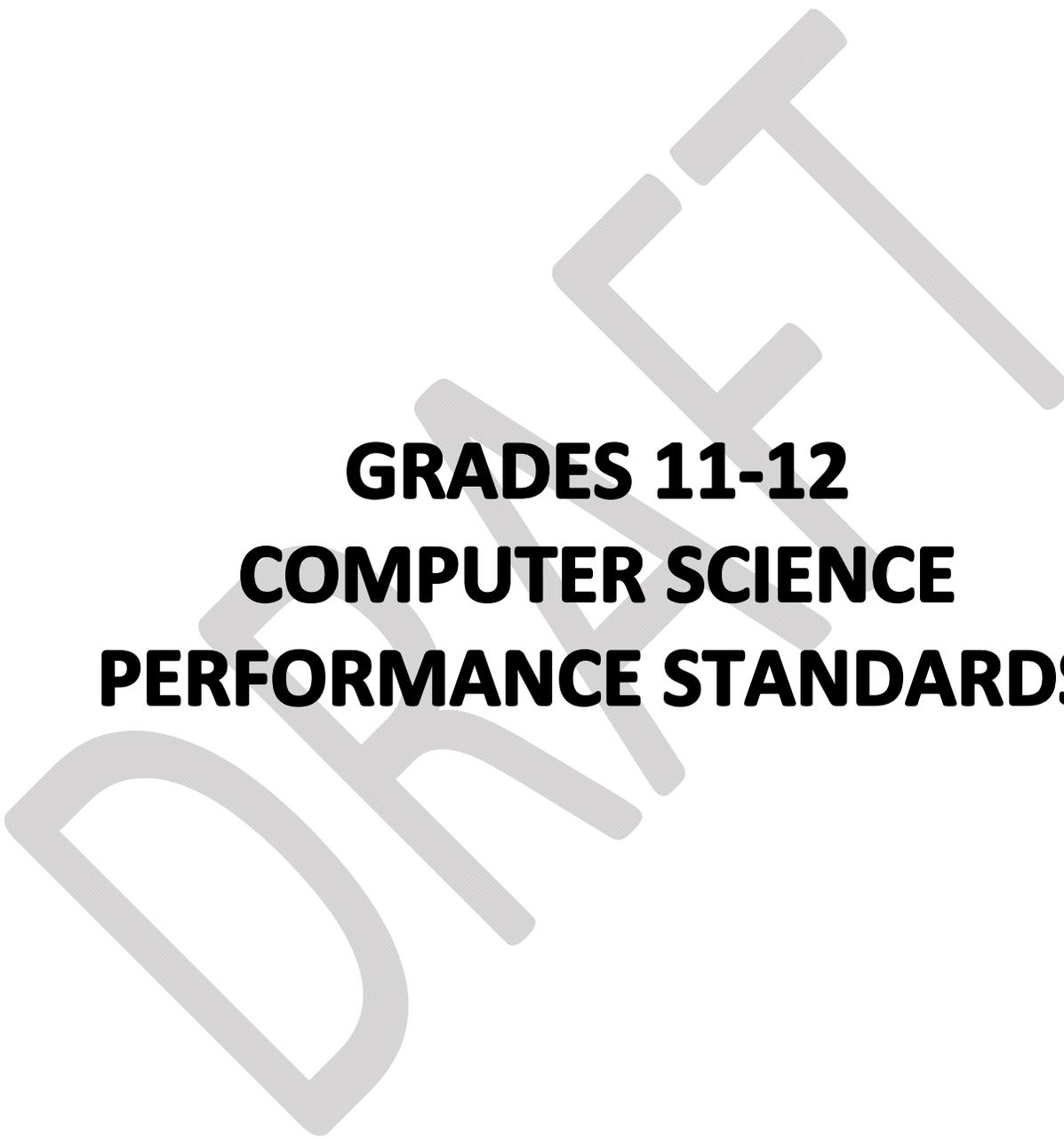


**GRADES 9-10
COMPUTER SCIENCE
PERFORMANCE STANDARDS**

GRADES 9-10 COMPUTER SCIENCE PERFORMANCE STANDARDS

Concept	Subconcept	By the End of the 10th Grade	
Computing Systems	Devices	9-10.CS.D.01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	
	Hardware & Software	9-10.CS.HS.01 Explain the abstraction and interactions between application software, system software and hardware.	
	Troubleshooting	9-10.CS.T.01 Develop, communicate and apply systematic troubleshooting strategies for correction of errors in computing systems.	
Networks and the Internet	Network Communication & Organization	9-10.NI.NCO.01 Evaluate the scalability and reliability of networks by identifying and illustrating the basic components of computer networks (e.g., routers, switches, servers) and network protocols (e.g., IP, DNS).	9-10.NI.NCO.02 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
	Cybersecurity	9-10.NI.C.01 Compare physical and cybersecurity measures by evaluating trade-offs between the usability and security of a computing system.	9-10.NI.C.02 Illustrate how sensitive data can be affected by attacks.
		9-10.NI.C.03 Recommend security measures to address various scenarios based on information security principles.	9-10.NI.C.04 Explain trade-offs when selecting and implementing cybersecurity recommendations from multiple perspectives such as the user, enterprise and government.
Data Analysis	Storage	9-10.DA.S.01 Translate and compare different bit representations of data types, such as characters, numbers and images.	9-10.DA.S.02 Evaluate the trade-offs in how data is organized and stored digitally.
	Collection, Visualization & Transformation	9-10.DA.CVT.01 Create data visualizations to help others better understand real-world phenomena.	9-10.DA.CVT.02 Explain the insights and knowledge gained from digitally processed data by using appropriate visualizations, notions and precise language.
		9-10.DA.CVT.03 Evaluate and refine computational artifacts to make them more usable and accessible.	
	Inference & Models	9-10.DA.IM.01 Show the relationships between collected data elements using computational models.	9-10.DA.IM.02 Refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.
Algorithms & Programming	Algorithms	9-10.AP.A.01 Create a prototype that uses algorithms (e.g., searching, sorting, finding shortest distance) to provide a possible solution for a real-world problem.	
	Variables	9-10.AP.V.01 Create problem solutions that utilize primitive variables (e.g., strings, ints, Booleans, doubles).	9-10.AP.V.02 Demonstrate the use of advanced variables (e.g., lists, arrays, objects) to simplify solutions, generalizing computational problems instead of repeatedly using primitive variables.

Concept	Subconcept	By the End of the 10th Grade	
Algorithms & Programming	Control	9-10.AP.C.01 Apply the concepts of specific control structures (e.g., sequence, conditionals, repetition, procedures) considering program efficiencies such as readability, performance and memory usage.	
	Modularity	9-10.AP.M.01 Break down a solution into procedures using systematic analysis and design utilizing functional abstraction.	9-10.AP.M.02 Create computational artifacts (file, graphic, video, audio) by systematically organizing, manipulating and/or processing data.
	Program Development	9-10.AP.PD.01 Using visual aids and documentation, illustrate the design elements and data flow (e.g., flowcharts, pseudocode) of the development of a program.	9-10.AP.PD.02 Create a program by analyzing a problem and/or process, developing and documenting a solution, testing outcomes, debugging errors and adapting the program for a variety of users.
		9-10.AP.PD.04 While collaborating in a team, develop, test and refine programs that solve practical problems or allow self-expression.	9-10.AP.PD.05 Evaluate and refine computational artifacts to make them more user-friendly, efficient and/or accessible.
Impacts of Computing	Culture	9-10.IC.C.01 Evaluate the ways computing impacts personal, ethical, social, economic and cultural practices.	9-10.IC.C.02 Test and refine computational artifacts to reduce bias and equity deficits.
		9-10.IC.C.03 Demonstrate how a given algorithm applies to problems across disciplines.	
	Social Interactions	9-10.IC.SI.01 Demonstrate through collaboration on a project how computing increases connectivity among people of various cultures.	9-10.IC.SI.02 Explain how the degrees of communication afforded by computing have impacted the nature and content of career fields.
	Safety, Laws & Ethics	9-10.IC.SLE.01 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	9-10.IC.SLE.02 Explain the privacy concerns related to the collection and analysis of information about individuals that may not be evident to users.
		9-10.IC.SLE.03 Evaluate the social and economic consequences of how law and ethics interact with digital aspects of privacy, data, property, information and identity.	9-10.AP.SLE.04 Define and classify a variety of software licensing schemes (e.g., open source, freeware, commercial) and discuss the advantages and disadvantages of each scheme in software development.
9-10.IC.SLE.05 Identify and explain the potential impacts and implications of emerging technologies on larger social economic and political structures with evidence from credible sources.			



**GRADES 11-12
COMPUTER SCIENCE
PERFORMANCE STANDARDS**

GRADES 11-12 COMPUTER SCIENCE PERFORMANCE STANDARDS

Concept		Subconcept		By the End of the 12th Grade	
Computing Systems	Devices	11-12.CS.D.01 Illustrate ways computing systems implement logic through hardware components.			
	Hardware & Software	11-12.CS.HS.01 Describe and categorize roles of an operating system.			
	Troubleshooting	11-12.CS.T.01 Describe how hardware components facilitate logic, input, output and storage in computing systems.			
Networks & the Internet	Network Communication & Organization	11-12.NI.NCO.01 Analyze the relationship between routers, switches, servers, topology and addressing.		11-12.NI.NCO.02 Describe key protocols and underlying processes of internet-based services (e.g., http/https and Simple Mail Transfer Protocol (SMTP)/internet Message Access Protocol (IMAP), routing protocols).	
		11-12.NI.NCO.03 Explain how the characteristics of the internet influence the systems developed on it.			
	Cybersecurity	11-12.NI.C.01 Compare and refine ways in which software developers protect devices and information from unauthorized access.		11-12.NI.C.02 Analyze cryptographic techniques to model the secure transmission of information.	
Data & Analysis	Storage	11-12.DA.S.01 Compare different bit representations of data types, such as characters, Booleans and numbers while recognizing when using each data type is appropriate.			
	Collection, Visualization & Transformation	11-12.DA.CVT.01 Generate data sets that use a variety of data collection tools and analysis techniques to support a claim and/or communicate information.			
	Inference & Models	11-12.DA.IM.01 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.			
Algorithms & Programming	Algorithms	11-12.AP.A.01 Critically examine and trace classic algorithms (e.g., selection sort, insertion sort, binary search, linear search).		11-12.AP.A.02 Implement an artificial intelligence algorithm to interact with a human or solve a problem.	
		11-12.AP.A.03 Describe how artificial intelligence algorithms drive many software and physical systems (e.g., autonomous robots, computer vision, pattern recognition, text analysis).		11-12.AP.A.04 Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency and clarity.	
	Variables	11-12.AP.V.01 Create problem solutions that utilize data structures (e.g., lists, arrays, ArrayLists).			
	Control	11-12.AP.C.01 Trace the execution of iteration (e.g., loops, recursion), illustrating output and changes in values of named variables.			
	Modularity	11-12.AP.M.01 Construct solutions to problems using student-created components (e.g., procedures, modules, objects).		11-12.AP.M.02 Create programming solutions by reusing existing code (e.g., libraries, Application Programming Interface (APIs), code repositories).	
11-12.AP.M.03 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.					

Concept	Subconcept	By the End of the 12 th Grade	
Algorithms & Programming	Program Development	11-12.AP.PD.01 Use integrated development environments (IDEs) and collaborative tools and practices (code documentation) in a software project.	11-12.AP.PD.02 Plan and develop programs using a development process (e.g., waterfall, iterative, spiral, rapid application development, agile).
		11-12.AP.PD.03 Identify and compare features of various programming languages that make them useful for solving problems and developing systems.	11-12.AP.PD.04 Design software using version control.
		11-12.AP.PD.05 Develop and use a series of test cases to verify that a program performs according to its design specifications.	11-12.AP.PD.06 Explain security issues that might lead to compromised computer programs.
		11-12.AP.PD.07 Evaluate key qualities of a program through a process such as a code review.	
Impacts of Computing	Culture	11-12.IC.C.01 Evaluate the impact of equity, access and influence on the distribution of computing resources in a global society.	
	Safety, Laws & Ethics	11-12.IC.SLE.01 Debate laws and regulations that impact the development and use of software.	

DRAFT

GRADES K-12

COMPUTER SCIENCE

PERFORMANCE STANDARDS

April 11, 2019

Changes made in response to comments received are in RED

**GRADES K-2
COMPUTER SCIENCE
PERFORMANCE STANDARDS**

GRADES K-2 COMPUTER SCIENCE PERFORMANCE STANDARDS

Concept	Subconcept	Kindergarten	1st Grade	2nd Grade
Computing Systems	Devices	K.CS.D.01 With guidance, follow directions and make appropriate choices to use computing devices to perform a variety of tasks.	1.CS.D.01 With guidance, select and use a computing device to perform a variety of tasks for an intended outcome.	2.CS.D.01 Select and use a computing device to perform a variety of tasks for an intended outcome.
	Hardware	K.CS.H.01 Use appropriate terminology in naming and describing the function of common computing devices and components (e.g., mouse is used to control the cursor, desktop computer, laptop computer, tablet device, monitor, keyboard, mouse, printer).	1.CS.H.01 Use appropriate terminology to locate and identify common computing devices and components, in a variety of environments (e.g., desktop computer, laptop computer, tablet device, monitor, keyboard, mouse, printer).	2.CS.H.01 Identify the components of a computer system and what the basic functions are (e.g., hard drive and memory) as well as peripherals (e.g., printers, scanners, external hard drives) and external storage features and their uses (e.g., cloud storage).
	Software	K.CS.S.02 With guidance, choose appropriate software to perform a variety of tasks.	1.CS.S.02 With little support, choose appropriate software to perform a variety of tasks.	2.CS.S.02 Independently choose appropriate software to perform a variety of tasks.
	Troubleshooting	K.CS.T.01 Recognize that computing systems might not work as expected and learn to use accurate terminology to identify simple hardware or software problems (e.g., volume turned down on headphones, monitor turned off, keyboard not working, mouse not working).	1.CS.T.01 Identify, using accurate terminology, simple hardware and software problems that may occur during use (e.g., app or program is not working as expected, no sound is coming from the device, caps lock turned on).	2.CS.T.01 Identify using accurate terminology, simple hardware and software problems that may occur during use (e.g., app or program is not working as expected, no sound is coming from the device, caps lock turned on) and discuss problems with peers and adults.
Networks & the Internet	Network Communication & Organization	K.NI.NCO.01 Discuss that computing devices can be connected together. (e.g., printers connect to devices, phone/tablets share information).	1.NI.NCO.01 Recognize that by connecting computing devices together they can share information (e.g., remote storage, printing, the internet).	2.NI.NCO.01 Recognize that computing devices can be connected at various scales (Bluetooth, Wi-Fi, hotspot, LAN, WAN, peer-to-peer).
	Cybersecurity	K.NI.C.01 Discuss what passwords are and why we do not share them with others. With guidance, use passwords to access technological devices, apps, etc.	1.NI.C.01 Identify what passwords are and explain why they are not shared. Discuss what makes a password strong. Independently, use passwords to access technological devices, apps, etc.	2.NI.C.01 Recognize what passwords are and why we do not share them. Explain why we use them and why we use strong passwords to protect devices and information from unauthorized access.
Data & Analysis	Storage	K.DA.S.01 With guidance, locate, open, modify and save an existing file with a computing device.	1.DA.S.01 With guidance locate, open, modify and save an existing file, use appropriate file-naming conventions and recognize that the file exists within an organizational structure (drive, folder, file).	2.DA.S.01 With guidance, create, copy, locate, modify and delete a file on a computing device, use appropriate file-naming conventions and recognize that the file exists within an organizational structure (drive, folder, file) - define the information stored as data.

Concept	Subconcept	Kindergarten	1st Grade	2nd Grade
Data & Analysis	Collection, Visualization & Transformation	K.DA.CVT.01 With guidance, collect data and present it visually.	1.DA.CVT.01 With guidance, collect data and present it two different ways.	2.DA.CVT.01 With guidance, collect and present the same data in various visual formats.
	Inference & Models	K.DA.IM.01 With guidance, draw conclusions and make predictions based on picture graphs or patterns (e.g., make predictions based on weather data presented in a picture graph or complete a pattern).	1.DA.IM.01 With guidance, identify and interpret data from a chart or graph (visualization) in order to make a prediction, with or without a computing device.	2.DA.IM.01 With guidance, construct and interpret data and present it in a chart or graph (visualization) in order to make a prediction, with or without a computing device.
Algorithms & Programming	Algorithms	K.AP.A.01 With guidance, model daily processes and follow algorithms (sets of step-by-step instructions) to complete tasks verbally, kinesthetically, with robot devices or a programming language.	1.AP.A.01 With guidance, model daily processes and follow algorithms (sets of step-by-step instructions) to complete tasks verbally, kinesthetically, with robot devices or a programming language.	2.AP.A.01 With guidance, model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks verbally, kinesthetically, with robot devices or a programming language.
	Variables	K.AP.V.01 With guidance, recognize that computers represent different types of data using numbers or other symbols.	1.AP.V.01 With guidance, model the way that a program accesses stored data using a variable name.	2.AP.V.01 Model the way a computer program manipulates grade level appropriate data (e.g., print, numbers, kinesthetic movement, symbols and robot manipulatives).
	Control	K.AP.C.01 With guidance, independently or collaboratively create programs to accomplish tasks using a programming language, robot device or unplugged activity that includes sequencing (e.g., emphasizing the beginning, middle and end).	1.AP.C.01 With guidance, independently or collaboratively create programs to accomplish tasks using a programming language, robot device or unplugged activity that includes sequencing and repetition.	2.AP.C.01 With guidance, create programs using a programming language, robot device or unplugged activity that utilize sequencing and simple looping to solve a problem or express ideas both independently and collaboratively.
	Program Development	K.AP.PD.01 With guidance, create a grade-level appropriate artifact to illustrate thoughts, ideas or sequence of events (step-by-step) manner (e.g., story map, storyboard and sequential graphic organizer).	1.AP.PD.01 Independently or with guidance, create a grade level appropriate document of the plan, ideas and sequence of events (step-by- step) manner (e.g., story map, storyboard and sequential graphic organizer) to illustrate what the program will do.	2.AP.PD.01 Independently or with guidance, create a grade level appropriate document of the plan, ideas and sequence of events (step-by- step) manner (e.g., story map, storyboard and sequential graphic organizer) to illustrate what the program will do.
		K.AP.PD.02 Independently or with guidance give credit to ideas, creations and solutions of others while developing algorithms.	1.AP.PD.02 Independently or with guidance give credit to ideas, creations and solutions of others while writing and/or developing programs.	2.AP.PD.02 Give credit to ideas, information, creations and solutions of others while writing and developing programs.

Concept	Subconcept	Kindergarten	1st Grade	2nd Grade
Algorithms & Programming	Program Development	K.AP.PD.03 With guidance, independently or collaboratively debug algorithms using a programming language and/or unplugged activity that includes sequencing.	1.AP.PD.03 With guidance, independently or collaboratively debug programs using a programming language and/or unplugged activity that includes sequencing and simple loops.	2.AP.PD.03 Independently and collaboratively, debug programs, which include sequencing and simple loops, to accomplish tasks as a means of creative expression or problem solving using a programming language and/or unplugged activities.
		K.AP.PD.04 Use correct terminology (beginning, middle, end) in the development of an algorithm to solve a simple problem.	1.AP.PD.04 Use correct terminology (first, second, third) and explain the choices made in the development of an algorithm to solve a simple problem.	2.AP.PD.04 Use correct terminology (e.g., debug, program input/output, code) to explain the development of an algorithm to solve a problem in an unplugged activity, hands on manipulatives or a programming language.
Impacts of Computing	Culture	K.IC.C.01 Discuss different ways in which types of technologies are used in daily life.	1.IC.C.01 Identify how people use different types of technologies in their daily work and personal lives	2.IC.C.01 Identify and describe how people use many types of technologies in their daily work and personal lives.
	Social Interactions	K.IC.SI.01 With guidance, identify appropriate manners while participating in an online environment and online behaviors.	1.IC.SI.01 With guidance, identify appropriate and inappropriate behavior. Act responsibly while participating in an online community and know how to report concerns of cyberbullying.	2.IC.SI.01 Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community. Identify and report inappropriate behavior and know how to report concerns of cyberbullying.
	Safety, Law & Ethics	K.IC.SLE.01 Exhibit good digital citizenship using technology safely, responsibly and ethically.	1.IC.SLE.01 Work respectfully and responsibly with others online. Learn what information that is put online is appropriate and can start a digital footprint.	2.IC.SLE.01 Identify safe and unsafe examples of online communications. Learn that the information put online leaves a digital footprint.

GRADES 3-5
COMPUTER SCIENCE
PERFORMANCE STANDARDS

GRADES 3-5 COMPUTER SCIENCE PERFORMANCE STANDARDS

Concept	Subconcept	3rd Grade	4th Grade	5th Grade
Computing Systems	Hardware & Software	3.CS.HS.01 Model how information flows through hardware and software to accomplish tasks.	4.CS.HS.01 Model that information is translated, transmitted and processed in order to flow through hardware and software.	5.CS.HS.01 Model that information is translated into bits in order to transmit and process between software to accomplish tasks.
	Troubleshooting	3.CS.T.01 Identify, using accurate terminology, simple hardware and software problems that may occur during everyday use, discuss problems with peers and adults and apply strategies for solving these problems (e.g., refresh the screen, closing and reopening an application or file, unmuting or adjusting the volume on headphones).	4.CS.T.01 Identify, using accurate terminology, simple hardware and software problems that may occur during everyday use, discuss problems with peers and adults and apply strategies for solving these problems (e.g., rebooting the computing device, checking the power, force shut down of an application).	5.CS.T.01 Identify, using accurate terminology, simple hardware and software problems that may occur during everyday use. Discuss problems with peers and adults, apply strategies for solving these problems and explain why the strategy should work.
Networks & the Internet	Network Communication & Organization	3.NI.NCO.01 Recognize how information changes when sent and received over physical or wireless paths. (Information is broken into smaller parts, sent to the destination and then reassembled into a whole.)	4.NI.NCO.01 Explain how information is broken down into packets, transmitted through multiple computing devices over networks and the Internet and reassembled at the destination.	5.NI.NCO.01 Model how information is broken down into packets, transmitted through multiple computing devices over networks and the Internet and reassembled at the destination.
	Cybersecurity	3.NI.C.01 Identify problems that relate to inappropriate use of computing devices and networks.	4.NI.C.01 Discuss real-world cybersecurity problems and identify strategies for how personal information can be protected.	5.NI.C.01 Analyze the credibility of digital information (comparing multiple accounts and sources, the author's point of view).
				5.NI.C.02 Discuss cybersecurity problems caused by information that is published for different reasons (inform, advertise, persuade, harm).
Data & Analysis	Storage	3.DA.S.01 Recognize that different types of information are stored in different formats that have associated programs (e.g., documents open in a word processor) and varied storage requirements.	4.DA.S.01 Choose different storage locations (physical, shared or cloud) based on the type of file, storage requirements (file size, availability, available memory) and sharing requirements.	5.DA.S.01 Evaluate trade-offs, including availability and quality, based on the type of file, storage requirements (file size, availability, available memory) and sharing requirements.

Concept	Subconcept	3rd Grade	4th Grade	5th Grade
Data & Analysis	Collection, Visualization & Transformation	3.DA.CVT.01 Collect data using various programs and formats (surveys, forms) and organize the data in various visual formats (charts, graphs, tables).	4.DA.CVT.01 Organize and present collected data visually to highlight comparisons.	5.DA.CVT.01 Organize and present collected data to highlight comparisons and support a claim.
	Inference & Models	3.DA.IM.01 With guidance, utilize data to make predictions and discuss whether there is adequate data to be useful and to make reliable predictions.	4.DA.IM.01 Determine how the accuracy of conclusions are influenced by the amount of useful and reliable data collected.	5.DA.IM.01 Use reliable data to highlight or propose cause and effect relationships, predict outcomes or communicate an idea.
Algorithms & Programming	Algorithms	3.AP.A.01 Compare multiple algorithms (sets of step-by-step instructions) for accomplishing the same task, verbally and kinesthetically, with robot devices or a programming language.	4.AP.A.01 Compare and simplify multiple algorithms (sets of step-by-step instructions) for accomplishing the same task, verbally and kinesthetically, with robot devices or a programming language.	5.AP.A.01 Compare and simplify multiple algorithms (sets of step-by-step instructions) for accomplishing the same task, verbally and kinesthetically, with robot devices or a programming language then determine which is the most efficient.
	Variables	3.A.V.01 Create programs that use variables to store and modify grade level appropriate data.	4.AP.V.01 Create programs that use variables to store and modify grade level appropriate data.	5.AP.V.01 Create programs that use variables to store and modify grade level appropriate data.
	Control	3.AP.C.01 Collaboratively create a program using control structures (sequence, conditionals, interactive-looping) to make decisions within a program.	4.AP.C.01 Create a program using control structures (sequence, conditionals, interactive-looping) to solve a problem or express ideas both independently and collaboratively.	5.AP.C.01 Create a program using control structures (sequence, conditionals, interactive-looping), event handlers and variables to solve a problem or express ideas both independently and collaboratively.
	Modularity	3.AP.M.01 Decompose (break down) the steps needed to solve a problem into precise sequence of instructions.	4.AP.M.01 Decompose (break down) large problems into smaller, manageable sub problems to facilitate the program development process.	5.AP.M.01 Decompose (break down) large problems into smaller, manageable sub problems and then into a precise sequence of instructions.
		3.AP.M.02 With grade appropriate complexity, modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	4.AP.M.02 With grade appropriate complexity, modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	5.AP.M.02 With grade appropriate complexity, modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
	Program Development	3.AP.PD.01 Use an iterative and collaborative process to plan the development of a program while solving simple problems.	4.AP.PD.01 Use an iterative and collaborative process to plan the development of a program that includes user preferences while solving simple problems.	5.AP.PD.01 Use an iterative and collaborative process to plan the development of a program that includes others' perspectives and user preferences while solving simple problems.

Concept	Subconcept	3rd Grade	4th Grade	5th Grade
Algorithms & Programming	Program Development	3.AP.PD.02 Observe intellectual property rights and give appropriate credit when creating or remixing programs.	4.AP.PD.02 Observe intellectual property rights and give appropriate credit when creating or remixing programs.	5.AP.PD.02 Observe intellectual property rights and give appropriate credit when creating or remixing programs.
		3.AP.PD.03 Analyze and debug a program that includes sequencing, repetition and variables in a programming language.	4.AP.PD.03 Analyze, create and debug a program that includes sequencing, repetition, conditionals and variables in a programming language.	5.AP.PD.03 Analyze, examine, create and debug a program that includes sequencing, repetition, conditionals and variables in a programming language.
		3.AP.PD.04 Communicate and explain your program development using comments, presentations and interactive demonstrations.	4.AP.PD.04 Communicate and explain your program development using comments, presentations and interactive demonstrations.	5.AP.PD.04 Communicate and explain your program development using comments, presentations and interactive demonstrations.
Impacts of Computing	Culture	3.IC.C.01 Identify computing technologies that have changed the world and express how those technologies influence, and are influenced by, cultural practices.	4.IC.C.01 Give examples of computing technologies that have changed the world and express how those technologies influence, and are influenced by, cultural practices.	5.IC.C.01 Give examples and explain how computing technologies have changed the world and express how computing technologies influence, and are influenced by, cultural practices.
		3.IC.C.02 Identify possible problems and how computing devices have built in features for increasing accessibility to all users.	4.IC.C.02 Brainstorm problems and ways to improve computing devices to increase accessibility to all users.	5.IC.C.02 Develop, test and refine digital artifacts to improve accessibility and usability.
	Social Interactions	3.IC.SI.01 Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community. (responsibilities of being a good digital citizen, private and personal information and showing respect for other people's work) Identify and report inappropriate behavior and know how to report cyberbullying.	4.IC.SI.01 Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community (Using strong passwords, creating a positive online community, recognizing spam and what to do about it, citing online sources). Identify and report inappropriate behavior and know how to report cyberbullying.	5.IC.SI.01 Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community. (Talking safely online, promoting good digital citizens, privacy settings and cyberbullying) Identify and report inappropriate behavior and know how to report cyberbullying.
Impacts of Computing	Social Interactions	3.IC.SI.02 Identify how computational products may be, or have been, improved to incorporate diverse perspectives.	4.IC.SI.02 As a team, consider each others' perspectives on improving a computational product.	5.IC.SI.02 As a team, collaborate with outside resources, other grade levels or online collaborative spaces, to include diverse perspectives to improve computational products.

Concept	Subconcept	3rd Grade	4th Grade	5th Grade
Impacts of Computing	Safety, Law & Ethics	3.IC.SLE.01 Identify types of digital data that may have intellectual property rights that prevent copying or require attribution.	4.IC.SLE.01 Discuss the social impact of violating intellectual property rights.	5.IC.SLE.01 Observe intellectual property rights and give appropriate credit when using resources.
		3.IC.SLE.02 Discuss the importance of a positive digital footprint.	4.IC.SLE.02 Discuss and understand the implications of a negative digital footprint.	5. IC.SLE.02 Continue to discuss and understand the implications of positive and negative digital footprints and that they never go away.

**GRADES 6-8
COMPUTER SCIENCE
PERFORMANCE STANDARDS**

GRADES 6-8 COMPUTER SCIENCE PERFORMANCE STANDARDS

Concept	Subconcept	By the End of the 8th Grade	
Computing Systems	Devices	6-8.CS.D.01 Evaluate the design of computing devices, based on the characteristics of each device and how users interact with it, to improve the overall user experience.	
	Hardware & Software	6-8.CS.HS.01 Design projects that combine hardware and software to collect and exchange data.	
	Troubleshooting	6-8.CS.T.01 Develop a systematic troubleshooting routine to identify the problem, research solutions and fix problems with computing devices, components and software.	
Networks & the Internet	Network Communication & Organization	6-8.NI.NCO.01 Model the different ways that data is transferred across a network and the protocols used to transmit the data.	
	Cybersecurity	6-8.NI.C.01 Recognize and determine computer threats and be able to identify programs and methods to protect electronic information.	6-8.NI.C.02 Demonstrate how data is transmitted through multiple methods of encryption.
Data & Analysis	Storage	6-8.DA.S.01 Represent data using multiple encoding schemes.	
	Collection, Visualization & Transformation	6-8.DA.VT.01 Collect data using computational tools and display it for the end user in an easy to understand way.	
Algorithms & Programming	Inference & Models	6-8.DA.IM.01 Analyze methods to refine computational models based on received data.	
	Algorithms	6-8.AP.A.01 Design algorithms with flow charts and/or pseudocode to show solutions to complex problems.	
	Variables	6-8.AP.V.01 Create clearly named variables to store and manipulate information.	
	Control	6-8.AP.C.01 Design and develop combinations of control structures, nested loops and compound conditionals.	
	Modularity	6-8.AP.M.01 Decompose problems and subproblems into parts to facilitate the design, implementation and review of programs.	6-8.AP.M.02 Create procedures with parameters to organize code and to make it easier to reuse.
	Program Development	6-8.AP.PD.01 Use flowcharts and/or pseudocode to solve problems using algorithms.	6-8.AP.PD.02 Use feedback from team members and users to refine solutions to meet user needs.
6-8.AP.PD.03 Give proper attribution to code, media, etc. that is used in their programs.		6-8.AP.PD.04 Test and refine programs using a range of test cases.	
6-8.AP.PD.05 Manage project tasks and timelines when collaboratively developing computational artifacts.			

Concept	Subconcept	By the End of the 8 th Grade	
Impacts of Computing	Culture	6-8.IC.C.01 Compare tradeoffs associated with computing technologies that have impacted people's activities, careers, and lives when solving global problems using the power of computing.	6-8.IC.C.02 Discuss issues of bias and accessibility in the design of existing technologies.
	Social Interaction	6-8.IC.SI.01 Collaborate through strategies such as crowdsourcing or surveys when creating a computational artifact.	
	Safety, Law & Ethics	6-8.IC.SLE.01 Describe tradeoffs between allowing information to be public and keeping information private and secure.	

**GRADES 9-10
COMPUTER SCIENCE
PERFORMANCE STANDARDS**

GRADES 9-10 COMPUTER SCIENCE PERFORMANCE STANDARDS

Concept	Subconcept	By the End of the 10th Grade	
Computing Systems	Devices	9-10.CS.D.01 Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	
	Hardware & Software	9-10.CS.HS.01 Explain the abstraction and interactions between application software, system software and hardware.	
	Troubleshooting	9-10.CS.T.01 Develop, communicate and apply systematic troubleshooting strategies for correction of errors in computing systems.	
Networks and the Internet	Network Communication & Organization	9-10.NI.NCO.01 Evaluate the scalability and reliability of networks by identifying and illustrating the basic components of computer networks (e.g., routers, switches, servers, etc.) and network protocols (e.g., IP, DNS, etc.).	9-10.NI.NCO.02 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology).
	Cybersecurity	9-10.NI.C.01 Compare physical and cybersecurity measures by evaluating trade-offs between the usability and security of a computing system.	9-10.NI.C.02 Illustrate how sensitive data can be affected by attacks.
		9-10.NI.C.03 Recommend security measures to address various scenarios based on information security principles.	9-10.NI.C.04 Explain trade-offs when selecting and implementing cybersecurity recommendations from multiple perspectives such as the user, enterprise and government.
Data Analysis	Storage	9-10.DA.S.01 Translate and compare different bit representations of data types, such as characters, numbers and images.	9-10.DA.S.02 Evaluate the trade-offs in how data is organized and stored digitally.
	Collection, Visualization & Transformation	9-10.DA.CVT.01 Create data visualizations to help others better understand real-world phenomena.	9-10.DA.CVT.02 Explain the insights and knowledge gained from digitally processed data by using appropriate visualizations, notions and precise language.
		9-10.DA.CVT.03 Evaluate and refine computational artifacts to make them more usable and accessible.	
	Inference & Models	9-10.DA.IM.01 Show the relationships between collected data elements using computational models.	9-10.DA.IM.02 Refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.
Algorithms & Programming	Algorithms	9-10.AP.A.01 Create a prototype that uses algorithms (e.g., searching, sorting, finding shortest distance) to provide a possible solution for a real-world problem.	
	Variables	9-10.AP.V.01 Create problem solutions that utilize primitive variables (e.g., strings, ints, booleans, doubles, etc.).	9-10.AP.V.02 Demonstrate the use of advanced variables (e.g., lists, arrays, objects) to simplify solutions, generalizing computational problems instead of repeatedly using primitive variables.

Concept	Subconcept	By the End of the 10th Grade	
Algorithms & Programming	Control	9-10.AP.C.01 Apply the concepts of specific control structures (e.g., sequence, conditionals, repetition, procedures) considering program efficiencies such as readability, performance and memory usage.	
	Modularity	9-10.AP.M.01 Break down a solution into procedures using systematic analysis and design utilizing functional abstraction.	9-10.AP.M.02 Create computational artifacts (file, graphic, video, audio, etc.) by systematically organizing, manipulating and/or processing data.
	Program Development	9-10.AP.PD.01 Using visual aids and documentation, illustrate the design elements and data flow (e.g., flowcharts, pseudocode) of the development of a program.	9-10.AP.PD.02 Create a program by analyzing a problem and/or process, developing and documenting a solution, testing outcomes, debugging errors and adapting the program for a variety of users.
		9-10.AP.PD.04 While collaborating in a team, develop, test and refine programs that solve practical problems or allow self-expression.	9-10.AP.PD.05 Evaluate and refine computational artifacts to make them more user-friendly, efficient and/or accessible.
Impacts of Computing	Culture	9-10.IC.C.01 Evaluate the ways computing impacts personal, ethical, social, economic and cultural practices.	9-10.IC.C.02 Test and refine computational artifacts to reduce bias and equity deficits.
		9-10.IC.C.03 Demonstrate how a given algorithm applies to problems across disciplines.	
	Social Interactions	9-10.IC.SI.01 Demonstrate through collaboration on a project how computing increases connectivity among people of various cultures.	9-10.IC.SI.02 Explain how the degrees of communication afforded by computing have impacted the nature and content of career fields.
	Safety, Laws & Ethics	9-10.IC.SLE.01 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.	9-10.IC.SLE.02 Explain the privacy concerns related to the collection and analysis of information about individuals that may not be evident to users.
		9-10.IC.SLE.03 Evaluate the social and economic consequences of how law and ethics interact with digital aspects of privacy, data, property, information and identity.	9-10.AP.SLE.04 Define and classify a variety of software licensing schemes (e.g., open source, freeware, commercial) and discuss the advantages and disadvantages of each scheme in software development.
		9-10.IC.SLE.05 Identify and explain the potential impacts and implications of emerging technologies on larger social economic and political structures with evidence from credible sources.	

GRADES 11-12
COMPUTER SCIENCE
PERFORMANCE STANDARDS

GRADES 11-12 COMPUTER SCIENCE PERFORMANCE STANDARDS

Concept	Subconcept	By the End of the 12th Grade	
Computing Systems	Devices	11-12.CS.D.01 Illustrate ways computing systems implement logic through hardware components.	
	Hardware & Software	11-12.CS.HS.01 Describe and categorize roles of an operating system.	
	Troubleshooting	11-12.CS.T.01 Describe how hardware components facilitate logic, input, output and storage in computing systems.	
Networks & the Internet	Network Communication & Organization	11-12.NI.NCO.01 Analyze the relationship between routers, switches, servers, topology and addressing.	11-12.NI.NCO.02 Describe key protocols and underlying processes of internet-based services (e.g., http/https and Simple Mail Transfer Protocol (SMTP)/Internet Message Access Protocol (IMAP), routing protocols).
		11-12.NI.NCO.03 Explain how the characteristics of the internet influence the systems developed on it.	
	Cybersecurity	11-12.NI.C.01 Compare and refine ways in which software developers protect devices and information from unauthorized access.	11-12.NI.C.02 Analyze cryptographic techniques to model the secure transmission of information.
Data & Analysis	Storage	11-12.DA.S.01 Compare different bit representations of data types, such as characters, booleans and numbers while recognizing when using each data type is appropriate.	
	Collection, Visualization & Transformation	11-12.DA.CVT.01 Generate data sets that use a variety of data collection tools and analysis techniques to support a claim and/or communicate information.	
	Inference & Models	11-12.DA.IM.01 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.	
Algorithms & Programming	Algorithms	11-12.AP.A.01 Critically examine and trace classic algorithms (e.g., selection sort, insertion sort, binary search, linear search).	11-12.AP.A.02 Implement an artificial intelligence algorithm to interact with a human or solve a problem.
		11-12.AP.A.03 Describe how artificial intelligence algorithms drive many software and physical systems (e.g., autonomous robots, computer vision, pattern recognition, text analysis).	11-12.AP.A.04 Evaluate algorithms (e.g., sorting, searching) in terms of their efficiency and clarity.
	Variables	11-12.AP.V.01 Create problem solutions that utilize data structures (e.g., lists, arrays, ArrayLists).	
	Control	11-12.AP.C.01 Trace the execution of iteration (loops or recursion), illustrating output and changes in values of named variables.	
	Modularity	11-12.AP.M.01 Construct solutions to problems using student-created components (e.g., procedures, modules, objects).	11-12.AP.M.02 Create programming solutions by reusing existing code (e.g., libraries, Application Programming Interface (APIs), code repositories).

Concept	Subconcept	By the End of the 12th Grade	
Algorithms & Programming	Modularity	11-12.AP.M.03 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.	
	Program Development	11-12.AP.PD.01 Use integrated development environments (IDEs) and collaborative tools and practices (code documentation) in a software project.	11-12.AP.PD.02 Plan and develop programs using a development process (e.g., waterfall, iterative, spiral, rapid application development and agile).
		11-12.AP.PD.03 Identify and compare features of various programming languages that make them useful for solving problems and developing systems.	11-12.AP.PD.04 Design software using version control.
		11-12.AP.PD.05 Develop and use a series of test cases to verify that a program performs according to its design specifications.	11-12.AP.PD.06 Explain security issues that might lead to compromised computer programs.
		11-12.AP.PD.07 Evaluate key qualities of a program through a process such as a code review.	
Impacts of Computing	Culture	11-12.IC.C.01 Evaluate the impact of equity, access and influence on the distribution of computing resources in a global society.	
	Safety, Laws & Ethics	11-12.IC.SLE.01 Debate laws and regulations that impact the development and use of software.	

GRADES K-12

COMPUTER SCIENCE

PERFORMANCE STANDARDS

Original draft language submitted
Amendments made
Rationale for change

Revised 4/18/2019

WORKGROUP AMENDMENTS

	STANDARD #	ORIGINAL STANDARD	AMENDED STANDARD	RATIONALE/COMMENTS
1	3.AP.A.01	Compare multiple algorithms for the same task.	Compare multiple algorithms (sets of step-by-step instructions) for accomplishing the same task verbally, kinesthetically, with robot devices or a programming language.	The standard has been clarified to identify its intended purpose. This also explains that algorithms do not need to occur only in a programming language. The workgroup decided on this change.
2	3.AP.C.01	Create programs using a programming language that utilize sequencing, repetition, conditionals and variables to solve a problem or express ideas both independently and collaboratively.	Collaboratively create a program using control structures (sequence, conditionals, interactive-looping) to make decisions within a program.	Public comment recommended the emphasis of control structures which programming languages use to make decisions first.
3	3.AP.PD.01	Use an iterative process to plan the development of a program while solving simple problems.	Use an iterative and collaborative process to plan the development of a program while solving simple problems.	After reviewing public comment, the decision to include more collaborative practices throughout the 3rd-5th grade was made.
4	3.AP.PD.03	Analyze and debug a program that includes sequencing, repetition and variables in a programming language (e.g., Java, Java Script, Python, Blockly, etc.).	Analyze and debug a program that includes sequencing, repetition and variables in a programming language.	After considering public comment, the decision to remove the examples of programming language was made.
5	3.IC.SI.01	Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community (e.g., responsibilities of being a good digital citizen, private and personal information and showing respect for other people's work). Identify and report inappropriate behavior.	Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community (e.g., responsibilities of being a good digital citizen, private and personal information, showing respect for other people's work). Identify and report inappropriate behavior and know how to report cyberbullying.	Needed to better align with K-2
6	3.IC.SI.02	Identify how computational products may be, or have been, improved to incorporate diverse perspectives.	This standard was deleted.	Just added a layer of confusion, not needed. The workgroup decided on this change.
7	3.IC.SLE.02	NEW	Discuss the importance of a positive digital footprint	Continuity with K-2
8	4.NI.NCO.01	Model how information is broken down into packets, transmitted through multiple computing devices over networks and the internet and reassembled at the destination.	Explain how information is broken down into packets, transmitted through multiple computing devices over networks and the internet and reassembled at the destination.	After considering public comment, the decision to distribute <i>Network Communication & Organization</i> standards throughout 3rd-5th grade was made.

	STANDARD #	ORIGINAL STANDARD	AMENDED STANDARD	RATIONALE/COMMENTS
9	4.AP.A.01	Compare and refine multiple algorithms for the same task.	Compare and simplify multiple algorithms (sets of step-by-step instructions) for accomplishing the same task verbally, kinesthetically, with robot devices or a programming language.	The standard has been clarified to identify its intended purpose. This also explains that algorithms do not need to occur only in a programming language. The workgroup decided on this change.
10	4.AP.C.01	Create programs using a programming language that utilize sequencing, repetition, conditionals and variables using math operations manipulate values to solve a problem or express ideas both independently and collaboratively.	Create a program using control structures (e.g., sequence, conditionals, interactive-looping) to solve a problem or express ideas both independently and collaboratively.	Public comment recommended, the emphasis of control structures which programming languages use to make decisions first.
11	4.AP.PD.01	Use an iterative process to plan the development of a program that includes user preferences while solving simple problems.	Use an iterative and collaborative process to plan the development of a program that includes user preferences while solving simple problems.	After reviewing public comment, the decision to include more collaborative practices throughout the 3rd-5th grade was made.
12	4.AP.PD.03	Analyze, create and debug a program that includes sequencing, repetition, conditionals and variables in a programming language (e.g., Java, Java Script, Python, Blockly, etc.).	Analyze, create and debug a program that includes sequencing, repetition, conditionals and variables in a programming language.	After considering public comment, the decision to remove the examples of programming language was made.
13	4.IC.SI.01	Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community (e.g., using strong passwords, creating a positive online community, recognizing spam and what to do about it, citing online sources). Identify and report inappropriate behavior.	Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community (e.g., using strong passwords, creating a positive online community, recognizing spam and what to do about it, citing online sources). Identify and report inappropriate behavior and know how to report cyberbullying.	The workgroup decided on this change.
14	4.IC.SI.02	As a team, consider each other's perspectives on improving a computational product.	This standard was deleted.	Just added a layer of confusion, not needed. The workgroup decided on this change.
15	4.IC.SLE.02	NEW	Discuss and understand the implications of a negative digital footprint.	Added digital footprint for continuity from K-2.
16	5.NI.NCO.01	NEW	Model how information is broken down into packets, transmitted through multiple computing devices over networks and the internet and reassembled at the destination.	The first draft did not have a 5th grade <i>Network Communication & Organization</i> standard.

	STANDARD #	ORIGINAL STANDARD	AMENDED STANDARD	RATIONALE/COMMENTS
17	5.AP.A.01	Compare and refine multiple algorithms for the same task and determine which is the most efficient.	Compare and simplify multiple algorithms (sets of step-by-step instructions) for accomplishing the same task verbally, kinesthetically, with robot devices or a programming language and determine which is the most efficient.	The standard has been clarified to identify its intended purpose. This also explains that algorithms do not need to occur only in a programming language. The workgroup decided on this change.
18	5.AP.C.01	Create programs using a programming language that utilize sequencing, repetition, conditionals, event handlers and variables using math operations to manipulate values to solve a problem or express ideas both independently and collaboratively.	Create a program using control structures (e.g., sequence, conditionals, interactive-looping), event handlers and variables to solve a problem or express ideas both independently and collaboratively.	Public comment recommended the emphasis of control structures which programming languages use to make decisions first.
19	5.AP.PD.01	Use an iterative process to plan the development of a program that includes others' perspectives and user preferences while solving simple problems.	Use an iterative and collaborative process to plan the development of a program that includes other perspectives and user preferences while solving simple problems.	After reviewing public comment, the decision to include more collaborative practices throughout the 3rd-5th grade was made.
20	5.AP.PD.03	Analyze, examine, create and debug a program that includes sequencing, repetition, conditionals and variables in a programming language (e.g., Java, Java Script, Python, Blockly, etc.).	Analyze, examine, create and debug a program that includes sequencing, repetition, conditionals and variables in a programming language.	After considering public comment, the decision to remove the examples of programming language was made.
21	5.IC.SI.01	Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community (e.g., talking safely online, promoting good digital citizens, privacy settings and cyberbullying). Identify and report inappropriate behavior.	Develop a code of conduct, explain and practice grade-level appropriate behavior and responsibilities while participating in an online community (e.g., talking safely online, promoting good digital citizens, privacy settings, cyberbullying). Identify and report inappropriate behavior and know how to report cyberbullying.	Added cyberbullying for continuity from K-2. The workgroup decided on this change.
22	5.IC.SI.02	As a team, collaborate with outside resources, other grade levels or online collaborative spaces, to include diverse perspectives to improve computational products.	This standard was deleted.	Just added a layer of confusion, not needed. The workgroup decided on this change.
23	5.IC.SLE.02	NEW	Continue to discuss and understand the implications of positive and negative digital footprints and that they never go away.	Added digital footprint for continuity from K-2. The workgroup decided on this change.

	STANDARD #	ORIGINAL STANDARD	AMENDED STANDARD	RATIONALE/COMMENTS
24	6-8.IC.C.01	Compare tradeoffs associated with computing technologies that have impacted people's activities, careers and globalization.	Compare tradeoffs associated with computing technologies that have impacted people's activities, careers and lives when solving global problems using the power of computing.	Public comment suggested integrating in real world problems would be a way to connect curricular areas and increase interest in computer science courses.
25	11-12.AP.M.03	NEW	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.	Public comment suggested that this was an important concept that was left out.
26	11-12.AP.PD.02	Plan and develop programs using a specific development process, such as SCRUM.	Plan and develop programs using a development process (e.g. waterfall, iterative, spiral, rapid application development, agile).	Public comment suggested that the draft standard was too specific and didn't allow for looking at the strengths and weaknesses of methodologies.
27	11-12.AP.PD.06	NEW	Explain security issues that might lead to compromised computer programs.	Public comment suggested that the standards were missing a standard about security issues that could arise from incorrect program development
28	11-12.AP.PD.07	NEW	Evaluate key qualities of a program through a process such as a code review.	Public comment addressed that the process of performing some type of code review was missing from the standards.