

# Computer Science 6-8 Standards



## 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](http://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. |
| <b>2</b>   | <b>AP</b>                  | <b>17</b> | <b>Identifier: 2-AP-17</b>   |

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

| The identifier code corresponds to:<br>Level – Concept – Number |    |                            |
|---|----|----------------------------|
| Identifier Code   |    | Key                        |
| <b>Levels</b>   | 1A | Grades K–2                 |
|   | 1B | Grades 3–5                 |
|   | 2  | Grades 6–8                 |
|   | 3A | Grades 9–10                |
|   | 3B | Grades 11–12               |
| <b>Concepts</b>   | 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:

|                  |   |   |
|------------------|---|---|
| <b>Practices</b> | 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>



## Standards

| Identifier | 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. (P. 3.3)                |
| 2-CS-02    | Design projects that combine hardware and software components to collect and exchange data. (P. 5.1)  |
| 2-CS-03    | Systematically identify and fix problems with computing devices and their components. (P. 6.2)  |
| 2-NI-04    | Model the role of protocols in transmitting data across networks and the Internet. (P. 4.4)   |
| 2-NI-05    | Explain how physical and digital security measures protect electronic information. (P. 7.2)   |
| 2-NI-06    | Apply multiple methods of encryption to model the secure transmission of information. (P. 4.4)  |
| 2-DA-07    | Represent data using multiple encoding schemes. (P. 4)  |
| 2-DA-08    | Collect data using computational tools and transform the data to make it more useful and reliable. (P. 6.3)                                     |
| 2-DA-09    | Refine computational models based on the data they have generated. (P. 5.3, P. 4.4)   |
| 2-AP-10    | Use flowcharts and/or pseudocode to address complex problems as algorithms. (P. 4.4, 4.1)   |
| 2-AP-11    | Create clearly named variables that represent different data types and perform operations on their values. (P. 5.1, P. 5.2)                     |
| 2-AP-12    | Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. (P. 5.1, P. 5.2)     |
| 2-AP-13    | Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. (P. 3.2)                        |
| 2-AP-14    | Create procedures with parameters to organize code and make it easier to reuse. (P. 4.1, P. 4.3)  |
| 2-AP-15    | Seek and incorporate feedback from team members and users to refine a solution that meets user needs. (P. 2.3, P. 1.1)                          |
| 2-AP-16    | Incorporate existing code, media, and libraries into original programs, and give attribution. (P. 4.2, P. 5.2, P. 7.3)                          |
| 2-AP-17    | Systematically test and refine programs using a range of test cases. (P. 6.1)   |
| 2-AP-18    | Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. (P. 2.2)                              |
| 2-AP-19    | Document programs in order to make them easier to follow, test, and debug. (7.2)  |
| 2-IC-20    | Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. (P. 7.2)                  |
| 2-IC-21    | Discuss issues of bias and accessibility in the design of existing technologies. (1.2)  |
| 2-IC-22    | Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact. (P. 2.4, P. 5.2) |
| 2-IC-23    | Describe tradeoffs between allowing information to be public and keeping information private and secure. (P. 7.2)                               |

## Computer Science Glossary

*The following glossary includes definitions of terms used in the statements in the Washington Computer Science K–12 Learning Standards. These terms are intended to increase teacher understanding and decrease biased language.*

**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. In elementary classrooms, abstraction is hiding unnecessary details to make it easier to think about a problem.

**abstraction (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. [MA-DLCS]

**abstraction (Code.org K–5)** Pulling out specific differences to make one solution work for multiple problems.

**algorithm:** A step-by-step process to complete a task.

A list of steps to finish a task. A set of instructions that can be performed with or without a computer. For example, the collection of steps to make a peanut butter and jelly sandwich is an algorithm. (Code.org K–5)

**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 computer science-specific definition.

**automation:** To link disparate systems and software in such a way that they become self-acting or self-regulating.

**backup:** The process of making copies of data or data files to use in the event the original data or data files are lost or destroyed. [Techopedia]

**binary:** A method of encoding data using two symbols (usually 1 and 0). To illustrate binary encoding, we can use any two symbols. [MA-DLCS]

A way of representing information using only two options. (Code.org K–5)

**Block-based programming language:** (Code.org K–5) Any programming language that lets users create programs by manipulating “blocks” or graphical programming elements, rather than writing code using text. Examples include Code Studio, Scratch, and Swift. (Sometimes called visual coding, drag and drop programming, or graphical programming blocks)

**bug:** An error in a software program. It may cause a program to unexpectedly quit or behave in an unintended manner. [TechTerms] The process of removing errors (bugs) is called debugging.

An error in a program that prevents the program from running as expected. (Code.org K–5)

**cloud:** Remote servers that store data and are accessed from the Internet. [Techopedia]

**code:** Any set of instructions expressed in a programming language. [MA-DLCS] One or more commands or algorithm(s) designed to be carried out by a computer. (Code.org K–5) See also: program

**command:** An instruction for the computer. Many commands put together make up algorithms and computer programs. (Code.org K–5)

**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.

**computational thinking:** Mental processes and strategies that include: decomposition, pattern matching, abstraction, algorithms (decomposing problems into smaller, more manageable problems, finding repeating patterns, abstracting specific differences to make one solution work for multiple problems, and creating step-by-step algorithms). (Code.org K–5)

**computer science:** Using the power of computers to solve problems. (Code.org K–5)

**conditionals:** Statements that only run under certain conditions or situations. (Code.org K–5)

**data:** Information. Often, quantities, characters, or symbols that are the inputs and outputs of computer programs. (Code.org K–5)

**debugging:** Finding and fixing errors in programs. (Code.org K–5)

**decompose:** Break a problem down into smaller pieces. (Code.org K–5)

**Digital divide:** the gulf between those who have ready access to computers and the Internet, and those who do not.

**encryption:** The process of encoding messages or information in such a way that only authorized parties can read it.

**event:** An action that causes something to happen. (Code.org K–5)

**execution:** The process of executing an instruction or instruction set.

**for loop:** A loop with a predetermined beginning, end, and increment (step interval) (Code.org K–5)

**function:** A type of procedure or routine. Some programming languages make a distinction between a function, which returns a value, and a procedure, which performs some operation, but does not return a value. [MA-DLCS] *Note: This definition differs from that used in math. A piece of code that you can easily call over and over again. Functions are sometimes called ‘procedures.’* (Code.org K–5)

**hardware:** The physical components that make up a computing system, computer, or computing device. [MA-DLCS]

**hierarchy:** An organizational structure in which items are ranked according to levels of importance. [TechTarget]

**HTTP:** (Hypertext Transfer Protocol) is the set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

**HTTPS:** encrypts and decrypts user page requests as well as the pages that are returned by the Web server. The use of HTTPS protects against eavesdropping and man-in-the-middle attacks.

**input:** The signals or instructions sent to a computer. [Techopedia]

**Internet:** The global collection of computer networks and their connections, all using shared protocols to communicate [CAS-Prim] A group of computers and servers that are connected to each other. (Code.org K–5)

**iterative:** Involving the repeating of a process with the aim of approaching a desired goal, target, or result. [MA-DLCS]

**logic (Boolean):** Boolean logic deals with the basic operations of truth values: AND, OR, NOT and combinations thereof. [FOLDOC]

**loop:** A programming structure that repeats a sequence of instructions as long as a specific condition is true. [TechTerms]

**looping:** Repetition, using a loop. The action of doing something over and over again. (Code.org K–5)

**memory:** Temporary storage used by computing devices. [MA-DLCS]

**model:** A representation of (some part of) a problem or a system. (Modeling (v): the act of creating a model) [MA-DLCS] *Note: This definition differs from that used in science.*

**network:** A group of computing devices (personal computers, phones, servers, switches, routers, and so on) connected by cables or wireless media for the exchange of information and resources.

**nested loop:** A loop within a loop, an inner loop within the body of an outer one.

**operating system:** Software that communicates with the hardware and allows other programs to run. An operating system (or “OS”) is comprised of system software, or the fundamental files a computer needs to boot up and function. Every desktop computer, tablet, and smartphone includes an operating system that provides basic functionality for the device. [TechTerms]

**operation:** An action, resulting from a single instruction, that changes the state of data. [Dictionary.com]

**packets:** Small chunks of information that have been carefully formed from larger chunks of information.

**pair programming:** A technique in which two developers (or students) team together and work on one computer. [TechTarget] The terms “driver” and “navigator” are often used for the two roles. In a classroom setting, teachers often specify that students switch roles frequently (or within a specific period of time).

**paradigm (programming):** A theory or a group of ideas about how something should be done, made, or thought about. A philosophical or theoretical framework of any kind. [Merriam-Webster] Common programming paradigms are object-oriented, functional, imperative, declarative, procedural, logic, and symbolic. [DC, Wikipedia]

**parameter:** A special kind of variable used in a procedure to refer to one of the pieces of data provided as input to the procedure. These pieces of data are called arguments. An ordered list of parameters is usually included in the definition of a subroutine so each time the subroutine is called, its arguments for that call can be assigned to the corresponding parameters. [MA-DLCS]

An extra piece of information that you pass to the function to customize it for a specific need. (Code.org)

**pattern matching:** Finding similarities between things. (Code.org K–5)

**persistence:** Trying again and again, even when something is very hard. (Code.org K–5)

**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 source code. This kind of code item can also be called a function or a subroutine. 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. A procedure may also be referred to as a function, subroutine, routine, method or subprogram. [Techopedia]

**processor:** The hardware within a computer or device that executes a program. The CPU (central processing unit) is often referred to as the brain of a computer.

**program; programming (n):** A set of instructions that the computer executes in order to achieve a particular objective. [MA-DLCS] **program (v):** To produce a program by programming. An algorithm that has been coded into something that can be run by a machine. (Code.org K–5)

**programming:** The craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them. [MA-DLCS] The art of creating a program. (Code.org K–5)

**protocol:** The special set of rules that end points in a telecommunication connection use when they communicate. Protocols specify interactions between the communicating entities. [TechTarget]

**prototype; prototype:** An early approximation of a final product or information system, often built for demonstration purposes. [TechTarget, Techopedia]

**pseudocode:** 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]

**RGB:** (red, green, and blue) Refers to a system for representing the colors to be used on a computer display. Red, green, and blue can be combined in various proportions to obtain any color in the visible spectrum.

**routing; router; routing:** Establishing the path that data packets traverse from source to destination. A device or software that determines the routing for a data packet. [TechTarget]



**run program:** Cause the computer to execute the commands you've written in your program. (Code.org K–5)

**security:** The protection against access to, or alteration of, computing resources, through the use of technology, processes, and training. [TechTarget]

**servers:** Computers that exist only to provide things to others. (Code.org K–5)

**simulate:** to imitate the operation of a real world process or system over time.

**simulation:** Imitation of the operation of a real world process or system over time. [MA-DLCS]

**software:** Programs that run on a computer system, computer, or other computing device.

**storage:** A place (usually a device) into which data can be entered, in which it can be held, and from which it can be retrieved at a later time. [FOLDOC] 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 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 paradigm.

**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] A collection of computing hardware and software integrated for the purpose of accomplishing shared tasks.

**troubleshooting:** A systematic approach to problem solving that is often used to find and resolve a problem, error, or fault within software or a computer system. [Techopedia, TechTarget]

**user:** A person for whom a hardware or software product is designed (as distinguished from the developers). [TechTarget]

**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 the 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-Prim, Techopedia] A placeholder for a piece of information that can change (Code.org K–5) **Note:** *This definition differs from that used in math.*

**wearable computing:** Miniature electronic devices that are worn under, with or on top of clothing.

## **Key to sources of multiple definitions in this glossary:**

CAS-Prim: Computing at School. Computing in the national curriculum: A guide for primary teachers  
(<http://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf>)

Code.org: Creative Commons License (CC BY-NC-SA 4.0)  
(<https://code.org/curriculum/docs/k-5/glossary>)

Computer Science Teachers Association: CSTA K–12 Computer Science Standards (2011)  
<https://csta.acm.org/Curriculum/sub/K12Standards.html>

FOLDOC: Free On-Line Dictionary of Computing. (<http://foldoc.org/>)

MA-DLCS: Massachusetts Digital Literacy and Computer Science Standards, Glossary (Draft, December 2015)

NIST/DADS: National Institute of Science and Technology Dictionary of Algorithms and Data Structures.  
(<https://xlinux.nist.gov/dads/>)

Techopedia: Techopedia. (<https://www.techopedia.com/dictionary>)

TechTarget: TechTarget Network. (<http://www.techtarget.com/network>)

TechTerms: Tech Terms Computer Dictionary. (<http://www.techterms.com>)

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](mailto: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