

Computer Science
K-12 Standards
Computing Systems



License



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/). Accordingly, individuals and organizations are free to share and adapt the materials in whole or in part, as long as they provide proper attribution, do not use for commercial purposes, and share contributions or derivations under the same license.

Attribution



The CSTA K–12 Computer Science Standards are created and maintained by members of the Computer Science Teachers Association (CSTA).



The Association for Computing Machinery (ACM) founded CSTA as part of its commitment to K–12 computer science education. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Suggested citation: Computer Science Teachers Association (2017). CSTA K–12 Computer Science Standards, Revised 2017. Retrieved from <http://www.csteachers.org/standards>.



The [K–12 Computer Science Framework](#), led by the [Association for Computing Machinery](#), [Code.org](#), [Computer Science Teachers Association](#), [Cyber Innovation Center](#), and [National Math and Science Initiative](#) in partnership with states and districts, informed the development of this work.

The CSTA Standards Revision Task Force crafted standards by combining concept statements and practices from the Framework. The Task Force also used descriptive material from the Framework when writing examples and clarifying statements to accompany the standards. The glossary referenced in the navigation header links directly to the Framework's glossary.

For more information about the Framework, please visit k12cs.org.

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

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

The identifier code corresponds to: Level – Concept – Number		
Identifier Code		Key
Levels	1A	Grades K–2
	1B	Grades 3–5
	2	Grades 6–8
	3A	Grades 9–10
	3B	Grades 11–12
Concepts	CS	Computing Systems
	NI	Networks and the Internet
	DA	Data and Analysis
	AP	Algorithms and Programming
	IC	Impacts of Computing

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

Figure 1: Standards Identifier Code –
Computer Science Teachers Association K–12 Computer Science Standards (2017)
Retrieved from <http://www.csteachers.org>



K-12 Computing Systems Standards

Level 1A: K–2	
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-CS-02	Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).
1A-CS-03	Describe basic hardware and software problems using accurate terminology.
Level 1B: 3–5	
1B-CS-01	Describe how internal and external parts of computing devices function to form a system.
1B-CS-02	Model how computer hardware and software work together as a system to accomplish tasks.
1B-CS-03	Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.
Level 2: 6–8	
2-CS-01	Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.
2-CS-02	Design projects that combine hardware and software components to collect and exchange data.
2-CS-03	Systematically identify and fix problems with computing devices and their components.
Level 3A: 9–10	
3A-CS-01	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.
3A-CS-02	Compare levels of abstraction and interactions between application software, system software, and hardware layers.
3A-CS-03	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
Level 3B: 11–12	
3B-CS-01	Categorize the roles of operating system software.
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.

OSPI provides equal access to all programs and services without discrimination based on sex, race, creed, religion, color, national origin, age, honorably discharged veteran or military status, sexual orientation, gender expression, gender identity, disability, or the use of a trained dog guide or service animal by a person with a disability. Questions and complaints of alleged discrimination should be directed to the Equity and Civil Rights Director at 360-725-6162; TTY: 360-664-3631; or P.O. Box 47200, Olympia, WA 98504-7200; or equity@k12.wa.us.

Download this material in PDF at <http://www.k12.wa.us/CurriculumInstruct/learningstandards.aspx>.

Please refer to this document number for quicker service: 16-0075.



Chris Reykdal • State Superintendent
Office of Superintendent of Public Instruction
Old Capitol Building • P.O. Box 47200
Olympia, WA 98504-7200