



2016 Massachusetts Digital Literacy and Computer Science (DLCS) Curriculum Framework

June 2016

Massachusetts Department of Elementary and Secondary Education
75 Pleasant Street, Malden, MA 02148-4906
Phone 781-338-3000 TTY: N.E.T. Relay 800-439-2370
www.doe.mass.edu

DLCS Standards

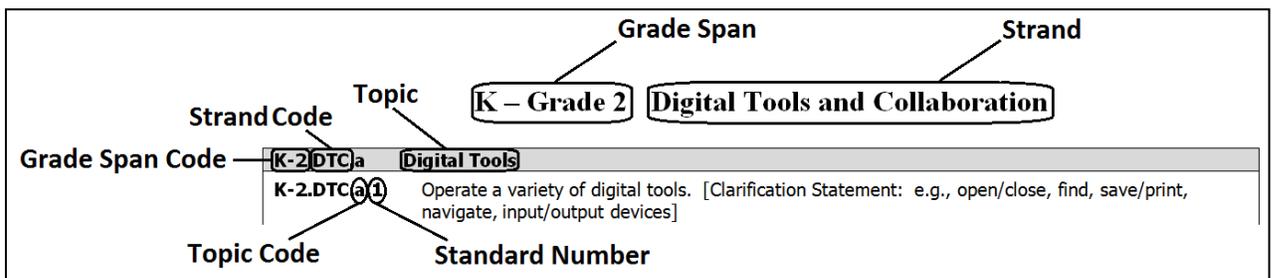
The coding system used for the standards identifies the strand and topic of the standard.

Grade Codes:	
Kindergarten to Grade 2	K-2
Grades 3 to 5	3-5
Grades 6 to 8	6-8
Grades 9 to 12	9-12

Strand Codes:	
Computing and Society	CAS
Digital Tools and Collaboration	DTC
Computing Systems	CS
Computational Thinking	CT

Topic Codes:
Topic codes are the letter of the topic.

Standard Number:
The Standard number is the numerical order of the standard as presented; the order does not imply teaching order.



Grades 9 to 12

The concepts and skills in grades 9–12 build on K–8 experiences and progress to more technical and sophisticated applications. Students continue to refine their skills in differentiating problems or sub-problems that are best solved by computing systems or digital tools and those best solved by humans. Students work independently and collaborative to achieve the high school standards. Students further develop their computational thinking problem solving skills, which facilitate the selection and use of technology. The high school standards provide opportunities for students to gain proficiency and incorporate substantive expectations of the College Board’s *Computer Science Principles*, the widely recognized benchmark for post–secondary preparation. The high school standards specify the skills that all students should study in order to be college and career ready.

Grade 9 to 12 standards integrate all seven practices. Standards in this grade span ask students to demonstrate the ability to:

Computing and Society (CAS)

- Understand safety and security concepts, security and recovery strategies, and how to deal with cyberbullying and peer pressure.
- Analyze the impact and intent of new technology laws.
- Interpret license agreements and permissions.
- Examine the impact of technology, assistive technology, technology proficiencies, and cybercrime in people’s lives, commerce, and society.

Digital Tools and Collaboration (DTC)

- Select and use ‘best’ digital tools or resources to create an artifact or solve a problem.
- Communicate and publish online.
- Advance research skills including advance searches, digital source evaluation, and synthesis of information.

Computing Systems (CS)

- Select and use ‘best’ computing devices to accomplish a real-world task.
- Understand how computing device components work.
- Use troubleshooting strategies to solve routine hardware and software problems.
- Decompose tasks/problems into sub-problems to plan solutions.
- Understand how networks communicate, their vulnerabilities and issues that may impact their functionality.
- Evaluate the benefits of using a service with respect to function and quality.

Computational Thinking (CT)

- Create a new representation through generalization and decomposition.
- Write and debug algorithms in a structured language (pseudocode).
- Understand how different data representation effects storage and quality.
- Create, modify, and manipulate data structures, data sets, and data visualizations.
- Use an iterative design process to create an artifact or solve a problem.
- Create models and simulations to formulate, test, analyze, and refine a hypothesis.

Throughout high school, students should develop increasingly sophisticated skills relevant to their goals for college and career. By the completion of high school, students should have the opportunity to use more specialized computing systems and digital tools, and develop an appreciation for the capabilities and capacities of technology in civic, college, and career contexts. Students should be knowledgeable about the role technology plays in various fields of work, enabling them to better plan for their careers in the twenty-first century.

Grades 9 – 12: Computing and Society (CAS)

9-12.CAS.a	Safety and Security
9-12.CAS.a.1	Evaluate and design an ergonomic work environment.
9-12.CAS.a.2	Explain safe practices when collaborating online, including how to anticipate potentially dangerous situations.
9-12.CAS.a.3	Construct strategies to combat cyberbullying/harassment.
9-12.CAS.a.4	Identify the mental health consequences of cyberbullying/harassment.
9-12.CAS.a.5	Explain how peer pressure in social computing settings influences choices.
9-12.CAS.a.6	Apply strategies for managing negative peer pressure and encouraging positive peer pressure.
9-12.CAS.b	Ethics and Laws
9-12.CAS.b.1	Model mastery of the school’s Acceptable Use Policy [AUP].
9-12.CAS.b.2	Identify computer-related laws and analyze their impact on digital privacy, security, intellectual property, network access, contracts, and consequences of sexting and harassment.
9-12.CAS.b.3	Discuss the legal and ethical implications associated with malicious hacking and software piracy.
9-12.CAS.b.4	Interpret software license agreements and application permissions.
9-12.CAS.c	Interpersonal and Societal Impact
9-12.CAS.c.1	Explain the impact of the digital divide on access to critical information.
9-12.CAS.c.2	Discuss the impact of computing technology on business and commerce (e.g., automated tracking of goods, automated financial transaction, e-commerce, cloud computing).
9-12.CAS.c.3	Describe the role that assistive technology can play in people’s lives.
9-12.CAS.c.4	Create a digital artifact that is designed to be accessible (e.g., closed captioning for audio, alternative text for images).
9-12.CAS.c.5	Analyze the beneficial and harmful effects of computing innovations (e.g., social networking, delivery of news and other public media, intercultural communication).
9-12.CAS.c.6	Cultivate a positive web presence (e.g., digital resume, portfolio, social media).
9-12.CAS.c.7	Identify ways to use technology to support lifelong learning.
9-12.CAS.c.8	Analyze the impact of values and points of view that are presented in media messages (e.g., racial, gender, political).
9-12.CAS.c.9	Discuss the social and economic implications associated with malicious hacking, software piracy, and cyber terrorism.

Grades 9 – 12: Digital Tools and Collaboration (DTC)

9-12.DTC.a	Digital Tools
9-12.DTC.a.1	Use digital tools to design and develop a significant digital artifact (e.g., multipage website, online portfolio, simulation).
9-12.DTC.a.2	Select digital tools or resources based on their efficiency and effectiveness to use for a project or assignment, and justify the selection.
9-12.DTC.b	Collaboration and Communication
9-12.DTC.b.1	Communicate and publish key ideas and details to a variety of audiences using digital tools and media-rich resources.
9-12.DTC.b.2	Collaborate on a substantial project with outside experts or others through online digital tools (e.g., science fair project, community service project, capstone project).
9-12.DTC.c	Research
9-12.DTC.c.1	Generate, evaluate, and prioritize questions that can be researched through digital resources or tools.
9-12.DTC.c.2	Perform advanced searches to locate information and/or design a data-collection approach to gather original data (e.g., qualitative interviews, surveys, prototypes, simulations).
9-12.DTC.c.3	Evaluate digital sources needed to solve a given problem (e.g., reliability, point of view, relevancy).
9-12.DTC.c.4	Gather, organize, analyze, and synthesize information using a variety of digital tools.
9-12.DTC.c.5	Create an artifact that answers a research question, communicates results and conclusions, and cites sources.

Grades 9 - 12: Computing Systems (CS)

9-12.CS.a	Computing Devices
9-12.CS.a.1	Select computing devices (e.g., probe, sensor, tablet) to accomplish a real-world task (e.g., collecting data in a field experiment) and justify the selection.
9-12.CS.a.2	Examine how the components of computing devices are controlled by and react to programmed commands.
9-12.CS.a.3	Apply strategies for identifying and solving routine hardware and software problems that occur in everyday life (e.g., update software patches, virus scan, empty trash, run utility software, close all programs, reboot, use help sources).
9-12.CS.a.4	Explain and demonstrate how specialized computing devices can be used for problem solving, decision-making and creativity in all subject areas.
9-12.CS.a.5	Describe how computing devices manage and allocate shared resources (e.g., memory, Central Processing Unit [CPU]).
9-12.CS.a.6	Examine the historical rate of change in computing devices (e.g., power/energy, computation capacity, speed, size, ease of use) and discuss the implications for the future.
9-12.CS.b	Human and Computer Partnerships
9-12.CS.b.1	Identify a problem that cannot be solved by humans or machines alone and design a solution for it by decomposing the task into sub-problems suited for a human or machine to accomplish (e.g., a human-computer team playing chess, forecasting weather, piloting airplanes).
9-12.CS.c	Networks
9-12.CS.c.1	Explain how network topologies and protocols enable users, devices, and systems to communicate with each other.
9-12.CS.c.2	Examine common network vulnerabilities (e.g., cyberattacks, identity theft, privacy) and their associated responses.
9-12.CS.c.3	Examine the issues (e.g., latency, bandwidth, firewalls, server capability) that impact network functionality.
9-12.CS.d	Services
9-12.CS.d.1	Compare the value of using an existing service versus building the equivalent functionality (e.g., using a reference search engine versus creating a database of references for a project).
9-12.CS.d.2	Explain the concept of quality of service (e.g., security, availability, performance) for services providers (e.g., online storefronts that must supply secure transactions for buyer and seller).

Grades 9 – 12: Computational Thinking (CT)

9-12.CT.a	Abstraction
9-12.CT.a.1	Discuss and give an example of the value of generalizing and decomposing aspects of a problem in order to solve it more effectively.
9-12.CT.b	Algorithms
9-12.CT.b.1	Recognize that the design of an algorithm is distinct from its expression in a programming language.
9-12.CT.b.2	Represent algorithms using structured language, such as pseudocode.
9-12.CT.b.3	Explain how a recursive solution to a problem repeatedly applies the same solution to smaller instances of the problem.
9-12.CT.b.4	Describe that there are ways to characterize how well algorithms perform and that two algorithms can perform differently for the same task.
9-12.CT.b.5	Explain that there are some problems which cannot be computationally solved.
9-12.CT.c	Data
9-12.CT.c.1	Describe how data types, structures, and compression in programs affect data storage and quality (e.g., digital image file sizes are affected by resolution and color depth).
9-12.CT.c.2	Create an appropriate multidimensional data structure that can be filtered, sorted, and searched (e.g., array, list, record).
9-12.CT.c.3	Create, evaluate, and revise data visualization for communication and knowledge.
9-12.CT.c.4	Analyze a complex data set to answer a question or test a hypothesis (e.g., analyze a large set of weather or financial data to predict future patterns).
9-12.CT.c.5	Identify different problems (e.g., large or multipart problems, problems that need specific expertise, problems that affect many constituents) that can benefit from collaboration when processing and analyzing data to develop new insights and knowledge.
9-12.CT.d	Programming and Development
9-12.CT.d.1	Use a development process in creating a computational artifact that leads to a minimum viable product and includes reflection, analysis, and iteration (e.g., a data-set analysis program for a science and engineering fair, capstone project that includes a program, term research project based on program data).
9-12.CT.d.2	Decompose a problem by defining functions which accept parameters and produce return values.
9-12.CT.d.3	Select the appropriate data structure to represent information for a given problem (e.g., records, arrays, lists).
9-12.CT.d.4	Analyze trade-offs among multiple approaches to solve a given problem (e.g., space/time performance, maintainability, correctness, elegance).
9-12.CT.d.5	Use appropriate looping structures in programs (e.g., FOR, WHILE, RECURSION).
9-12.CT.d.6	Use appropriate conditional structures in programs (e.g., IF-THEN, IF-THEN-ELSE, SWITCH).
9-12.CT.d.7	Use a programming language or tool feature correctly to enforce operator precedence.
9-12.CT.d.8	Use global and local scope appropriately in program design (e.g., for variables).
9-12.CT.d.9	Select and employ an appropriate component or library to facilitate programming

	solutions (e.g., turtle, Global Positioning System [GPS], statistics library).
9-12.CT.d.10	Use an iterative design process, including learning from making mistakes, to gain a better understanding of the problem domain.
9-12.CT.d.11	Engage in systematic testing and debugging methods to ensure program correctness.
9-12.CT.d.12	Demonstrate how to document a program so that others can understand its design and implementation.
9-12.CT.e	Modeling and Simulation
9-12.CT.e.1	Create models and simulations to help formulate, test, and refine hypotheses.
9-12.CT.e.2	Form a model from a hypothesis generated from research and run a simulation to collect and analyze data to test that hypothesis.