Standards Alignment Review

SUGGESTED CITATION: Computer Science Teachers Association. (2026). *Standards alignment review*. https://csteachers.org/k12standards/research.

Curriculum Alignment to Standards

We analyzed eight widely used K–12 Computer Science curricula (see Table 1) for their alignment to current Computer Science Teachers Association (CSTA) standards, prioritizing those reported as frequently used in the CSTA Teacher Landscape Survey. The curriculum selection was reviewed by the CSTA team to ensure that it included curricula highly regarded by educators and aligned with best practices in CS education. We used this approach to enhance the generalizability of our findings. However, we recognize that some less commonly used curricula might offer superior coverage of certain standards overlooked by the most popular curricula. For this reason, we also triangulated our findings with an analysis of the concept coverage of 79 curricula verified by CSforALL's AlignCS project.

In conducting the curriculum-alignment-to-standards analysis, we relied on each curriculum provider's standards alignment documentation. The providers varied in what portion of their curriculum they applied standards to (e.g., activity, lesson); we refer to these portions as *nodes*.

We made two modifications to the curriculum providers' documentation:

- Scratch Jr. and some parts of Bootstrap were aligned to the K–12 Computer Science Framework. We mapped these nodes to the equivalent CSTA standards.
- Bootstrap mislabeled two standards. It used 3B-NI-05 for what is actually 3B-DA-05, and it used 3B-NI-07 for what is actually 3B-DA-07. We used the appropriate labels in our analysis file.

First we give overall findings from our alignment analysis; then we highlight key trends for topic areas. A comprehensive alignment can be found in the accompanying spreadsheet *Standards to Curricula Mapping*.

Table 1. *Curricula Included in the Standards Alignment Review*

Elementary Level	Middle School Level	High School Level
• CS First • Code.org: CS Fundamentals • Scratch Jr.	• Code.org: CS Discoveries • Beauty and Joy of Computing (BJC) Sparks • Bootstrap: Data Literacy	• CodeHS: CS A (Nitro) • Code.org: CS Principles

Table 2. *Percentages of Nodes Applying Standards in Each Subconcept (per Curriculum)*

		Elementary			Middle			High	
		CS First	Code.org: CS Fundamentals	Scratch Jr.	Code.org: CS Discoveries	BJC Sparks	Bootstrap: Data Literacy	CodeHS: CS A (Nitro)	Code.org: CS Principles
Concept	**Subconcept \ Total Nodes**	**19**	**97**	**72**	**158**	**19**	**21**	**76**	**85**
Computing Systems	Devices	0%	6%	17%	9%	11%	0%	3%	0%
	Hardware & Software	0%	4%	14%	11%	5%	0%	0%	2%
	Troubleshooting	0%	7%	22%	8%	11%	0%	1%	1%
Networks & the Internet	Network Communication & Organization	0%	1%	0%	0%	5%	0%	0%	12%
	Cybersecurity	0%	8%	0%	1%	5%	0%	0%	2%
Data & Analysis	Storage	0%	2%	0%	4%	11%	0%	0%	12%
	Collection, Visualization, & Transformation	11%	4%	0%	6%	11%	62%	0%	6%
	Inference & Models	21%	5%	0%	3%	0%	43%	0%	2%
Algorithms & Programming	Algorithms	37%	14%	47%	11%	0%	5%	13%	18%
	Variables	37%	40%	0%	24%	5%	10%	8%	16%
	Control	100%	52%	17%	14%	5%	14%	3%	16%
	Modularity	100%	69%	20%	35%	21%	19%	13%	19%
	Program Development	100%	53%	0%	54%	26%	10%	8%	22%
Impacts of Computing	Culture	0%	5%	0%	15%	5%	14%	1%	7%
	Social Interactions	0%	5%	0%	11%	11%	5%	0%	4%
	Safety, Law, & Ethics	21%	6%	0%	4%	0%	10%	1%	6%

Table 3. *Numbers of Times Standards in Each Subconcept Were Applied (per Curriculum)*

		Elementary			Middle			High	
		CS First	Code.org: CS Fundamentals	Scratch Jr.	Code.org: CS Discoveries	BJC Sparks	Bootstrap: Data Literacy	CodeHS: CS A (Nitro)	Code.org: CS Principles
Concept	**Subconcept \ Total Nodes**	**19**	**97**	**72**	**158**	**19**	**21**	**76**	**85**
Computing Systems	Devices	0	6	12	15	2	0	2	0
	Hardware & Software	0	4	10	18	1	0	0	2
	Troubleshooting	0	7	16	14	2	0	1	1
Networks & the Internet	Network Communication & Organization	0	1	0	0	1	0	0	21
	Cybersecurity	0	8	0	2	1	0	0	4
Data & Analysis	Storage	0	2	0	7	2	0	0	13
	Collection, Visualization, & Transformation	2	4	0	9	2	31	0	14
	Inference & Models	4	5	0	5	0	16	0	2
Algorithms & Programming	Algorithms	8	14	34	17	0	1	12	17
	Variables	13	39	0	39	1	4	8	14
	Control	31	51	12	22	1	5	2	15
	Modularity	49	78	16	64	5	13	16	32
	Program Development	30	74	0	187	7	13	6	30
Impacts of Computing	Culture	0	6	0	29	1	11	1	11
	Social Interactions	0	5	0	17	2	1	0	3
	Safety, Law, & Ethics	4	6	0	6	0	9	1	12

Table 4. *Percentages of CSforALL Vetted Curricula That Include a Focus on Each Concept*

Concept	K–2	3–5	6–8	9–12	Total
Computing Systems	63%	63%	62%	65%	59%
Networks & the Internet	47%	50%	48%	41%	42%
Data & Analysis	58%	53%	52%	56%	54%
Algorithms & Programming	100%	97%	96%	94%	96%
Impacts of Computing	68%	67%	66%	65%	63%

Takeaways for Computing Systems and Networks & the Internet

- **Trend 1.** Standards for Computing Systems and for Networks & the Internet are covered less than standards for Algorithms & Programming, but are covered at a similar level as standards for Impacts of Computing (see Table 2 and Table 3).
- **Trend 2.** Standards for Computing Systems are covered more than standards for Networks & the Internet (see Table 2 and Table 3).
- **Trend 3.** Trends 1 and 2 are similar to trends seen in the CSforALL AlignCS project (see Table 4), suggesting that our analysis of the eight selected curricula may be somewhat representative of broader curriculum trends.
- **Trend 4.** Computing Systems and Networks & the Internet standards for Level 1A (grades K–2) and Level 2 (grades 6–8) are covered more than standards for Level 1B (grades 3–5) and Levels 3A and 3B (high school) (see Table 5).
- **Trend 5.** Some standards were rarely applied. Seven of the 14 standards for Computing Systems and 9 of the 13 standards for Networks & the Internet appeared two times or less across all eight curricula (see Table 6 and Table 7).
- **Trend 6.** 82 nodes applied a Computing Systems standard. Less than a third (n = 23) of nodes applied multiple subconcepts at the same time (see Table 8).

Table 5. *Numbers of Times Each Computing Systems and Networks & the Internet Standard Was Applied (All Curricula)*

		Level 1A (by the end of grade 2)		Level 1B (by the end of grade 5)		Level 2 (by the end of grade 8)		Level 3A (by the end of grade 10)		Level 3B (by the end of grade 12)	
Concept	**Subconcept**	**Standard**	***n***	**Standard**	***n***	**Standard**	***n***	**Standard**	***n***	**Standard**	***n***
Computing Systems	Devices	1A-CS-01	16	1B-CS-01	2	2-CS-01	17	3A-CS-01	2		
	Hardware & Software	1A-CS-02	12	1B-CS-02	2	2-CS-02	19	3A-CS-02	2	3B-CS-01	0
	Troubleshooting	1A-CS-03	18	1B-CS-03	5	2-CS-03	16	3A-CS-03	2	3B-CS-02	0
Networks & the Internet	Network Communication & Organization			1B-NI-04	1	2-NI-04	7	3A-NI-04	6	3B-NI-03	5
	Cybersecurity	1A-NI-04	2	1B-NI-05	6	2-NI-05	1	3A-NI-05	2	3B-NI-04	2
						2-NI-06	2	3A-NI-06	2		
								3A-NI-07	2		
								3A-NI-08	0		

Table 6. *Computing Systems Standards Rarely Used in the Reviewed Curricula*

Standard	Description	Times Used
1B-CS-01	Describe how internal and external parts of computing devices function to form a system.	2
1B-CS-02	Model how computer hardware and software work together as a system to accomplish tasks.	2
3A-CS-01	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.	2
3A-CS-02	Compare levels of abstraction and interactions between application software, system software, and hardware layers.	2
3A-CS-03	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.	2
3B-CS-01	Categorize the roles of operating system software.	0
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.	0

Table 7. *Networks & the Internet Standards Rarely Used in the Reviewed Curricula*

Standard	Description	Times Used
1A-NI-04	Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access.	2
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.	1
2-NI-05	Explain how physical and digital security measures protect electronic information.	1
2-NI-06	Apply multiple methods of encryption to model the secure transmission of information.	2
3A-NI-05	Give examples to illustrate how sensitive data can be affected by malware and other attacks.	2
3A-NI-06	Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts.	2
3A-NI-07	Compare various security measures, considering tradeoffs between the usability and security of a computing system.	2
3A-NI-08	Explain tradeoffs when selecting and implementing cybersecurity recommendations.	0
3B-NI-04	Compare ways software developers protect devices and information from unauthorized access.	2

Table 8. *Curricula That Applied Multiple Computing Systems Standards to the Same Lesson*

		Code.org: CS Fundamentals (Elementary)		Scratch Jr. (Elementary)			Code.org: CS Discoveries (Middle)
Subconcept	**Standard**	**Course A**	**Course D**	**K**	**1st**	**2nd**	**Unit 6A**
Devices	**1A-CS-01** Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use.			●	●	●	
	1B-CS-01 Describe how internal and external parts of computing devices function to form a system.		●				
	2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.						●
Hardware & Software	**1A-CS-02** Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).	●		●	●	●	
	1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks.		●				
	2-CS-02 Design projects that combine hardware and software components to collect and exchange data.						●
Troubleshooting	**1A-CS-03** Describe basic hardware and software problems using accurate terminology.	●			●		
	2-CS-03 Systematically identify and fix problems with computing devices and their components.						●

Takeaways for Data & Analysis

- **Trend 1.** For most curricula, standards for Data & Analysis are covered less than standards for Algorithms & Programming, but are covered at a similar level as standards for Impacts of Computing (see Table 2 and Table 3). An exception to this trend is Bootstrap's Data Literacy course, which is designed to address data science topics.
- **Trend 2.** Standards for Collection, Visualization & Transformation are covered more than standards for Storage and for Inference & Models (see Table 9).
- **Trend 3.** There are 15 standards for Data & Analysis. Four of them appeared two times or less across all eight curricula. Three of these represent all the Data & Analysis standards at Level 1A:
  - » 1A-DA-05 Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data.
  - » 1A-DA-06 Collect and present the same data in various visual formats.
  - » 1A-DA-07 Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.
  - » 3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.
  - » 3B-DA-07 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.

Table 9. *Numbers of Times Each Data & Analysis Standard Was Applied (All Curricula)*

		Level 1A (by the end of grade 2)		Level 1B (by the end of grade 5)		Level 2 (by the end of grade 8)		Level 3A (by the end of grade 10)		Level 3B (by the end of grade 12)	
Concept	**Subconcept**	**Standard**	***n***	**Standard**	***n***	**Standard**	***n***	**Standard**	***n***	**Standard**	***n***
Data & Analysis	Storage	1A-DA-05	2			2-DA-07	12	3A-DA-09	7		
								3A-DA-10	3		
	Collection, Visualization, & Transformation	1A-DA-06	2	1B-DA-06	9	2-DA-08	19	3A-DA-11	21	3B-DA-05	7
										3B-DA-06	4
	Inference & Models	1A-DA-07	1	1B-DA-07	11	2-DA-09	12	3A-DA-12	2	3B-DA-07	1

Overlapping Standards

A total of 66 nodes applied a Data & Analysis standard; 14 applied multiple subconcepts at the same time.

Table 10. *Overlapping Data & Analysis Standards by Curriculum*

Curriculum	Overlapping Standards
Bootstrap: Data Literacy course (Middle School)	Eight nodes overlapped in Collection, Visualization, & Transformation and Inference & Models: • CV&T standards: 1B-DA-06, 2-DA-08, 3A-DA-11, 3B-DA-05 • I&M standards: 1B-DA-07, 2-DA-09, 3A-DA-12, 3B-DA-07
Code.org: CS Discoveries Unit 4: The Design Process (Middle School)	One lesson used both 2-DA-08 (CV&T) and 2-DA-09 (I&M).
CS First > Sports (Elementary School)	One lesson used both 2-DA-08 (CV&T) and 1B-DA-07 (I&M).
Code.org: CS Fundamentals Course C: Data (Elementary School)	Two lessons used both 1A-DA-05 (Storage) and 1A-DA-06 (CV&T). One of these two lessons also used 1A-DA-07 (I&M).
Code.org: CS Fundamentals Courses E and F (Elementary School)	One lesson in each course used both 1B-DA-06 (CV&T) and 1B-DA-07 (I&M).

Takeaways for Algorithms & Programming

- **Trend 1.** The Algorithms & Programming topic area received the most coverage across all reviewed curricula with the exception of Bootstrap's Data Literacy course (see Table 2 and Table 3). This trend is similar to trends seen in the CSforALL AlignCS project (see Table 4).
- **Trend 2.** Elementary curricula seemed to have uneven distributions of standards across subconcepts (e.g., focusing heavily on some subconcepts while barely covering other subconcepts). In contrast, high school curricula applied more even distributions of standards across subconcepts (see Table 2 and Table 3).
- **Trend 3.** In the elementary curricula, Algorithms and Variables were more frequently covered at the K–2 levels while Control, Modularity, and Program Development were more frequently covered at the 3–5 levels (see Table 11).
- **Trend 4.** Most standards within the Algorithms & Programming topic area were addressed across all eight curricula. Only 11 of the 56 standards, all at the high school level, appeared two times or less (see Table 12).
- **Trend 5.** A total of 334 nodes applied an Algorithms & Programming standard. Many nodes (*n* = 199; 60%) applied multiple subconcepts at the same time. The curricula varied in how they applied multiple standards across their nodes. Table 13 shows the most commonly co-occurring standards by subconcept.

Table 11. *Numbers of Times Each Algorithms & Programming Standard Was Applied (All Curricula)*

		Level 1A (by the end of grade 2)		Level 1B (by the end of grade 5)		Level 2 (by the end of grade 8)		Level 3A (by the end of grade 10)		Level 3B (by the end of grade 12)	
Concept	**Subconcept**	**Standard**	***n***	**Standard**	***n***	**Standard**	***n***	**Standard**	***n***	**Standard**	***n***
Algorithms & Programming	Algorithms	1A-AP-08	39	1B-AP-08	16	2-AP-10	24	3A-AP-13	6	3B-AP-08	1
										3B-AP-09	2
										3B-AP-10	8
										3B-AP-11	7
	Variables	1A-AP-09	30	1B-AP-09	17	2-AP-11	53	3A-AP-14	15	3B-AP-12	3
	Control	1A-AP-10	27	1B-AP-10	56	2-AP-12	40	3A-AP-15	6	3B-AP-13	1
								3A-AP-16	9		
	Modularity	1A-AP-11	52	1B-AP-11	45	2-AP-13	67	3A-AP-17	9	3B-AP-15	1
				1B-AP-12	38	2-AP-14	25	3A-AP-18	11	3B-AP-14	14
										3B-AP-16	9
	Program Development	1A-AP-12	4	1B-AP-13	10	2-AP-15	27	3A-AP-19	2	3B-AP-17	0
		1A-AP-13	4	1B-AP-14	5	2-AP-16	35	3A-AP-20	0	3B-AP-22	3
		1A-AP-14	12	1B-AP-15	41	2-AP-17	60	3A-AP-21	6	3B-AP-21	4
				1B-AP-16	10	2-AP-18	17	3A-AP-22	5	3B-AP-20	0
		1A-AP-15	3	1B-AP-17	10	2-AP-19	74	3A-AP-23	7	3B-AP-23	10
										3B-AP-18	0
										3B-AP-19	0
										3B-AP-24	0

Table 12. *Algorithms & Programming Standards Rarely Used in the Reviewed Curricula*

Subconcept	Standard	Description	Times Used
Algorithms	**3B-AP-08**	Describe how artificial intelligence drives many software and physical systems.	1
	3B-AP-09	Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.	2
Control	**3B-AP-13**	Illustrate the flow of execution of a recursive algorithm.	1
Modularity	**3B-AP-15**	Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.	1
Program Development	**3A-AP-19**	Systematically design and develop programs for broad audiences by incorporating feedback from users.	2
	3A-AP-20	Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries.	0
	3B-AP-17	Plan and develop programs for broad audiences using a software life cycle process.	0
	3B-AP-20	Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.	0
	3B-AP-18	Explain security issues that might lead to compromised computer programs.	0
	3B-AP-19	Develop programs for multiple computing platforms.	0
	3B-AP-24	Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.	0

Table 13. *Most Commonly Applied Algorithms & Programming Standards That Co-Occurred With Standards in Another Subconcept*

Subconcept	Standard	Description	Number of Co-Occurrences
Algorithms	**1A-AP-08**	Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.	18
	1B-AP-08	Compare and refine multiple algorithms for the same task and determine which is the most appropriate.	15
	2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.	21
Variables	**1A-AP-09**	Model the way programs store and manipulate data by using numbers or other symbols to represent information.	28
	1B-AP-09	Create programs that use variables to store and modify data.	16
	2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.	50
Control	**1A-AP-10**	Develop programs with sequences and simple loops, to express ideas or address a problem.	20
	1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	55
	2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	38
Modularity	**1A-AP-11**	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.	43
	1B-AP-11	Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.	45
	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.	38
	2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	63

Subconcept	Standard	Description	Number of Co-Occurrences
Program Development	1B-AP-15	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	41
	2-AP-16	Incorporate existing code, media, and libraries into original programs, and give attribution.	25
	2-AP-17	Systematically test and refine programs using a range of test cases.	52
	2-AP-19	Document programs in order to make them easier to follow, test, and debug.	57

