MARYLAND STATE DEPARTMENT OF EDUCATION
EQUITY AND EXCELLENCE

Karen B. Salmon, Ph.D.
State Superintendent of Schools

TO:	Members of the State Board of Education
FROM:	Karen B. Salmon, Ph.D. State Superintendent of Schools
DATE:	September 25, 2018
SUBJECT:	Report on the Maryland K-12 Academic Standards for Computer Science Acceptance

Purpose:
The purpose of this item is to request the Board's acceptance of the Maryland K-12 Academic Standards for Computer Science. The K-12 Computer Science Standards provide guidance on the necessary knowledge, skills, and applications for writers to successfully compose coherent, rigorous curricula.

Historical Background:
In October 2016, The Association for Computing Machinery, Code.org, Computer Science Teachers Association (CSTA), Cyber Innovation Center, and the National Math and Science Initiative partnered with stakeholders from across the country, including Maryland, to develop the K-12 Computer Science Framework. Following the publication of the K-12 Computer Science Framework, CSTA released newly revised K-12 Computer Science Standards in July 2017. These standards are based on the core concepts and practices described in the framework.

More recently, on November 2, 2017, Executive Order 01.01.2017.27 *Computer Science Education and Professional Development* was signed by Governor Larry Hogan declaring computer science a priority in Maryland public schools. This Executive Order was further supported by legislation passed by the General Assembly of Maryland in April 2018 requiring all public high schools in Maryland to offer a computer science course by the 2021-2022 academic school year (House Bill 281 – *Securing the Future: Computer Science Education for All*).

In response to both the legislation and updated National Standards from the CSTA, the need for Maryland-adapted computer science standards was evident. The Division of Career and College Readiness (DCCR), in partnership with the Division of Curriculum, Instructional Improvement and Professional Learning (DCIIP) convened a statewide design team comprised of computer science teachers, local school district central office personnel, members of local industry, representatives from higher education, and individuals from non-profit organizations. The Maryland K-12 Academic Standards for Computer Science place a considerable emphasis on cybersecurity concepts to meet the needs of Maryland's workforce and economic development. Additionally, rather than creating a standards document that encompasses grade-bands, a grade-by-grade approach was taken to offer supplemental support to curriculum developers.

Following completion and review, the Maryland K-12 CS standards were submitted to the CSTA and were determined to be well-aligned to the CSTA K-12 Computer Science Standards, the National K-12 Computer Science Framework, and the Maryland K-12 Academic Standards for Computer Science.

Summary:
The Maryland K-12 Academic Standards for Computer Science are broken down into five core concepts and then further dissected into standards. A copy of the standards are attached. The five broad, overarching concepts are:

1. Computing Systems
2. Networks and the Internet
3. Data and Analysis
4. Algorithms and Programming
5. Impacts of Computing

Executive Summary:
These broad, measurable standards will be used by local school system staff to develop necessary indicator statements and objectives that begin to narrow the focus and provide very clear information about knowledge and specific skills that students will acquire. Next steps include developing a supplemental resource to use when creating standards-based curricula and convening an informational meeting with Supervisors of Computer Science.

Action:
Request Acceptance of the Maryland K-12 Academic Standards for Computer Science by the Maryland State Board of Education

KBS:LMG:csn

Attachment: Maryland K-12 Academic Standards for Computer Science

Maryland's K-12 Computer Science Standards

MARYLAND STATE DEPARTMENT OF EDUCATION

EQUITY AND EXCELLENCE

September 2018

Maryland K-12 Computer Science Standards Design Team

Design Team Leadership	
Lynne Gilli, Ed.D.	Maryland State Department of Education
Val Emrich	Maryland State Department of Education
Marquita Friday	Maryland State Department of Education
Scott Nichols	Maryland State Department of Education
Debra Ward	Maryland State Department of Education

Design Team Participants	
Jeffery Babich	Allegany County Public Schools
Quianna Bannerman	Prince George’s County Public Schools
Elizabeth Bell	Montgomery County Public Schools
Katie Campbell	Harford County Public Schools
Dwight Carr	Johns Hopkins Applied Physics Laboratory
Jena Collins	Apple
Stacey Davis	Baltimore City Public Schools
Diana De Los Santos	Girls Who Code
Luke Erickson	Everfi
Taylor Estes	Somerset County Public Schools
Vanessa Felder	Maryland State Department of Education
William Forrester	Anne Arundel County Public Schools
Shawn Grimes	Digital Harbor Foundation
Stephanie Grimes	Digital Harbor Foundation
Ashley Hart	Everfi
Michael Hines	Prince George’s County Public Schools
Brian Hoffman	Baltimore City Public Schools
Laura Jacob	Kent County Public Schools
Sharon Kramer	Howard County Public Schools
Iris Kutch	Towson University
Amanda Lattimore	Baltimore County Public Schools
Carrie Leary	Anne Arundel Community College
Gretchen LeGrand	Code in the Schools
Kara Lynch	Baltimore County Public Schools
Denise Mandis	St. Mary’s County Public Schools
Felicia Martin-Latief	Prince George’s County Public Schools
Ted McNett	Carroll County Public Schools
Mark Moylan	Anne Arundel County Public Schools
Ed Mullin	Baltimore Robotics Center
Christine Newman	Johns Hopkins University
Loyce Pailen	University of Maryland University College
Karen Parisi	University of Maryland Baltimore County
Rebecca Pearson	Charles County Public Schools
Jandelyn Plane	University of Maryland College Park

Design Team Participants	
Jonathan Prozzi	Digital Harbor Foundation
Andrea Robertson	Montgomery County Public Schools
Corinne Roller	Girls Who Code
Justin Serota	Anne Arundel County Public Schools
Amy Shepler	Caroline County Public Schools
Lisa Shifflett	Anne Arundel County Public Schools
Jennifer Smith	Baltimore City Public Schools
David Starkey	Carroll County Public Schools
LuAnn Stout	Charles County Public Schools
Diane Stulz	Worcester County Public Schools
Shane Wines	Calvert County Public Schools
Bradley Wray	Anne Arundel County Public Schools
Timothy Yauch	Charles County Public Schools
Pat Yongpradit	Code.org

Concept: Computing Systems

	Subconcept: Devices
K.CS.D.01	Select and operate the appropriate computing device to perform a variety of different tasks.
1.CS.D.01	Select and operate the appropriate application/software to perform a variety of tasks or obtain a desired outcome.
2.CS.D.01	Compare and discuss preferences for applications/software with the same primary functionality.
3.CS.D.01	Identify internal and external parts of computing devices that function together to form a system.
4.CS.D.01	Describe how internal and external parts of computing devices function to form a system.
5.CS.D.01	Describe and model how internal and external parts of computing devices function to form a system. Describe how some components rely on others for correct functionality.
6.CS.D.01	Evaluate existing computing devices and make recommendations for improvements to design based on analysis of personal interactions and how others interact with the devices.
7.CS.D.01	Evaluate existing computing devices and make recommendations for improvements to design that consider usability through a variety of lenses (accessibility, ergonomics, learnability, security).
8.CS.D.01	Develop and implement a process to evaluate existing computing devices and make recommendations for improvements to design based on analysis of user interaction and other lenses.
10.CS.D.01	Explain how abstractions hide the underlying implementation of computing systems embedded in everyday objects.
12.CS.D.01	Not addressed at this level.

	Concept: Computing Systems
	Subconcept: Hardware and Software
K.CS.HS.01	Identify by name and locate common computing devices and external hardware in a variety of environments, using appropriate technical terminology (e.g., mobile devices, desktop computer, laptop computer, mouse, keyboard, wearables).
1.CS.HS.01	Identify and describe functions of common computing devices and external hardware (e.g., mobile devices, desktop computer, laptop computer, mouse, keyboard, printer, wearables).
2.CS.HS.01	Identify internal and external components of a computer system and their basic functions (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).
3.CS.HS.01	Identify a variety of ways computer hardware and software work together as a system to accomplish a task.
4.CS.HS.01	Identify and describe a variety of ways computer hardware and software work together as a system to accomplish a task, using appropriate technical terminology (input, output, processors, sensors, storage).
5.CS.HS.01	Model and explain how information flows through hardware and software to accomplish a task.
6.CS.HS.01	Identify ways that hardware and software are combined and work synchronously to collect, store, retrieve, and exchange data.
7.CS.HS.01	Select appropriate hardware and software components for a project considering what type of data will be collected, stored, retrieved, and exchanged.
8.CS.HS.01	Design and refine systems where secure hardware and software are combined to collect, store, retrieve, and exchange data and explain why specific components were chosen for optimality.
10.CS.HS.01	Explain and compare levels of abstraction between application software, system software, and hardware layers.
12.CS.HS.01	Identify, categorize, and illustrate the roles of operating systems to include memory management, data storage/retrieval, process management, and access control.

Concept: Computing Systems

Subconcept: Troubleshooting

K.CS.T.01	Recognize the possibility computing systems might not work as expected and identify basic hardware and software problems using appropriate technical terminology (e.g., monitor turned off, volume decreased on headphones).
1.CS.T.01	Identify and communicate basic hardware and software problems that may occur during use (e.g., application/program not working correctly, no sound coming from device, caps lock turned on), using appropriate technical terminology.)
2.CS.T.01	Identify and summarize basic troubleshooting techniques to solve basic hardware and software problems (e.g., turning off and on a device to restart, closing and reopening an application/program, turning on speakers).
3.CS.T.01	Identify and troubleshoot, using appropriate technical terminology, simple hardware and software problems that may occur during everyday use, discuss problems with peers and adults (e.g. viruses, malware, versions of software and non-working applications, refresh screen, closing/reopening application, adjusting volume on headphones or speakers).
4.CS.T.01	Identify, using appropriate technical terminology, simple hardware and software problems that may occur during everyday use, discuss problems with peers and adults, and apply various strategies for solving these problems (e.g., rebooting the device, checking the power, forced shutdown of an application, running anti-virus).
5.CS.T.01	Identify, using appropriate technical terminology, simple hardware and software problems that may occur during everyday use, discuss problems with peers and adults, apply a variety of strategies for solving these problems, and provide evidence why these strategies did or did not work.
6.CS.T.01	Troubleshoot problems with computing devices and networked components (e.g. peripherals, routers, cables, etc.).
7.CS.T.01	Identify and fix problems with computing devices and their interfaced components using a variety of strategies (e.g. lost data retrieval, hardware password recovery, file restoration, key logging).
8.CS.T.01	Systematically identify and fix problems with computing devices and their interfaced components by using a structured system such as a troubleshooting flow diagram.
10.CS.T.01	Develop and evaluate guidelines and criteria that convey systematic troubleshooting strategies that can be used to identify/fix errors.
12.CS.T.01	Not addressed at this level.

Concept: Networks and the Internet

	Subconcept: Network Communication & Organization
K.NI.NCO.01	Recognize that basic computing devices and components can be connected to one another.
1.NI.NCO.01	Recognize that computing devices can be connected through physical or wireless pathways.
2.NI.NCO.01	Recognize that by connecting computing devices together they can share information (printers, scanners, internet, display devices).
3.NI.NCO.01	Recognize how information is sent and received over physical or wireless pathways.
4.NI.NCO.01	Summarize how information is sent and received over physical and wireless pathways (e.g.,information is deconstructed into smaller pieces called packets, transmitted to final destination, and reassembled).
5.NI.NCO.01	Model how information is deconstructed into packets (smaller pieces), transmitted through multiple devices over the internet and networks, and reassembled at the final destination.
6.NI.NCO.01	Model a simple protocol for transferring information, using packets, across networks and the internet.
7.NI.NCO.01	Explain and model the process to replace lost packets using a protocol for information transfer.
8.NI.NCO.01	Model and explain how data is sent using protocols to choose the fastest pathway, to deal with missing information, and to deliver data securely.
10.NI.NCO.01	Evaluate the scalability and reliability of networks by identifying and describing the relationship between routers, switches, servers, topology, and addresses.
10.NI.NCO.02	Describe the issues that impact network functionality (e.g. Bandwidth, load, delay, topology, TCP/IP and OSI Models).
12.NI.NCO.01	Evaluate the scalability and reliability of networks by identifying and describing the relationship between routers, switches, servers, topology, protocols, and addressing.

Concept: Networks and the Internet

Subconcept: Cybersecurity

K.NI.C.01	Identify and use passwords and discuss why they are not shared with others.
1.NI.C.01	Recognize what passwords are, why they are used, and why they are not shared.
2.NI.C.01	Identify differences between strong and weak passwords and explain the importance of choosing strong passwords to protect devices and information from unauthorized users.
3.NI.C.01	Discuss basic issues that relate to responsible use of computing devices and describe consequences of inappropriate use in a variety of locations.
4.NI.C.01	Identify problems that relate to unsecure networks and inappropriate use of computing devices and potential subsequent consequences.
5.NI.C.01	Define personal identifiable information (e.g., digital footprint) and why it should be protected as related to real world cyber security problems.
5.NI.C.02	Discuss real-world cybersecurity problems and explain how personal information can be protected (e.g., antivirus software, backing up data, strong passwords).
6.NI.C.01	Identify existing cybersecurity concerns with the internet, its connected devices (i.e. IoT) and the systems it uses.
7.NI.C.01	Explain how to protect electronic information using both physical (hard drive) and digital measures; explain existing cybersecurity concerns with the internet and the systems it uses.
8.NI.C.01	Evaluate physical and digital security measures that have been developed and implemented to protect electronic information; discuss the impacts of hacking, ransomware, scams, fake scams, and ethical/legal concerns.
10.NI.C.01	Illustrate how sensitive data and critical infrastructure can be affected by malware and other attacks and recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.
10.NI.C.02	Explain tradeoffs when selecting and implementing cybersecurity recommendations from multiple perspectives such as the user, enterprise, and government.

10.NI.C.03	Understand and identify the relationship between Confidentiality, Integrity, Availability (CIA) Triad and the security measures that address the balance between them as it pertains to data.
10.NI.C.04	Identify ethical concerns about individual privacy, intellectual property, entering systems without permission, and destroying data and demonstrate the ability to exercise proper judgement and best practices in a variety of different scenarios.
10.NI.C.05	Recognize and prevent social engineering attacks. Differentiate between legitimate and fraudulent information.
12.NI.C.01	Compare and refine ways software developers protect devices and information from unauthorized access including complex encryption algorithms such as public key encryption.

	Concept: Data and Analysis
	Subconcept: Storage
K.DA.S.01	Identify that information from our everyday lives that can be stored and accessed via computing devices.
1.DA.S.01	Identify, access, modify, and save an existing file with a computing device.
2.DA.S.01	Create, copy, manipulate, and delete a file on a computing device. Identify the information stored as data.
3.DA.S.01	Recognize that different types of information are stored in different formats that have varying characteristics, which could include associated programs and storage requirements.
4.DA.S.01	Store information in various formats for specific purposes (e.g. file type, file size, file compression).
5.DA.S.01	Convert different types of information into various formats to be used across multiple software/ hardware.
6.DA.S.01	Identify multiple encoding schemes that can be used to represent the same data.
7.DA.S.01	Represent data using multiple encoding schemes.
8.DA.S.01	Evaluate different schemes of encoding data in order to effectively choose the most appropriate method of representation.
10.DA.S.01	Translate, compare, and evaluate different bit representations of real-world phenomena (large data sets), such as characters, numbers, and images and how they are organized and stored.
12.DA.S.01	Not addressed at this level.

Concept: Data and Analysis	
	Subconcept: Collection, Visualization, and Transformation
K.DA.CVT.01	With guidance, collect data on a basic topic (e.g. weather, temperature) and present it visually.
1.DA.CVT.01	With guidance, collect and organize data. Present data effectively in two different ways.
2.DA.CVT.01	With guidance, collect, organize, and present the same data in a variety of visual ways (bar graph, pie chart, table, etc.).
3.DA.CVT.01	Collect, organize, and present the same data in a variety of visual formats (e.g. charts, graphs, tables, etc.)
4.DA.CVT.01	Organize and present collected data in a variety of visual formats to emphasize particular aspects or parts of the data set to make interpretation easier.
5.DA.CVT.01	Interpret and communicate data in a variety of visual formats to highlight the relationships among the data to support a claim.
6.DA.CVT.01	Collect data using computational tools and transform the data to make it more useful.
7.DA.CVT.01	Collect data using computational tools and hardware (e.g., sensors) and transform the data to make it more useful and reliable.
8.DA.CVT.01	Develop and implement a refined process that uses computational tools to transform data to a more useful and reliable state.
10.DA.CVT.01	Use software tools to develop interactive data representations that help others to better understand real-world phenomena
12.DA.CVT.01	Use data analysis tools and techniques to identify patterns in data representing complex systems.
12.DA.CVT.02	Use a variety of robust data collection techniques and tools to generate data sets that support a claim or communicate information.

	Concept: Data and Analysis
	Subconcept: 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; complete a pattern), with or without a computing device.
1.DA.IM.01	With guidance, identify, interpret, and analyze 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, collect, organize, present, and analyze data in a chart or graph (visualization) in order to make a prediction, with or without a computing device.
2.DA.IM.01	With guidance, collect, organize, present, and analyze data in a chart or graph (visualization) in order to make a prediction, with or without a computing device.
3.DA.IM.01	Utilize data to make predictions and discuss whether there are sufficient data to make these predictions and extrapolations.
4.DA.IM.01	Discuss the potential accuracy of conclusions and predictions based on the adequacy of the data set (number of data).
5.DA.IM.01	Refer to data sets to highlight or propose cause-and-effect relationships, predict outcomes, or communicate ideas.
6.DA.IM.01	Identify relevant data points and use models and simulations to formulate, refine, and test hypotheses.
7.DA.IM.01	Verify a model's accuracy by comparing the results with observed data.
8.DA.IM.01	Refine existing or develop and implement new computational models based on observed and generated data.
10.DA.IM.01	Design computational models that identify and represent the relationships among different elements of data collected from a phenomenon or process.
12.DA.IM.01	Evaluate the ability of models and simulations to test and support refinement of hypotheses.

Concept: Algorithms and Programming

Subconcept: Algorithms

K.AP.A.01	Model daily processes and follow basic algorithms (step-by-step lists of instructions) to complete tasks.
1.AP.A.01	Model daily processes and follow basic algorithms (step-by-step lists of instructions) to complete tasks verbally, kinesthetically, via a programming language, or using a device.
2.AP.A.01	Model daily processes by creating and following algorithms (step-by-step lists of instructions) to complete tasks verbally, kinesthetically, via a programming language, or using a device.
3.AP.A.01	Develop and compare multiple algorithms for the same task.
4.AP.A.01	Develop, compare, and refine multiple algorithms for the same task.
5.AP.A.01	Develop, compare, and refine multiple algorithms for the same task and determine which algorithm is the most appropriate.
6.AP.A.01	Use an existing algorithm or pseudocode to solve a problem.
7.AP.A.01	Select and modify existing algorithms and pseudocode to solve complex problems.
8.AP.A.01	Develop and implement algorithms and pseudocode to solve complex problems.
10.AP.A.01	Develop prototypes that use algorithms (e.g., sequencing, selection, iteration, recursion, etc.) to solve computational problems by leveraging prior student knowledge and personal interest.
10.AP.A.02	Design and implement an algorithm to play a game against a human opponent or solve a problem.
12.AP.A.01	Describe how artificial intelligence drives many software and physical systems (e.g., autonomous robots, computer vision, pattern recognition, test analysis).
12.AP.A.02	Design and implement an algorithm to play a game against a human opponent or solve a problem.
12.AP.A.03	Design and implement encryption algorithms to securely store and retrieve information.
12.AP.A.04	Analyze and refine classic algorithms to solve problems.
12.AP.A.05	Evaluate algorithms (e.g., searching, sorting) in terms of their efficiency, correctness, and clarity.

Concept: Algorithms and Programming

Subconcept: Variables

K.AP.V.01	With guidance, model the way programs store and manipulate grade-level data by using numbers or other symbols to represent information (e.g., encode or decode words using numbers, pictographs, or symbols to letters, words, or direction).
1.AP.V.01	With guidance, model the way programs store and manipulate grade-level data by using numbers or other symbols to represent information (e.g., encode or decode words using numbers, pictographs, or symbols to letters, words, or direction).
2.AP.V.01	Model the way programs store and manipulate grade-level data by using numbers or other symbols to represent information (e.g., encode or decode words using numbers, pictographs, or symbols to letters, words, or direction).
3.AP.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.
6.AP.V.01	Decide when and how to declare and name new variables.
7.AP.V.01	Create clearly named variables that represent different types of data.
8.AP.V.01	Create clearly named variables of different data types that utilize naming conventions to improve program readability; perform operations on variable values.
10.AP.V.01	Identify common features in multiple lines of code and substitute a single segment that uses lists (arrays) to account for differences.
10.AP.V.02	Utilize lists to simplify solutions, generalizing computational problems, instead of repeatedly utilizing simple variables.
12.AP.V.01	Compare and contrast foundational data structures and their primary functions.

Concept: Algorithms and Programming

Subconcept: Control

K.AP.C.01	With guidance, create a set of instructions (programs) to accomplish task using a programming language, device, or unplugged activity, including sequencing, emphasizing the beginning, middle, and end.
1.AP.C.01	With guidance, create programs by using creative expression or problem solving, to accomplish tasks that include sequencing and repetition. Programming languages, robot devices, or unplugged activities can serve as the means.
2.AP.C.01	Create programs using a programming language, robot device, or unplugged activity that utilize sequencing and repetition to solve a problem or express creative ideas.
3.AP.C.01	Using a programming language, create programs that include sequences, loops, conditionals, and variables to solve a problem or express an idea.
4.AP.C.01	Using a programming language, create programs that include sequences, loops, conditionals, and variables that utilize mathematics operations to manipulate values in order to solve a problem or express an idea.

5.AP.C.01	Using a programming language, create programs that include sequences, loops, conditionals, event handlers, and variables that utilize mathematics operations to manipulate values in order to solve a problem or express an idea.
6.AP.C.01	Develop secure programs that utilize combinations of loops, conditionals, and the manipulation of variables representing different data types.
7.AP.C.01	Develop secure programs that utilize combinations of loops, compound conditionals, and the manipulation of variables representing different data types.
8.AP.C.01	Develop secure programs that utilize combinations of nested loops, compound conditionals, procedures with and without parameters, and the manipulation of variables representing different data types.
10.AP.C.01	Justify and explain the rationale behind the selection of specific control structures when tradeoffs involve implementation, readability, and program performance.
10.AP.C.02	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
12.AP.C.01	Illustrate the flow of execution of a recursive algorithm.

Concept: Algorithms and Programming	
	Subconcept: Modularity
3.AP.M.01	Decompose a simple problem into a precise set of sequenced instructions.
3.AP.M.02	Modify, remix, or incorporate portions of an existing program into one's own work, to develop or add more advanced features (grade-level appropriate).
4.AP.M.01	Decompose a large problem into smaller, manageable sub-problems to facilitate the program development process.
4.AP.M.02	Modify, remix, or incorporate portions of an existing program into one's own work, to develop or add more advanced features (grade-level appropriate).
5.AP.M.01	Decompose a large problem into smaller, manageable sub-problems and then further into sets of sequenced instructions to facilitate the program development process.
5.AP.M.02	Modify, remix, or incorporate portions of an existing program into one's own work, to develop or add more advanced features (grade-level appropriate).
6.AP.M.01	Decompose problems and sub-problems into parts to facilitate the secure design, implementation, and review of programs.
6.AP.M.02	Identify and use functions within a program to repeat instructions.
7.AP.M.01	Decompose problems and sub-problems into parts to facilitate the secure design, implementation, and review of increasingly complex programs.
7.AP.M.02	Create and use a function in a program to repeat instructions in order to organize code and make it easier to reuse.
8.AP.M.01	Decompose problems and sub-problems into parts to facilitate the secure design, implementation, and review of complex programs.
8.AP.M.02	Create and use a function with parameters in a program to repeat instructions in order to organize code and make it easier to reuse.
10.AP.M.01	Systematically analyze problems, using top down design, in order to break them down into smaller components, using procedures, modules, and/or objects to implement abstractions.
10.AP.M.02	Create computational artifacts by using common structures to organize, manipulate, and process data.

12.AP.M.01	Construct solutions to problems using student-created components, such as procedures, modules, and objects to implement abstractions.
12.AP.M.02	Analyze a large-scaled computational problem and identify generalizable patterns that can be applied to a solution.
12.AP.M.03	Create programming solutions using libraries and APIs through the application of code reuse.

Concept: Algorithms and Programming	
	Subconcept: Program Development
K.AP.PD.01	With guidance, create a grade-level appropriate document to illustrate thoughts, ideas, or stories in a sequential manner (e.g., storyboard, story map, sequential graphic organizer).
K.AP.PD.02	Give attribution to ideas, solutions, and creations of others, verbally, while developing algorithms.
K.AP.PD.03	Identify errors in an algorithm that includes sequencing and repeated procedures using a programming language or unplugged activity. Discuss how errors in the algorithm could be corrected.
K.AP.PD.04	Use correct terminology (e.g., first, second, etc.) in the development of an algorithm to solve a simple problem.
1.AP.PD.01	Create a grade-level appropriate document to illustrate thoughts, ideas, or stories in a sequential manner (e.g., storyboard, story map, sequential graphic organizer).
1.AP.PD.02	Give attribution to ideas, solutions, and creations of others, verbally or written, while writing or developing algorithms and programs.
1.AP.PD.03	Identify and correct errors (debug) in programs which include sequencing and repetition to accomplish a task, through variety of techniques and strategies that could include an unplugged activity (e.g., changing order or sequence, following steps, trial and error).
1.AP.PD.04	Use correct terminology (e.g., beginning, middle, end, etc.), and explain choices made during the development of an algorithm and/or program to solve a simple problem.
2.AP.PD.01	With guidance, create a grade-level appropriate document to clarify the steps that will be needed to create a sequential program and can be used to check if the program is correct.
2.AP.PD.02	Give attribution to ideas, solutions, and creations of others, verbally and written, while writing and developing programs.
2.AP.PD.03	Develop and debug programs that include sequencing and repetition to accomplish a task, though the use of a programming language and/or unplugged activity.
2.AP.PD.04	Use correct terminology (e.g., debug, program input/output, code, etc.) to explain the development of a program to solve a problem in an unplugged activity, hands-on manipulative, or programming language.

3.AP.PD.01	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences while solving simple problems.
3.AP.PD.02	Identify instances of remixing, when ideas are borrowed and iterated upon, and provide attribution.
3.AP.PD.03	Analyze and debug an existing program or algorithm that includes sequencing, repetition, and variables in a programming language.
3.AP.PD.04	Communicate and explain program development to peers and adults using comments, presentations, and demonstrations.
4.AP.PD.01	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences while solving simple problems.
4.AP.PD.02	Observe intellectual property rights and give appropriate attribution when creating or remixing programs.
4.AP.PD.03	Create and debug a program or algorithm that includes sequencing, repetition, and variables in a programming language.
4.AP.PD.04	Communicate and explain program development to peers and adults using comments, presentations, and demonstrations.
5.AP.PD.01	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences while solving problems.
5.AP.PD.02	Observe intellectual property rights and give appropriate attribution when creating or remixing programs.
5.AP.PD.03	Create, test, and debug a program that includes sequencing, repetition, and variables in a programming language to ensure it runs as intended.
5.AP.PD.04	Communicate and explain program development to peers and adults using comments, presentations, and demonstrations.
6.AP.PD.01	Seek and incorporate feedback from team members to refine the solution to a problem.
6.AP.PD.02	Incorporate existing code, media, and libraries into original programs from secure sources, and give appropriate attribution.
6.AP.PD.03	Test and refine existing and original programs.

6.AP.PD.04	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
6.AP.PD.05	Document programs in order to make them easier to understand, test, and debug.
7.AP.PD.01	Seek and incorporate feedback from team members and users to refine the solution to a problem.
7.AP.PD.02	Incorporate existing code, media and libraries into original programs of increasing complexity, from secure sources, and give appropriate attribution.
7.AP.PD.03	Test and refine existing and original programs using user input and secure software development guidance.
7.AP.PD.04	Explain how effective communication between participants is required for successful collaboration when developing computational artifacts.
7.AP.PD.05	Document complex programs in order to make them easier to understand, test, and debug.
8.AP.PD.01	Seek and incorporate feedback from team members and users to refine the solution to a problem that meets the needs of a diverse group of users.
8.AP.PD.02	Incorporate existing code, media, and libraries into original programs of increasing complexity, from secure sources, and give appropriate attribution.
8.AP.PD.03	Develop a method or implement an existing method to systematically test and refine existing and original programs using user input and secure software development guidance.
8.AP.PD.04	Evaluate communication between participants to determine best practices in collaboration when developing computational artifacts.
8.AP.PD.05	Document complex programs, using multiple methods, in order to make them easier to understand, test, and debug.
10.AP.PD.01	Systematically design and implement programs for broad audiences, solicit user feedback, and refine programs based on user feedback.
10.AP.PD.02	Identify and evaluate licenses that limit or restrict use of computational artifacts and consider implications on original work, especially when incorporating libraries and other resources.
10.AP.PD.03	Evaluate and refine computational artifacts to improve usability, accessibility, and efficiency.

10.AP.PD.04	Design and develop computational artifacts while working collaboratively.
10.AP.PD.05	Represent the design elements and data flow (e.g., flowcharts, pseudocode, etc.) of the development of a complex program through the use of various visual aids and documentation techniques.
12.AP.PD.01	Utilize a software life cycle process, that considers security, to plan and develop programs for all types of users.
12.AP.PD.02	Explain security issues that might lead to compromised computer programs.
12.AP.PD.03	Develop different programs for various computing platforms (e.g., desktop, web, mobile).
12.AP.PD.04	Design software collaboratively using integrated development environments (IDEs), with version control and collaboration systems.
12.AP.PD.05	Develop and use a series of test cases to verify that a program performs according to its design specifications.
12.AP.PD.06	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
12.AP.PD.07	Compare multiple programming languages or libraries and discuss how their features make them suitable for solving different types of problems.
12.AP.PD.08	Evaluate key qualities of a program through a process such as code review.

Concept: Impacts of Computing	
	Subconcept: Culture and Diversity
K.IC.C.01	Use grade-level appropriate language to identify and describe how people use a variety of technologies and applications in their daily work and personal lives.
1.IC.C.01	Use grade-level appropriate language to identify and describe how people use a variety of technologies and applications in their daily work and personal lives.
2.IC.C.01	Use grade-level appropriate language to identify and describe how people use a variety of technologies and applications in their daily work and personal lives and the impact of new technologies on daily life.
3.IC.C.01	Identify how different technologies created by people from diverse backgrounds have contributed to computing and helped to change the world.
3.IC.C.02	Identify potential problems that limit accessibility/usability and how Computing devices have built-in features to increase accessibility for all users.
4.IC.C.01	Summarize how different technologies created by people from diverse backgrounds have contributed to computing and helped to change the world.
4.IC.C.02	Brainstorm solutions to improve accessibility/usability and ways computing could be improved to increase accessibility for all users.
5.IC.C.01	Evaluate how different technologies created by people from diverse backgrounds have contributed to computing and helped to change the world.
5.IC.C.02	Develop, test, and refine computational artifacts to improve accessibility and usability for all users.
6.IC.C.01	Identify tradeoffs associated with computing technologies that affect people's everyday activities and future opportunities (e.g., college acceptances, career choices that include security clearances).
6.IC.C.02	Identify issues of bias and accessibility that occur in the design of existing computing technologies.
7.IC.C.01	Explain how computing impacts people's everyday activities, career options, and diversity in innovation in computing and non-computing fields.

7.IC.C.02	Explain issues of bias and accessibility that occur in the design of existing computing technologies and describe the role and responsibility of a designer in reducing bias.
8.IC.C.01	Compare the tradeoffs associated with computing concepts (e.g., automation, communication, privacy, cybersecurity), explaining their effects on economics and global societies.
8.IC.C.02	Analyze issues of bias and accessibility that occur in the design of everyday computing technologies, the role and responsibility of the designer, and make recommendations for how these issues could be rectified to reduce bias.
10.IC.C.01	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
10.IC.C.02	Evaluate and refine computational artifacts to reduce bias and equity deficits.
10.IC.C.03	Demonstrate and explain how an existing algorithm/computational Innovation applies to problems across disciplines.
10.IC.C.04	Demonstrate and explain how an existing algorithm applies to problems in society.
12.IC.C.01	Evaluate the positive and negative implications computational artifacts have on society.
12.IC.C.02	Evaluate the impact of equity, access, and influence on the distribution of computing resources in the global society.
12.IC.C.04	Predict evolutionary trends of computational innovations that have revolutionized aspects of global society.
12.IC.C.05	Predict how computational innovations may revolutionize aspects of global society.

	Concept: Impacts of Computing
	Subconcept: Social Interactions
K.IC.SI.01	Identify appropriate and safe behaviors when participating online.
1.IC.SI.01	Identify and describe appropriate and inappropriate behaviors when participating online.
2.IC.SI.01	Develop a code of conduct and explain responsible practices when participating online. Practice the code of conduct and identify and report inappropriate behavior.
3.IC.SI.01	Develop a code of conduct, explain, and practice grade-level appropriate behavior and responsibilities while participating online. Identify and report inappropriate behavior.
3.IC.SI.02	Identify how computing devices and computational products have been, or can be, improved by incorporating diverse perspectives.
4.IC.SI.01	Develop a code of conduct, explain, and practice grade-level appropriate behavior and responsibilities while participating online. Identify and report inappropriate behavior.
4.IC.SI.02	Discuss ways existing computing devices or computational products can be improved by collaborating with peers to gain their diverse perspectives.
5.IC.SI.01	Develop a code of conduct, explain, and practice grade-level appropriate behavior and responsibilities while participating online. Identify and report inappropriate behavior.
5.IC.SI.02	Discuss ways existing computing devices or computational products can be improved by collaborating with outside resources (other grade levels, businesses) to gain their diverse perspectives.
6.IC.SI.01	Describe and use safe, appropriate, and responsible practices when participating online (e.g., discussion groups, blogs, social networking sites) to maintain a clean record for Post-secondary choices that require security clearances and background checks.
7.IC.SI.01	Individually and collaboratively use advanced tools to design and create online content (e.g., digital portfolio, multimedia, blog, webpage).
8.IC.SI.01	Communicate and publish key ideas and details individually or collaboratively in a way that informs, persuades, and/or entertains using a variety of digital tools and media-rich resources.

10.IC.SI.01	Demonstrate and explain how various methods of collaboration can increase diverse ideas and solutions.
12.IC.SI.01	Select and justify the tools and methods used for collaboration on a project to increase diverse ideas and solutions.

Concept: Impacts of Computing

	Subconcept: Safety, Law, Ethics
K.IC.SLE.01	Keep login information private and log off of devices appropriately.
1.IC.SLE.01	Keep login information private and log off of devices appropriately.
2.IC.SLE.01	Keep login information private and log off of devices appropriately.
3.IC.SLE.01	Introduce intellectual property concepts and identify types of digital data (music, videos, photos) that may have intellectual property rights preventing copying and/or requiring attribution.
4.IC.SLE.01	Observe intellectual property law and give appropriate credit when using resources.
5.IC.SLE.01	Discuss personal consequences and social impact of violating intellectual property rights or failing to provide appropriate attribution.
6.IC.SLE.01	Differentiate between appropriate and inappropriate content on the internet, and identify unethical and illegal online behavior and consequences.
7.IC.SLE.01	Explain the connection between the longevity of data on the internet, personal online identity, and personal privacy.
8.IC.SLE.01	Discuss the social impacts and ethical considerations associated with cybersecurity, including the positive and malicious purposes of hacking.
10.IC.SLE.01	Explain the positive and negative consequences that intellectual property laws can have on innovation.
10.IC.SLE.02	Explain the privacy concerns related to the collection, generation, and analysis of large-scaled data that may not be evident to users.
10.IC.SLE.03	Evaluate the social and economic implications of privacy in the context of safety, law, and ethics.
12.IC.SLE.01	Debate the laws and regulations that govern and impact the development of computing innovations and policies.
12.IC.SLE.02	Investigate reasons new technologies require evaluation of existing laws and regulations and the creation of new legislation.

Computer Science Standards Glossary

Term	Definition
Abstraction	**(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]
Accessibility	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]
Algorithm	A step-by-step process to complete a task.
Analog	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]
App	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]
Artifact	Anything created by a human. See computational artifact for the definition used in computer science.
Artificial Intelligence	The capability of a machine to imitate human behavior.
Audience	Expected end user of a computational artifact or system.
Accessibility	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. [Wickipedia]
Attribution	The action of ascribing a work or remark to a particular author, artist, or person. (re: copyright)
Authentication	The verification of the identity of a person or process. [FOLDOC]
Automate	To link disparate systems and software so that they become self-acting or self-regulating. [Ross, 2016]

Term	Definition
Automation	The process of automating.
Backdoor	A backdoor is a technique in which a system security mechanism is bypassed undetectably to access a computer or its data. The backdoor access method is sometimes written by the programmer who develops a program. [Technoedia]
Black-hat hacking	Unauthorized user who attempts to or gains access to an information system.
Boolean	A type of data or expression with two possible values: true and false. [FOLDOC]
Buffer overflow	A condition at an interface under which more input can be placed into a buffer or data holding area than the capacity allocated, overwriting other information. Attackers exploit such a condition to crash a system or to insert specially crafted code that allows them to gain control of the system. [SOURCE: SP 800-28; CNSSI-4009] NIST, 2010
Bug	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]
Checksum	Value computed on data to detect error or manipulation. [SOURCE: CNSSI-4009] NIST, 2010
Cloud Computing	The practice of using of a set of networks, telecommunications, and computers that store and transmit data by a 3rd party who holds the files in what is called “the cloud”. Cloud computing essentially make it so that (1) users do not have to manage the space and data on their personal or local company devices, and (2) users can access their data (and company data) from any Internet connected device. Simply put, this means that all files including pictures, music, corporate information and other data are not physically stored or saved on the device that is on hand; they are stored in “the cloud”. [Pailen, 2017] **or** A model for enabling on-demand network access to a shared pool of configurable IT capabilities/ resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. It allows users to access technology-based services from the network cloud without knowledge of, expertise with, or control over the technology infrastructure that supports them. This cloud model is composed of five essential characteristics (on-demand self-service, ubiquitous network access, location independent resource pooling, rapid elasticity, and measured service); three service delivery models (Cloud Software as a Service [SaaS], Cloud Platform as a Service [PaaS], and Cloud Infrastructure as a Service [IaaS]); and four models for enterprise access (Private cloud, Community cloud, Public cloud, and Hybrid cloud). Note: Both the user’s data and the essential security services may reside in and be managed within the network cloud. [Source: CNSSI-4009] NIST, 2010

Term	Definition
Code	Any set of instructions expressed in a programming language. [MDESE, 2016]
Comment	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)
Complexity	The minimum amount of resources, such as memory, time, or messages, needed to solve a problem or execute an algorithm. [NIST/DADS]
Component	An element of a larger group. Usually, a component provides a particular service or group of related services. [Tech Terms, TechTarget]
Computational	Relating to computers or computing methods.
Computational Artifact	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]
Computational Innovation	Innovation that includes a computer or program code as an integral part of its function
Computational Thinking	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]
Computer	A machine or devices that performs processes, calculations, and operations based on instructions provided by a software or hardware program. [Techopedia]
Computer Science	The study of computers and algorithmic processes, including their principles, their hardware and software designs, their implementation, and their impact on society. [ACM, 2006]
Computing	Any goal-oriented activity requiring, benefiting from, or creating algorithmic processes. [MDESE, 2016]
Computing Device	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.

Term	Definition
Computing System	A collection of one or more computers or computing devices, together with their hardware and software, integrated for the purpose of accomplishing shared tasks. Although a 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.
Conditional	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.)
Configuration	**(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]
Connection	A physical or wireless attachment between multiple computing systems, computers, or computing devices.
Connectivity	A program's or device's ability to link with other programs and devices. [Webopedia]
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.
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 behavior of a culture.

Term	Definition
Cybersecurity	The protection against access to, or alteration of, computing resources through the use of technology, processes, and training. [TechTarget] Often described as overall concerns for confidentiality, integrity and availability of systems and data.
Data	Information that is collected and used for reference or analysis. Data can be digital or nondigital and an be in many forms, including numbers, text, show of hands, images, sounds, or video. [CAS, 2013; Tech Terms]
Data Structure	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]
Data type	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.
Debugging	The process of finding and correcting errors (bugs) in programs. [MDESE, 2016]
Decompose	To break down into components.
Decomposition	Breaking down a problem or system into components. [MDESE, 2016]
Device	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]
Digital	A characteristic of electronic technology that uses discrete values, generally 0 and 1, to generate, store, and process data. [Techopedia]
Digital Citizenship	The norms of appropriate, responsible behavior with regard to the use of technology. [MDESE, 2016]
Digital Forensics	The application of science to the identification, collection, examination, and analysis of data while preserving the integrity of the information and maintaining a strict chain of custody for the data. SOURCE: SP 800-86 [NIST]
Efficiency	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]
Encapsulation	The technique of combining data and the procedures that act on it to create a type. [FOLDOC]

Term	Definition
Encryption	The conversion of electronic data into another form, called ciphertext, which cannot be easily understood by anyone except authorized parties. [TechTarget]
End User (or user)	A person for whom a hardware or software product is designed (as distinguished from the developers). [TechTarget]
Event	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]
Event Handler	A procedure that specifies what should happen when a specific event occurs.
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]
Hardware	The physical components that make up a computing system, computer, or computing device. [MDESE, 2016]
Hierarchy	An organizational structure in which items are ranked according to levels of importance. [TechTarget]
Human–Computer Interaction (HCI)	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]
Identifier	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]
Implementation	The process of expressing the design of a solution in a programming language (code) that can be made to run on a computing device.
Inference	A conclusion reached on the basis of evidence and reasoning. [Oxford]
Input	The signals or instructions sent to a computer. [Techopedia]
Integrity	The overall completeness, accuracy, and consistency of data. [Techopedia]

Term	Definition
Intellectual Property	Useful artistic, technical, and/or industrial information, knowledge or ideas that convey ownership and control of tangible or virtual usage and/or representation. SOURCE: SP 800-32 [NIST] Creations of the mind such as musical, literary, and artistic works; inventions; and symbols, names, images, and designs used in commerce, including copyrights, trademarks, patents, and related rights. Under intellectual property law, the holder of one of these abstract “properties” has certain exclusive rights to the creative work, commercial symbol, or invention by which it is covered. SOURCE: CNSSI-4009[NIST] add: Creative Commons is an aspect of copyrights commonly used in education.
Internet	The global collection of computer networks and their connections, all using shared protocols to communicate. [CAS, 2013]
Iterative	Involving the repeating of a process with the aim of approaching a desired goal, target, or result. [MDESE, 2016]
Libraries	need “from the standpoint of storing code”
Logic Bomb	A piece of code intentionally inserted into a software system that will set off a malicious function when specified conditions are met. [SOURCE: CNSSI-4009] NIST, 2010
Loop	A programming structure that repeats a sequences of instructions as long as a specific condition is true. [Tech Terms]
Malware	Apps and Software that run on computers that are placed there by a hacker for malicious purposes. [Pailen, 2017] or A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim’s data, applications or operating system or of otherwise annoying or disrupting the victim. [SOURCE: SP 800-83; SP 800-41] NIST, 2010

Term	Definition
Memory	Temporary storage used by computing devices. [MDESE, 2016]
Model	A representation of some part of a problem or a system. [MDESE, 2016] Note: This definition differs from that used in science.
Modularity	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]
Module	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]
Network	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.
Operation	An action, resulting from a single instruction, that changes the state of data. [Free Dictionary]
Packet	The unit of data sent over a network. [Tech Terms]
Parameter	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]
Parity	Bit(s) used to determine whether a block of data has been altered. [SOURCE: CNSSI-4009] NIST, 2010
Peripherals	Any device connected to the computer such printers, monitors, keyboards and other input/output devices.
Phishing	Phishing, a social engineering attack in the digital world, is the act of hackers sending fraudulent emails (or making bogus phone calls) to unsuspecting targets in an effort to steal valuable personal information for nefarious uses.
Piracy	The illegal copying, distribution, or use of software. [TechTarget]
Procedure	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.

Term	Definition
Process	A series of actions or steps taken to achieve a particular outcome. [Oxford]
Program; Programming	**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]
Protocol	The special set of rules used by endpoints in a telecommunication connection when they communicate. Protocols specify interactions between the communicating entities. [TechTarget]
Prototype	An early approximation of a final product or information system, often built for demonstration purposes. [TechTarget, Techopedia]
Pseudocode	Pseudocode (pronounced SOO-doh-kohd) is a detailed yet readable description of what a computer program or algorithm must do, expressed in a formally-styled natural language rather than in a programming language. [TechTarget, Techopedia
Ransomware	Malicious software or (malware) that can stop users from accessing their computer, laptop, tablet or smartphone, putting a "lock" on files, pictures, data, contacts, and screens until a ransom is paid to the hacker. [Pailen, 2017]
Redundancy	A system design in which a component is duplicated, so if it fails, there will be a backup. [TechTarget]
Reliability	An attribute of any system that consistently produces the same results, preferably meeting or exceeding its requirements. [FOLDOC]
Remix	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]
Router	A device or software that determines the path that data packets travel from source to destination. [TechTarget]
Scalability	The capability of a network to handle a growing amount of work or its potential to be enlarged to accommodate that growth. [Wikipedia]
Security	See the definition for cybersecurity.
Simulate; Simulation	**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]
Software	Programs that run on a computing system, computer, or other computing device.

Term	DEFINITION
SQL Injection	An SQL injection is a computer attack in which malicious code is embedded in a poorly-designed application and then passed to the backend database. The malicious data then produces database query results or actions that should never have been executed. [Techopedia]
Storage	(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]
String	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]
Structure	A general term used in the framework to discuss the concept of encapsulation without specifying a particular programming methodology.
Switch	A high-speed device that receives incoming data packets and redirects them to their destination on a local area network (LAN). [Techopedia]
System	A collection of elements or components that work together for a common purpose. [TechTarget] See also the definition for computing system.
Test Case	A set of conditions or variables under which a tester will determine whether the system being tested satisfies requirements or works correctly. [STF]
Time Bomb	Resident computer program that triggers an unauthorized act at a predefined time. [SOURCE: CNSSI-4009]NIST, 2010
Topology	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]
Troubleshooting	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]
User	See the definition for end user.

Term	Definition
Variable	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] Note: This definition differs from that used in math.
White-Hat Hacking	Authorized user who attempts to or gains access to an information system in order to test the system security.

References

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

Source	Source Location
ACM, 2006	**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.
CAS, 2013	**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. Computing at school PDF
College Board, 2016	**College Board Advanced Placement® Computer Science Principles** College Board. (2016). *AP Computer Science Principles course and exam description*. New York, NY: College Board. AP Course and Exam Computer Science Description
FOLDOC	**Free On-Line Dictionary of Computing** Free on-line dictionary of computing. (n.d.). Retrieved from Dictionary of Computing
Free Dictionary	**The Free Dictionary** The free dictionary. (n.d.). Retrieved from Free Dictionary
Kafai & Burke, 2014	**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.
Lee, 2016	**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. CSTA Voice
MDESE, 2016	**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. Digital Literacy And Computer Science Framework PDF

NCSS, 2013	**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. **C3 Framework Social Studies PDF**
NIST, 2010	**Draft Glossary of Key Information Security Terms.** Richard Kissel, Editor May 26, 2010
NIST/DADS	**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 Dictionary of Algorithms
Oxford	**Oxford Dictionaries** Oxford dictionaries. (n.d.). Retrieved from Oxford Dictionary
Pailen, 2017 PCmag	**Super Cybersecurity Grandma, (2017)**. Jastin Enterprises LLC. **PCmag.com Encyclopedia** PCmag.com encyclopedia. (n.d.). Retrieved from PC Encyclopedia
Ross, 2016	**What Is Automation** Ross, B. (2016, May 10). *What is automation and how can it improve customer service? Information Age*. Retrieved from Information Age - Automation
STF	**Software Testing Fundamentals** Software testing fundamentals. (n.d.). Retrieved from Software Testing Fundamentals
Tech Terms	**Tech Terms** Tech terms computer dictionary. (n.d.). Retrieved from Tech Terms
Techopedia	**Techopedia** Techopedia technology dictionary. (n.d.). Retrieved from Technology Dictionary
TechTarget	**TechTarget Network** TechTarget network. (n.d.) Retrieved from Network of Technology-Specific Websites

Webopedia	**Webopedia** Webopedia. (n.d.). Retrieved from Online Tech Dictionary
Wikipedia	**Wikipedia** Wikipedia: The free encyclopedia. (n.d.). Retrieved from Wikipedia

Maryland K-12 Academic Standards for Computer Science

MARYLAND STATE DEPARTMENT OF EDUCATION

EQUITY AND EXCELLENCE

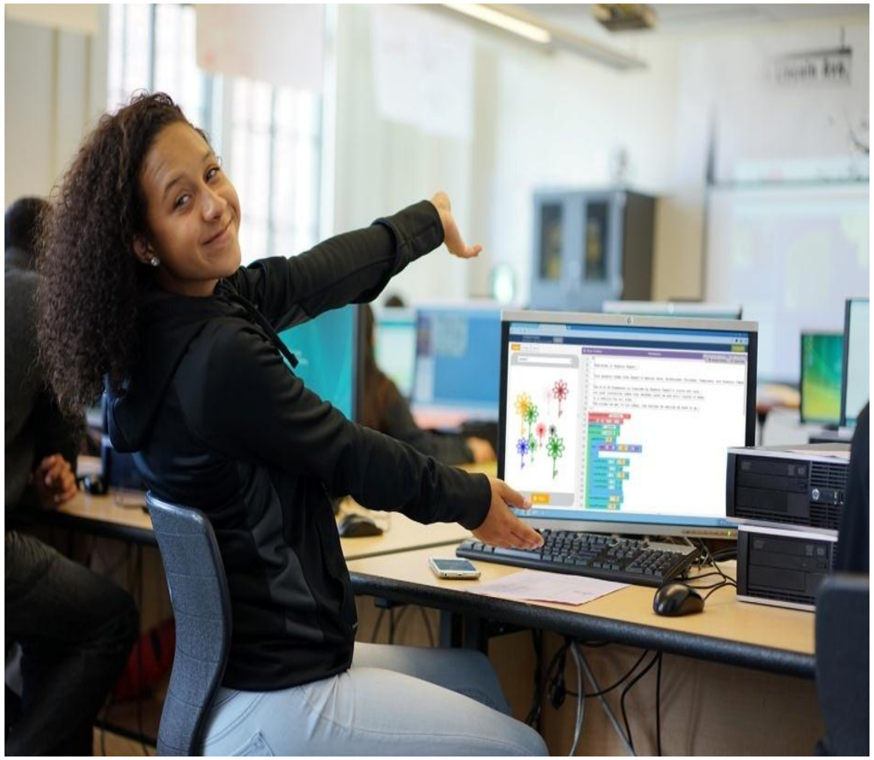
STATE BOARD MEETING

September 25, 2018

Table of Contents

MARYLAND STATE DEPARTMENT OF EDUCATION
EQUITY AND EXCELLENCE

Maryland K-12 Academic Standards for Computer Science



K12 COMPUTER SCIENCE FRAMEWORK

acm

CODE

CSTEACHERS.ORG
COMPUTER SCIENCE TEACHERS ASSOCIATION

cyber innovation center®
Collaboration • Research • Technology

NATIONAL MATH + SCIENCE INITIATIVE

K12 Computer Science Framework

- Released October 2016
- High level, conceptual guide of concepts and practices
- Informs standards, curriculum, professional development, and implementation of Computer Science Pathways

Who was involved in developing the Framework?

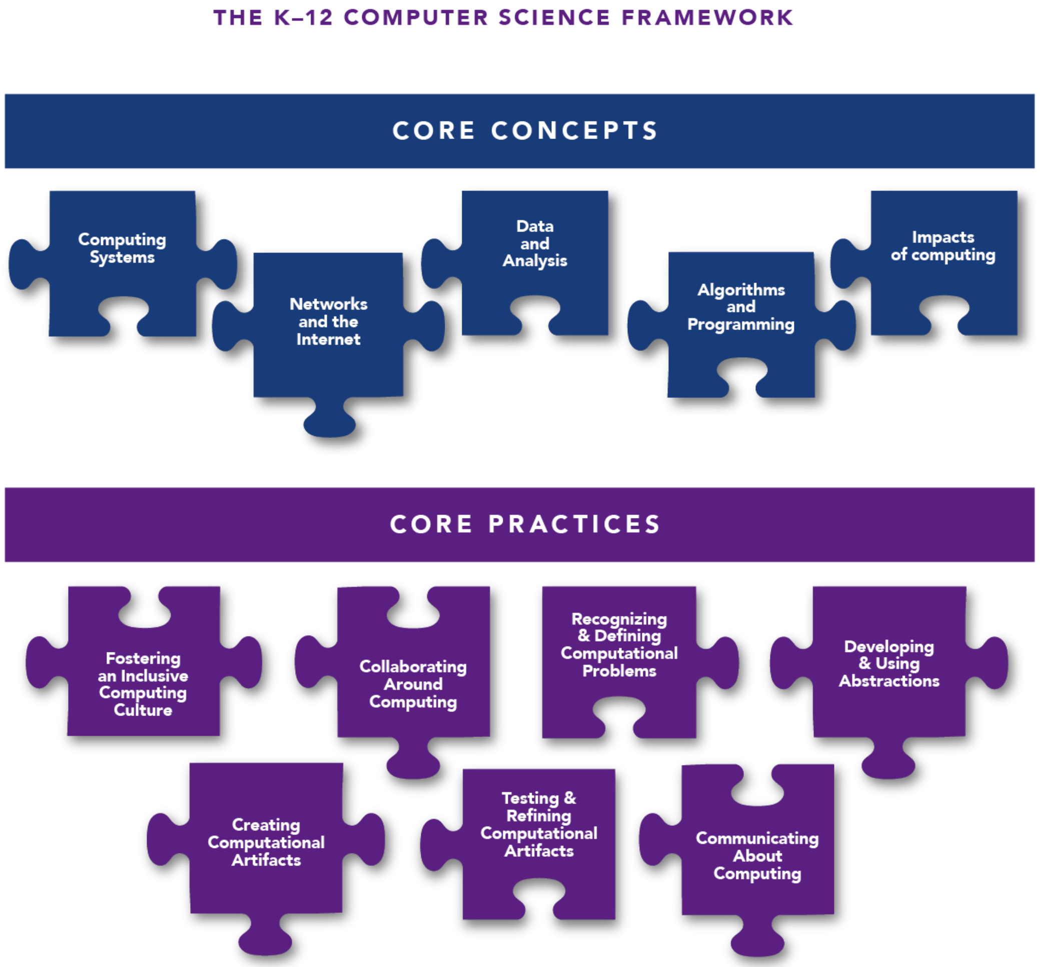
- States and Districts

Arkansas	Iowa	New Jersey	Districts:
California	**Maryland**	North Carolina	**Charles County Public Schools**
Georgia	Massachusetts	Utah	Chicago Public Schools
Idaho	Nebraska	Washington	NYC Department of Education
Indiana	Nevada		San Francisco Unified

- Industry: Google, Amazon, Microsoft, Apple...
- Organizations: College Board, PLTW, Achieve, ISTE, NAF...
- Convener: ACM, Code.org, NICERC, National Math and Science Initiative

Note: ACM – Association for Computing Machinery   NICERC – National Integrated Cyber Education Research Center

- 5 Core Concepts
  - Computing Systems
  - Networks and the Internet
  - Data and Analysis
  - Algorithms and Programming
  - Impacts of Computing
- 7 Practices
  - Fostering an Inclusive Computing Culture
  - Collaborating Around Computing
  - Recognizing & Defining Computational Problems
  - Developing & Using Abstractions
  - Creating Computational Artifacts
  - Testing & Refining Computational Artifacts
  - Communicating About Computing


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

CSTA K-12 Computer Science Standards

- Released July 2017 by the Computer Science Teachers Association (CSTA)
- Designed to be specific, measurable, and include performance expectations
- Included are 5 core concepts further broken down into 16 subconcepts
- Structured in grade level bands (e.g., K-2, 3-5, 6-8, 9-10, 11-12)
- Used to inform curriculum, professional development, and implementation of Career and Technology Education (CTE) Computer Science Programs of Study
- Aligned with the Governor's Executive Order on Computer Science Education and Professional Development aimed at growing the Information Technology Workforce (CTE programs in Cisco Cybersecurity, Oracle – Java Programming, Project Lead The Way Computer Science, and local programs)
- Computer Science Courses are accepted to fulfill the Technology Education Graduation Requirement and one credit in Mathematics

Standards: 5 Concepts & 16 Subconcepts

Computing Systems	Devices
	Hardware and Software
	Troubleshooting
Networks and The Internet	Network Communication and Organization
	Cybersecurity (Emphasized in Maryland due to employers' needs)
Data and Analysis	Storage
	Collection, Visualization, and Transformation
	Inference and Models
Algorithms and Programming	Algorithms
	Variables
	Control
	Modularity
	Program Development
Impacts of Computing	Culture
	Social Interactions
	Safety, Law, and Ethics

Maryland K-12 Academic Standards for Computer Science

- *Representatives from 31 organizations convened from January 2018 – May 2018
  - 15 local schools districts
  - 10 local businesses/non-profit organizations
  - 6 postsecondary institutions
- CSTA reviewed Maryland's Standards and confirmed alignment to the National Standards
- Stronger emphasis on Cybersecurity to meet Maryland's economic and workforce development needs
- Organized by individual grades rather than grade bands
- 2 Versions
  - Similar format and architecture to CSTA Standards
  - Accessible version for MSDE website

* Representatives are listed in the Computer Science Standards Document

Contributors to the Standards Development Process

ACPS ALLEGANY COUNTY PUBLIC SCHOOLS

ANNE ARUNDEL COMMUNITY COLLEGE

ANNE ARUNDEL COUNTY PUBLIC SCHOOLS

BALTIMORE CITY PUBLIC SCHOOLS

The Baltimore County Public Schools

Baltimore Robotics Center

Light the Future Calvert County Public Schools

Caroline County Public Schools

Building the Future CCPS Carroll County Public Schools

Charles County Public Schools

CISCO

CODE

EVERFI

girls who code

HARFORD COUNTY PUBLIC SCHOOLS Inspire • Prepare • Achieve

Howard County Public School System

JOHNS HOPKINS UNIVERSITY

APL JOHNS HOPKINS APPLIED PHYSICS LABORATORY

ANCHORED IN EXCELLENCE KENT COUNTY PUBLIC SCHOOLS

MONTGOMERY COUNTY PUBLIC SCHOOLS

PGCPS Great By Choice

St. Mary's County Public Schools

SUCCESS NOTHING LESS Somerset County Public Schools

STAR ACADEMY

TOWSON

UMBC

UNIVERSITY OF MARYLAND

UMUC University of Maryland University College

Standards Nomenclature

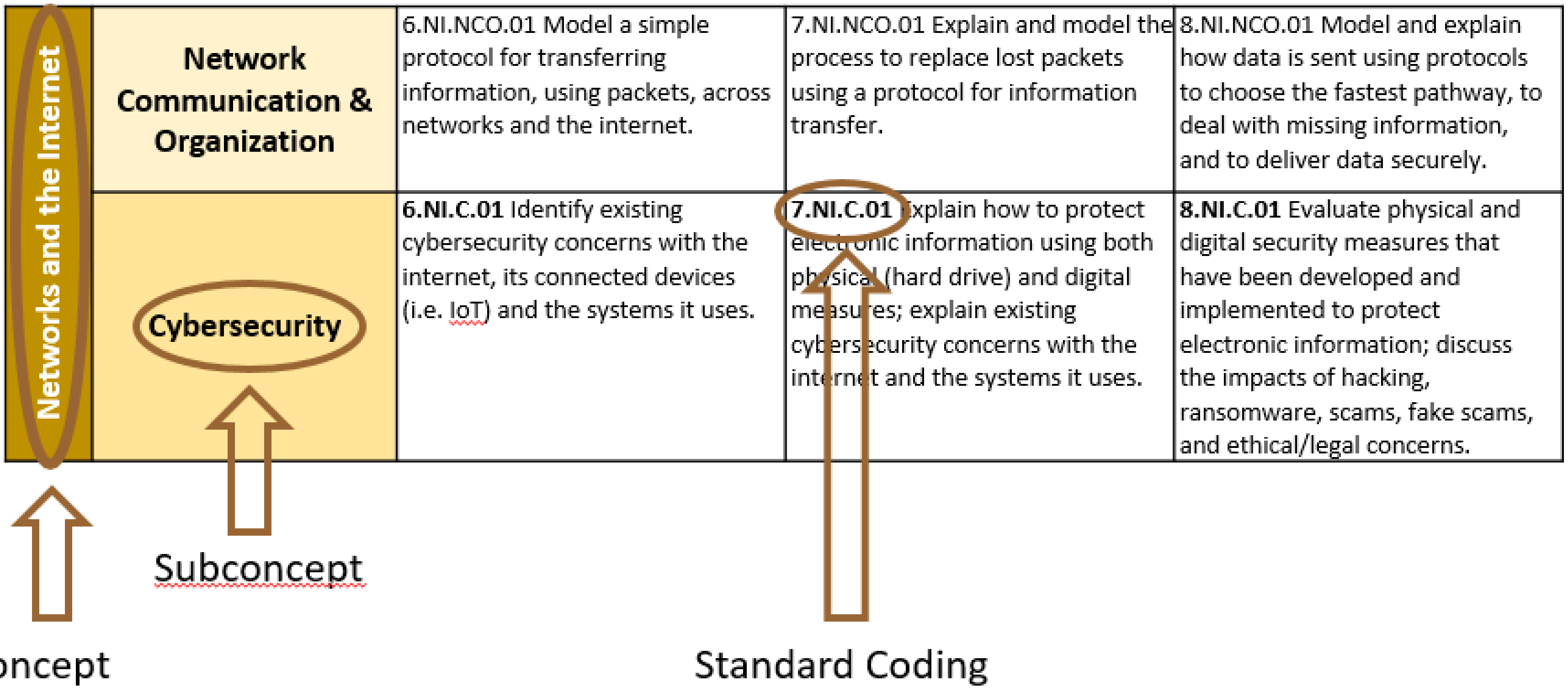
7.NI.C.01

7 – Grade Level

NI – Concept: *Networks & The Internet*

C – Subconcept: *Cybersecurity*

01 – Standard Number



Networks and the Internet	Network Communication & Organization	6.NI.NCO.01 Model a simple protocol for transferring information, using packets, across networks and the internet.	7.NI.NCO.01 Explain and model the process to replace lost packets using a protocol for information transfer.	8.NI.NCO.01 Model and explain how data is sent using protocols to choose the fastest pathway, to deal with missing information, and to deliver data securely.
	Cybersecurity	**6.NI.C.01** Identify existing cybersecurity concerns with the internet, its connected devices (i.e. IoT) and the systems it uses.	**7.NI.C.01** Explain how to protect electronic information using both physical (hard drive) and digital measures; explain existing cybersecurity concerns with the internet and the systems it uses.	**8.NI.C.01** Evaluate physical and digital security measures that have been developed and implemented to protect electronic information; discuss the impacts of hacking, ransomware, scams, fake scams, and ethical/legal concerns.

Next Steps

Develop Annotated Standards

- Annotated descriptions of the standards to provide additional clarity and serve as a tool for local school system curriculum writers

Conduct CS Supervisors' Meeting

- Convene CS Supervisors' from local school districts to review and discuss standards document

Continue Partnerships

- Partner with Maryland Center for Computing Education to provide teacher professional development on the implementation of standards

House Bill 281 – Chapter 358 – passed during the 2018 Legislative Session

Disseminate the CS Standards

- Share the standards with all appropriate stakeholder groups including business, industry, secondary and postsecondary educators, non-profits, and other organizations.