

Legend for Identifiers

Unique Numbering System for the 2016 K–12 Computer Science Learning Standards

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

Grades	Identifier	Computer Science K–12 Learning Standard	Framework Concept	Framework Practice
9–10	3A-A-2-1	Design and develop a software artifact working in a team.	Algorithms and Programming	Collaborating

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

The identifier code corresponds to: Level – Concept – Practice – Identifier		
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	A	Algorithms and Programming
	C	Computing Systems
	D	Data and Analysis
	I	Impacts of Computing
	N	Networks and the Internet
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 4: Standards Identifier Code - Interim Computer Science Teachers Association K–12 Computer Science Standards (2016)
Retrieved from <http://www.csteachers.org>

9–10	Level 3A
3A-A-2-1	Design and develop a software artifact working in a team.
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-A-7-3	Compare and contrast various software licensing schemes (e.g., open source, freeware, commercial).
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-A-5-5	Use user-centered research and design techniques (e.g., surveys, interviews) to create software solutions
3A-A-5-6	Integrate grade-level appropriate mathematical techniques, concepts, and processes in the creation of computing artifacts.
3A-A-4-7	Understand the notion of hierarchy and abstraction in high-level languages, translation, instruction sets, and logic circuits.
3A-A-4-8	Deconstruct a complex problem into simpler parts using predefined constructs (e.g., functions and parameters and/or classes).
3A-A-4-9	Demonstrate the value of abstraction for managing problem complexity (e.g., using a list instead of discrete variables).
3A-A-3-10	Design algorithms using sequence, selection, and iteration.
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-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-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-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	Level 3A
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.
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 .
3A-D-5-17	Create computational models that simulate real-world systems (e.g., ecosystems, epidemics, spread of ideas).
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-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-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-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-I-7-23	Compare and contrast information access and distribution rights.
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-I-7-25	Describe how computation shares features with art and music by translating human intention into an artifact.
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-I-1-27	Demonstrate how computing enables new forms of experience, expression, communication, and collaborating.
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.
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.

9–10	Level 3A
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-N-1-32	Compare and contrast multiple viewpoints on cybersecurity (e.g., from the perspective of security experts, privacy advocates, the government).
3A-N-3-33	Explain the principles of information security (confidentiality, integrity, availability) and authentication techniques.
3A-N-3-34	Use simple encryption and decryption algorithms to transmit/receive an encrypted message.
3A-N-6-35	Identify digital and physical strategies to secure networks and discuss the tradeoffs between ease of access and need for security.