washington
STATE LEARNING STANDARDS

kindergarten – 12th grade

Computer Science

Crosswalks

Legend for Identifiers

Unique Numbering System for the Washington Computer Science K–12 Learning Standards

To help organize and track each individual standard, a unique identifier was developed. An example appears below:

Level	Framework Concept	Number	Computer Science K–12 Learning Standard
Grades 6-8	Algorithms and Programming	17	Systematically test and refine programs using a range of test cases.
2	**AP**	**17**	**Identifier: 2-AP-17**

The identifier code corresponds to:
Level – Concept – Number

Use the following legend to interpret the unique identifier for each Computer Science K–12 Learning Standard:

Identifier Code	Levels
1A	Grades K–2
1B	Grades 3–5
2	Grades 6–8
3A	Grades 9–10
3B	Grades 11–12

Identifier Code	Concepts
CS	Computing Systems
NI	Networks and the Internet
DA	Data and Analysis
AP	Algorithms and Programming
IC	Impacts of Computing

Figure 1: Standards Identifier Code –

Integrated into classroom activities through practices:

#	Practices
1	Fostering an Inclusive Computing Culture
2	Collaborating
3	Recognizing and Defining Computational Problems
4	Developing and Using Abstractions
5	Creating Computational Artifacts
6	Testing and Refining
7	Communicating about Computing

K-2 Crosswalk

K-2 Final 2018	Level 1A	K–2 Interim 2016	Level 1A
1A-CS-01	Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use.	1A-C-7-9	Identify and use software that controls computational devices (e.g., use an app to draw on the screen, use software to write a story or control robots).
1A-CS-02	Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).	1A-C-7-10	Use appropriate terminology in naming and describing the function of common computing devices and components (e.g., desktop computer, laptop computer, tablet device, monitor, keyboard, mouse, printer).
1A-CS-03	Describe basic hardware and software problems using accurate terminology	1A-C-6-11	Identify, using accurate terminology, simple hardware and software problems that may occur during use (e.g., app or program not working as expected, no sound, device won't turn on).
1A-NI-04	Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access	1A-N-7-17	Use passwords to protect private information and discuss the effects of password misuse
1A-DA-05	Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data	1A-D-4-13	Use a computing device to store, search, retrieve, modify, and delete information and define the information stored as data.
1A-DA-06	Collect and present the same data in various visual formats	1A-D-7-12	Collect data over time and organize it in a chart or graph in order to make a prediction.
1A-DA-07	Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.	1A-D-4-14	Create a model of an object or process in order to identify patterns and essential elements (e.g., water cycle, butterfly life cycle, seasonal weather patterns).
1A-AP-08	Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks	1A-A-3-7	Construct and execute algorithms (sets of step-by-step instructions) that include sequencing and simple loops to accomplish a task, both independently and collaboratively, with or without a computing device.


CSTA & ACM, (2016), *Interim CSTA K-12 Computer Science Standards*

K-2 Crosswalk

K-2 Final 2018	Level 1A	K–2 Interim 2016	Level 1A
1A-AP-09	Model the way programs store and manipulate data by using numbers or other symbols to represent information.	A-A-4-4	Use numbers or other symbols to represent data (e.g., thumbs up/down for yes/no, color by number, arrows for direction, encoding/decoding a word using numbers or pictographs).
1A-AP-10	Develop programs with sequences and simple loops, to express ideas or address a problem.	1A-A-5-2	Construct programs, to accomplish a task or as a means of creative expression, which include sequencing, events, and simple loops, using a block-based visual programming language, both independently and collaboratively (e.g., pair programming).
1A-AP-11	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.	1A-A-3-5	Decompose (break down) a larger problem into smaller sub-problems with teacher guidance or independently.
1A-AP-12	Develop plans that describe a program's sequence of events, goals, and expected outcomes.	1A-A-5-3	Plan and create a design document to illustrate thoughts, ideas, and stories in a sequential (step-by-step) manner (e.g., story map, storyboard, sequential graphic organizer).
1A-AP-13	Give attribution when using the ideas and creations of others while developing programs.	1A-A-7-1	Give credit when using code, music, or pictures (for example) that were created by others.
1A-AP-14	Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.	1A-A-6-8	Analyze and debug (fix) an algorithm that includes sequencing and simple loops, with or without a computing device.
1A-AP-15	Using correct terminology, describe steps taken and choices made during the iterative process of program development.	1A-N-2-16	Use computers or other computing devices to connect with people using a network (e.g., the Internet) to communicate, access, and share information as a class.
1A-IC-16	Compare how people live and work before and after the implementation or adoption of new computing technology.	1A-I-7-15	Compare and contrast examples of how computing technology has changed and improved the way people live, work, and interact.


CSTA & ACM, (2016), *Interim CSTA K-12 Computer Science Standards*

K-2 Crosswalk

1A-IC-17	Work respectfully and responsibly with others online		New
1A-IC-18	Keep login information private, and log off of devices appropriately.		New
	Removed	1A-A-3-6	Categorize a group of items based on the attributes or actions of each item, with or without a computing device.

3-5 Crosswalk

3-5 Final 2018	Level 1B	3-5 Interim 2016	Level 1B
1B-CS-01	Describe how internal and external parts of computing devices function to form a system.	1B-C-7-9	Model how a computer system works. [Clarification: Only includes basic elements of a computer system, such as input, output, processor, sensors, and storage.]
1B-CS-02	Model how computer hardware and software work together as a system to accomplish tasks.	1B-C-7-10	Use appropriate terminology in naming internal and external components of computing devices and describing their relationships, capabilities, and limitations.
1B-CS-03	Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	1B-C-6-11	Identify, using accurate terminology, simple hardware and software problems that may occur during use, and apply strategies for solving problems (e.g., reboot device, check for power, check network availability, close and reopen app).
1B-NI-04	Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.	1B-N-4-21	Model how a device on a network sends a message from one device (sender) to another (receiver) while following specific rules.
1B-NI-05	Discuss real-world cybersecurity problems and how personal information can be protected.	1B-I-1-19	Explain problems that relate to using computing devices and networks (e.g., logging out to deter others from using your account, cyberbullying, privacy of personal information, and ownership).
1B-NI-05	Discuss real-world cybersecurity problems and how personal information can be protected.	1B-N-7-20	Create examples of strong passwords, explain why strong passwords should be used, and demonstrate proper use and protection of personal passwords.
1B-DA-06	Organize and present collected data visually to highlight relationships and support a claim.	1B-D-5-13	Answer a question by using a computer to (e.g., sort, total and/or average, chart, graph) and analyze data that has been collected by the class or student.
1B-DA-07	Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.		
1B-AP-08	Compare and refine multiple algorithms for the same task and determine which is the most appropriate.	1B-A-2-1	Apply collaboration strategies to support problem solving within the design cycle of a program.

3-5 Crosswalk

3-5 Final 2018	Level 1B	3-5 Interim 2016	Level 1B
1B-AP-09	Create programs that use variables to store and modify data. Variables are used to store and modify data.	1B-D-4-14	Use numeric values to represent non-numeric ideas in the computer (binary, ASCII, pixel attributes such as RGB).
1B-AP-09	Create programs that use variables to store and modify data. Variables are used to store and modify data.	1B-A-5-5	Use mathematical operations to change a value stored in a variable.
1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	1B-A-5-4	Construct programs, in order to solve a problem or for creative expression, that include sequencing, events, loops, conditionals, parallelism, and variables, using a block-based visual programming language or text-based language, both independently and collaboratively (e.g., pair programming).
1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	1B-A-3-7	Construct and execute an algorithm (set of step-by-step instructions) that includes sequencing, loops, and conditionals to accomplish a task, both independently and collaboratively, with or without a computing device.
1B-AP-11	Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.	1B-A-3-6	Decompose (break down) a larger problem into smaller sub-problems, independently or in a collaborative group.
1B-AP-12	Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	1B-A-7-2	Use proper citations and document when ideas are borrowed and changed for their own use (e.g., using pictures created by others, using music created by others, remixing programming projects).
1B-AP-13	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	1B-A-5-3	Create a plan as part of the iterative design process, both independently and with diverse collaborative teams (e.g., storyboard, flowchart, pseudo-code, story map).
1B-AP-14	Observe intellectual property rights and give appropriate attribution when creating or remixing programs.	1B-A-7-2	Use proper citations and document when ideas are borrowed and changed for their own use (e.g., using pictures created by others, using music created by others, remixing programming projects).

CSTA & ACM, (2016), *Interim CSTA K-12 Computer Science Standards*

3-5 Crosswalk

-		-	
1B-AP-15	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	1B-A-6-8	Analyze and debug (fix) an algorithm that includes sequencing, events, loops, conditionals, parallelism, and variables.
1B-AP-16	Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.	1B-A-2-1	Apply collaboration strategies to support problem solving within the design cycle of a program.
1B-AP-17	Describe choices made during program development using code comments, presentations, and demonstrations.		New
1B-IC-18	Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.	1B-I-7-15	Evaluate and describe the positive and negative impacts of the pervasiveness of computers and computing in daily life (e.g., downloading videos and audio files, electronic appliances, wireless Internet, mobile computing devices, GPS systems, wearable computing).
1B-IC-18	Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.	1B-I-7-16	Generate examples of how computing can affect society, and also how societal values can shape computing choices.
1B-IC-19	Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.	1B-I-1-18	Brainstorm ways in which computing devices could be made more accessible to all users.
1B-IC-20	Seek diverse perspectives for the purpose of improving computational artifacts.	1B-I-1-17	Seek out and compare diverse perspectives, synchronously or asynchronously, to improve a project.
1B-IC-20	Seek diverse perspectives for the purpose of improving computational artifacts.	1B-D-5-12	Create a computational artifact to model the attributes and behaviors associated with a concept (e.g., solar system, life cycle of a plant).
1B-IC-21	Use public domain or creative commons media, and refrain from copying or using material created by others without permission.	1B-A-7-2	Use proper citations and document when ideas are borrowed and changed for their own use (e.g., using pictures created by others, using music created by others, remixing programming projects).

6-8 Crosswalk

-		-	
2-CS-01	Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.	2-A-7-4	Interpret the flow of execution of algorithms and predict their outcomes. [Clarification: Algorithms can be expressed using natural language, flow and control diagrams, comments within code, and pseudocode.]
2-CS-02	Design projects that combine hardware and software components to collect and exchange data.	2-C-7-11	Justify the hardware and software chosen to accomplish a task (e.g., comparison of the features of a tablet vs. desktop, selecting which sensors and platform to use in building a robot or developing a mobile app).
2-CS-03	Systematically identify and fix problems with computing devices and their components.	2-C-6-13	Use a systematic process to identify the source of a problem within individual and connected devices (e.g., follow a troubleshooting flow diagram, make changes to software to see if hardware will work, restart device, check connections, swap in working components).
2-NI-04	Model the role of protocols in transmitting data across networks and the Internet.	2-N-4-25	Simulate how information is transmitted as packets through multiple devices over the Internet and Networks.
2-NI-05	Explain how physical and digital security measures protect electronic information.	2-I-1-22	Describe ethical issues that relate to computing devices and networks (e.g., equity of access, security and plagiarism), hacking, intellectual property, copyright, Creative Commons licensing.
2-NI-06	Apply multiple methods of encryption to model the secure transmission of information.		
2-DA-07	Represent data using multiple encoding schemes.	2-D-4-17	Represent data using different encoding schemes (e.g., binary, Unicode, Morse code, shorthand, student-created codes).
2-DA-08	Collect data using computational tools and transform the data to make it more useful and reliable.	2-D-7-15	Explain the processes used to collect, transform, and analyze data to solve a problem using computational tools (e.g., use an app or spreadsheet form to collect data, decide which data to use or ignore, and choose a visualization method.).
2-DA-09	Refine computational models based on the data they have generated.		New

6-8 Crosswalk

-		-	
2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.	2-A-6-10	Use an iterative design process (e.g., define the problem, generate ideas, build, test, and improve solutions) to solve problems, both independently and collaboratively.
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.	2-A-5-7	Create variables that represent different types of data and manipulate their values.
2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	2-A-5-6	Develop programs, both independently and collaboratively, that include sequences with nested loops and multiple branches. [Clarification: At this level, students may use block-based and/or text-based programming languages.]
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	2-A-3-9	Decompose a problem into parts and create solutions for each part.
2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.	2-A-4-8	Define and use procedures that hide the complexity of a task and can be reused to solve similar tasks. [Clarification: Students use and modify, but do not necessarily create, procedures with parameters.]
2-AP-15	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	2-A-2-1	Solicit and integrate peer feedback as appropriate to develop or refine a program.
2-AP-16	Incorporate existing code, media, and libraries into original programs, and give attribution.	2-A-7-3	Provide proper attribution when code is borrowed or built upon.
2-AP-17	Systematically test and refine programs using a range of test cases.	2-D-5-16	Revise computational models to more accurately reflect real-world systems (e.g., ecosystems, epidemics, spread of ideas).
2-AP-18	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	2-A-5-5	Design, develop, and present computational artifacts such as mobile applications that address social problems both independently and collaboratively.
2-AP-19	Document programs in order to make them easier to follow, test, and debug.		New

6-8 Crosswalk

6-8 Final 2018	Level 2	6-8 Interim 2016	Level 2
2-IC-20	Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	2-C-4-12	Analyze the relationship between a device's computational components and its capabilities. [Clarification: Computing Systems include not only computers, but also cars, microwaves, smartphones, traffic lights, and flash drives.]
2-IC-21	Discuss issues of bias and accessibility in the design of existing technologies.	2-I-7-18	Summarize negative and positive impacts of using data and information to categorize people, predict behavior, and make recommendations based on those predictions (e.g., customizing search results or targeted advertising, based on previous browsing history, can save search time and limit options at the same time).
		2-I-6-23	Redesign a computational artifact to remove barriers to universal access (e.g., using captions on images, high contrast colors, and/or larger font sizes).
2-IC-22	Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	2-I-7-19	Explain how computer science fosters innovation and enhances nearly all careers and disciplines.
2-IC-23	Describe tradeoffs between allowing information to be public and keeping information private and secure.	2-N-7-24	Summarize security risks associated with weak passwords, lack of encryption, insecure transactions, and persistence of data.
	Removed	2-A-7-2	Compare different algorithms that may be used to solve the same problem, but one might be faster than the other. (e.g., different algorithms solve the same problem, but one might be faster than the other). [Clarification: Students are not expected to quantify these differences.]
	Removed	2-D-7-14	Describe how different formats of stored data represent tradeoffs between quality and size. [Clarification: compare examples of music, text and/or image formats.]
	Removed	2-I-1-20	Provide examples of how computational artifacts and devices impact health and wellbeing, both positively and negatively.

CSTA & ACM, (2016), *Interim CSTA K-12 Computer Science Standards*

6-8 Crosswalk

6-8 Final 2018	Level 2	6-8 Interim 2016	Level 2
	Removed	2-I-1-21	Describe ways in which the Internet impacts global communication and collaborating.

CSTA & ACM, (2016), *Interim CSTA K-12 Computer Science Standards*

9-10 Crosswalk

9–10 Final	Level 3A	9–10 Interim	Level 3A
3A-CS-01	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	3A-A-4-9	Demonstrate the value of abstraction for managing problem complexity (e.g., using a list instead of discrete variables).
3A-CS-02	Compare levels of abstraction and interactions between application software, system software, and hardware layers.	3A-A-4-7	Understand the notion of hierarchy and abstraction in high-level languages, translation, instruction sets, and logic circuits.
3A-CS-03	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	3A-A-6-12	Use a systematic approach and debugging tools to independently debug a program (e.g., setting breakpoints, inspecting variables with a debugger).
3A-NI-04	Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing	3A-N-7-30	Describe key protocols and underlying processes of Internet-based services (e.g., http/https and SMTP/IMAP, routing protocols).
		3A-N-4-31	Illustrate the basic components of computer networks (e.g., draw logical and topological diagrams of networks including routers, switches, servers, and end user devices; create model with string and paper).
3A-NI-05	Give examples to illustrate how sensitive data can be affected by malware and other attacks.	3A-N-3-33	Explain the principles of information security (confidentiality, integrity, availability) and authentication techniques
3A-NI-06	Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	3A-N-6-35	Identify digital and physical strategies to secure networks and discuss the tradeoffs between ease of access and need for security.
3A-NI-07	Compare various security measures, considering tradeoffs between the usability and security of a computing system.	3A-N-1-32	Compare and contrast multiple viewpoints on cybersecurity (e.g., from the perspective of security experts, privacy advocates, the government).

CSTA & ACM, (2016), *Interim CSTA K-12 Computer Science Standards*

9-10 Crosswalk

9–10 Final	Level 3A	9–10 Interim	Level 3A
3A-NI-08	Explain tradeoffs when selecting and implementing cybersecurity recommendations	3A-N-3-33	Explain the principles of information security (confidentiality, integrity, availability) and authentication techniques
3A-DA-09	Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.	3A-D-4-18	Convert between binary, decimal, and hexadecimal representations of data (e.g., convert hexadecimal color codes to decimal percentages, ASCII/Unicode representation).
3A-DA-10	Evaluate the tradeoffs in how data elements are organized and where data is stored.	3A-D-4-19	Analyze the representation tradeoffs among various forms of digital information (e.g., lossy versus lossless compression, encrypted vs. unencrypted, various image representations).
		3A-D-3-20	Discuss techniques used to store, process, and retrieve different amounts of information (e.g., files, databases, data warehouses).
3A-DA-11	Create interactive data visualizations using software tools to help others better understand real-world phenomena.	3A-A-3-11	Explain and demonstrate how modeling and simulation can be used to explore natural phenomena (e.g., flocking behaviors, queueing, life cycles).
3A-DA-12	Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.	3A-D-5-17	Create computational models that simulate real-world systems (e.g., ecosystems, epidemics, spread of ideas).
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.	3A-D-3-21	Apply basic techniques for locating and collecting small- and large-scale data sets (e.g., creating and distributing user surveys, accessing real-world data sets).
3A-AP-14	Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.		New

CSTA & ACM, (2016), *Interim CSTA K-12 Computer Science Standards*

9-10 Crosswalk

9–10 Final	Level 3A	9–10 Interim	Level 3A
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.		New
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	3A-A-5-4	Design, develop, and implement a computing artifact that responds to an event (e.g., robot that responds to a sensor, mobile app that responds to a text message, sprite that responds to a broadcast).
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.	3A-A-4-8	Deconstruct a complex problem into simpler parts using predefined constructs (e.g., functions and parameters and/or classes).
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.		New
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.	3A-A-2-2	Demonstrate how diverse collaborating impacts the design and development of software products (e.g., discussing real-world examples of products that have been improved through having a diverse design team or reflecting on their own team's development experience).
3A-AP-20	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	3A-A-7-3	Compare and contrast various software licensing schemes (e.g., open source, freeware, commercial).
3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.	3A-A-5-5	Use user-centered research and design techniques (e.g., surveys, interviews) to create software solutions

9-10 Crosswalk

9–10 Final	Level 3A	9–10 Interim	Level 3A
		3A-C-7-13	Develop and apply criteria (e.g., power consumption, processing speed, storage space, battery life, cost, operating system) for evaluating a computer system for a given purpose (e.g., system specification needed to run a game, web browsing, graphic design or video editing).
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.	3A-A-2-1	Design and develop a software artifact working in a team.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.		New
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.	3A-I-1-26	Compare and debate the positive and negative impacts of computing on behavior and culture (e.g., evolution from hitchhiking to ridesharing apps, online accommodation rental services).
3A-IC-25	Test and refine computational artifacts to reduce bias and equity deficits.	3A-I-6-29	Redesign user interfaces (e.g., webpages, mobile applications, animations) to be more inclusive, accessible, and minimizing the impact of the designer's inherent bias.
3A-IC-26	Demonstrate ways a given algorithm applies to problems across disciplines.	3A-I-7-25	Describe how computation shares features with art and music by translating human intention into an artifact.
3A-IC-27	Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.	3A-I-1-27	Demonstrate how computing enables new forms of experience, expression, communication, and collaborating.

9-10 Crosswalk

9–10 Final	Level 3A	9–10 Interim	Level 3A
3A-IC-28	Explain the beneficial and harmful effects that intellectual property laws can have on innovation	3A-I-2-22	Debate the social and economic implications associated with ethical and unethical computing practices (e.g., intellectual property rights, hacktivism, software piracy, diesel emissions testing scandal, new computers shipped with malware).
3A-IC-29	Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.	3A-I-7-24	Discuss implications of the collection and large-scale analysis of information about individuals (e.g., how businesses, social media, and government collect and use personal data).
3A-IC-30	Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.	3A-I-7-23	Compare and contrast information access and distribution rights.
	Removed	3A-I-1-28	Explain the impact of the digital divide (i.e., uneven access to computing, computing education, and interfaces) on access to critical information.
	Removed	3A-A-5-6	Integrate grade-level appropriate mathematical techniques, concepts, and processes in the creation of computing artifacts.
	Removed	3A-A-3-10	Design algorithms using sequence, selection, and iteration.
	Removed	3A-C-5-14	Create, extend, or modify existing programs to add new features and behaviors using different forms of inputs and outputs (e.g., inputs such as sensors, mouse clicks, data sets; outputs such as text, graphics, sounds).

9-10 Crosswalk

9–10 Final	Level 3A	9–10 Interim	Level 3A
	Removed	3A-C-4-15	Demonstrate the role and interaction of a computer embedded within a physical system, such as a consumer electronic, biological system, or vehicle, by creating a diagram, model, simulation, or prototype.
	Removed	3A-C-4-16	Describe the steps necessary for a computer to execute high compilation to machine language, interpretation, fetch-decode-execute https://www.cise.ufl.edu/~mssz/CompOrg/CDAintro.html.
	Removed	3A-N-3-34	Use simple encryption and decryption algorithms to transmit/receive an encrypted message.

11-12 Final	Level 3B	11–12 Interim	Level 3B
Crosswalks for the 11-12 grade band were not completed			

CSTA & ACM, (2016), *Interim CSTA K-12 Computer Science Standards*