Massachusetts Department of
ELEMENTARY & SECONDARY
EDUCATION

DRAFT

Massachusetts
Digital Literacy and Computer Science (DL&CS) Standards

Grades Kindergarten to 12

December 2015

Please direct all comments, suggested edits, and questions regarding these standards to:
mathsciencetech@doe.mass.edu

Table of Contents

Introduction to the Standards

Vision

Meaningful participation in modern society requires fluency in the uses and impact of technology for living, learning, and working. People need to be able to apply knowledge, skills, and dispositions central to digital literacy and computer science to their chosen interests and career fields. They must become computational thinkers so they can create, apply, or change existing computer hardware, software, and devices to meet the computing and information needs of their life and work.

The Digital Literacy and Computer Science (DL&CS) standards in this *Framework* articulate critical learning outcomes for Kindergarten through Grade 12 to help prepare students for the world that awaits them after high school. The standards represent the core elements of digital literacy and computer science and are intended to drive coherent, rigorous instruction that results in the mastery and application of digital literacy and computer science knowledge, reasoning, and skills.

Key Features

1. *The DL&CS Standards are organized into four strands: Computing and Society (CAS), Digital Tools and Collaboration (DTC), Computing Systems (CS), and Computational Thinking (CT).*

2. *The DL&CS Standards coherently progress from Kindergarten to grade 12.*
   The standards emphasize a focused and coherent progression of knowledge and skills. As students progress through their K-12 education, they acquire increasingly sophisticated knowledge, skills and dispositions in digital literacy and computer science.

3. *The DL&CS Standards articulate* ***practices*** *necessary for success.*
   The ***practices*** cultivate the internalization of dispositions that skillful people in digital literacy and computer science apply in reasoning, creation, and problem solving. ***Practices*** speak to the types of performance students should be able to demonstrate in the standards.

4. *The DL&CS Standards complement other Massachusetts Curriculum Frameworks.*
   The Frameworks and DL&CS Standards overlap in meaningful and substantive ways and offer an opportunity for all students to better apply and learn digital literacy and computer science. Much of the knowledge, skills, and dispositions central to digital literacy and computer science, such as computational thinking, also apply to other subjects, including, but not limited to, science, technology and engineering and mathematics.

Considerations

- The standards reflect what a student should know and be able to do as a result of instruction within each grade span (K-2, 3-5, 6-8, and 9-12). Educators have the flexibility of arranging the standards in any order within a grade span to suit the needs of students and the DL&CS programs.

- The ***practices*** articulate the dispositions students acquire across standards. Coupling practice with content gives the context for performance.
- Curricula and assessment must be developed in a way that builds students' knowledge and ability toward mastery of the standards. Effective instruction will engage students in multiple ***practices*** simultaneously.
- The standards identify the most essential material for students to know and be able to do. They are not an exhaustive list of all that could be included in a student's digital literacy and computer science education, nor should they prevent students from going beyond the standards where appropriate.
- Many standards include examples or additional clarification of the performance expectations.
- Some standards have multiple functions or multiple components, the purpose of which is to convey the richness of expected outcomes.

Digital Literacy and Computer Science (DL&CS) Standards Model

The standards for Kindergarten to grade12 are organized by **grade span**: Kindergarten to grade 2, grade 3 to grade 5, grade 6 to grade 8, and grade 9 to grade 12. Within each grade span, standards are grouped in four **strands**: Computing and Society, Digital Tools and Collaboration, Computing Systems, and Computational Thinking. Each strand is further subdivided into **topics** comprised of related **standards**. Standards define performance expectations, as well as what students should know and be able to do. Standards from different strands or topics may sometimes be closely related. Standards in every grade span and strand demonstrate a range of cognitive complexity such as reflected in Bloom's Revised Taxonomy (Anderson, Krathwohl et al. 2001): remembering, understanding, applying, analyzing, evaluating, and creating.

Vision

Meaningful participation in modern society requires fluency in the uses and impact of technology for living, learning, and working. People need to be able to apply knowledge, skills, and dispositions central to digital literacy and computer science to their chosen interests and career fields. They must become computational thinkers so they can create, apply, or change existing computer hardware, software, and devices to meet the computing and information needs of their life and work.

Learning Progression

Grade Spans	**Strands**			
K-2	**CAS: Computing and Society** a. Safety and Security b. Ethics and Laws c. Interpersonal and Societal Impact	**DTC: Digital Tools and Collaboration** a. Digital Tools b. Collaboration and Communication c. Research	**CS: Computing Systems** a. Computing Devices b. Human and Computer Partnerships c. Networks d. Services	**CT: Computational Thinking** a. Abstraction b. Algorithms c. Data d. Programming and Development e. Modeling and Simulation
3-5				
6-8				
9-12				

Practices: Connecting, Creating, Abstracting, Analyzing, Communicating, Collaborating, and Research

Strands

Learning Progressions

The Kindergarten through grade 12 content standards in this framework are organized by **grade span**: Kindergarten to grade 2, grade 3 to grade 5, grade 6 to grade 8, and grade 9 to grade 12. Within each grade span, standards are grouped into four strands: Computing and Society, Digital Tools and Collaboration, Computing Systems, and Computational Thinking. Each strand is further subdivided into **topics** comprised of related **standards**. Standards define performance expectations and what students should know and be able to do. Standards from different strands or topics may sometimes be closely related.

1. **Computing and Society (CAS)**:
   Computing impacts all people and has global consequences on such things as communications, assistive technology, social networking, and the economy. Computing innovations are valued by society. Computing is a key component of many professions and the content of digital media influences all citizens and society. Global disparities in access to the Internet, media, and devices may lead to an imbalance in equity and power. Principles of privacy, ethics, security, and copyright law influence digital safety and security, as well as interpersonal and societal relations.
   a) **Safety and Security**: Responsible citizens in the modern world apply principles of personal privacy and network security to the use of computing systems, software, the Internet, media, and data.
   b) **Ethics and Laws**: Ethics include standards of conduct, fairness, and responsible use of the Internet, data, media, and computing devices. An understanding of principles and laws of software licenses, copyrights, and acceptable use policies are necessary to be responsible citizens in the modern world.
   c) **Interpersonal and Societal Impact**: The use of computing devices, assistive technologies and applying a computational perspective to solving problems changes the way people think, work, live, and play. Computational approaches lead to new understanding, discoveries, challenges, and questions. Most professions rely on technology and advances in computing foster innovations in many fields. Differential access to principles of computing, computing devices, digital tools, and media in the global society, has potentially significant effects.

2. **Digital Tools and Collaboration (DTC)**:
   Digital tools are applications that produce, manipulate, or store data in a digital format, e.g., word processors, drawing programs, image/video/music editors, simulators, Computer-Aided Design (CAD) applications, publishing programs, etc. Digital tools are critical for conducting research, communicating, collaborating and creating in social, work, and personal environments. The use of digital tools is integral to success in school and career.
   a) **Digital Tools**: Digital tools are used to create, manipulate, analyze, edit, publish, or develop artifacts. Individuals and groups identify, evaluate, select, and adapt new tools as they emerge.
   b) **Collaboration and Communication**: A variety of digital tools are used to work collaboratively anytime and anywhere, inside and outside the classroom, both synchronously and asynchronously, to develop artifacts or solve problems, contribute to the learning of others, and communicate.

c) **Research:** A variety of digital tools are used to conduct research, answer questions, and develop artifacts to facilitate learning and convey understanding. Access to the Internet and digital tools allows people to gather, evaluate (for validity, bias, relevance, accuracy, etc.), organize, analyze, and synthesize information, data and other media from a variety of sources. Effective use of information, data, and media requires consideration of validity, ethics and attribution of sources.

3. Computing Systems (CS):

Computing systems are comprised of components, such as devices, software, interfaces, and networks that connect communities, devices, people, and services. They empower people to create, collaborate and learn via human-computer partnerships. The design of many computing systems empowers people to debug, extend, and create new systems. Computing systems require troubleshooting and maintenance to consistently function.

a) **Computing Devices**: Computing devices take many forms (e.g., car, insulin pump, or robot), not just personal computers, phones and tablets. They use many types of input data (collected via gesture, voice, movement, location, and other data) and run instructions in the form of programs to produce certain outputs (e.g., images, sounds, and actions). Computing will continue to be increasingly embedded into devices that are used in social, recreational, personal, and workplace environments.

b) **Human and Computer Partnerships**: Some tasks, such as repetitive tasks, or those involving complex computations, are best done by computers, while other tasks that don't have defined rules or are dynamic in nature, are best done by humans, however, many tasks are done through human-computer partnerships. Human-computer partnerships are characterized by the interaction of humans with devices and systems that work together to achieve a purpose or solution that would not be independently possible.

c) **Networks**: Network components, including hardware and software, carry out specific functions to connect computing devices, people and services. The Internet facilitates global communication and relies on considerations of network functionality and security.

d) **Services**: Data storage and computing occurs in many interconnected devices creating computational "services" that are the building blocks of computing systems. These services make use of data, algorithms, hardware, and connectivity that may occur on remote systems.

4. Computational Thinking (CT):

Computational thinking is a problem solving process that requires people to think in new ways to enable effective use of computing to solve problems and create solutions. The capacity of computers to rapidly and precisely execute programs makes new ways of designing, creating, and problem solving possible. Computational thinking is characterized by:

- analyzing, modeling, and abstracting ideas and problems so people and computers can work with them;
- designing solutions and algorithms to manipulate these abstract representations (including data structures); and
- Identifying and executing solutions (e.g., via programming).

a) **Abstraction**: Abstraction is a 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. This process creates a new representation which successfully reframes the problem. At the most basic level of abstraction, data structures are used to represent information so that algorithms can operate on the data to create a result.

b) **Algorithms**: An algorithm is a sequence of precisely defined steps to solve a particular problem. Carefully designed algorithms are essential to solving complex problems using computers. Effective algorithms are efficient, clear, reusable, and accurate.

c) **Data**: Collecting, managing, and interpreting a vast amount of raw data is part of the foundation of our information society and economy. The storage of data impacts how data is used and accessed. Computational tools enable insights and decisions through new techniques for data collection and analysis.

d) **Modeling and Simulation**: Computational modeling and simulation help people to represent and understand complex processes and phenomena. Computational models and simulations are used, modified, and created to analyze, identify patterns, and answer questions of real phenomena and hypothetical scenarios.

e) **Programming and Development**: Programming articulates and communicates instructions in such a way that a computer can execute a task. Programming makes use of abstractions, algorithms, and data to implement ideas and solutions as executable code through an iterative process of design and debugging. The process of creating software includes understanding the development life cycle, such as testing, usability, documentation, and release. Software development is the application of engineering principles (usually by a team) to produce useful, reliable software at scale and to integrate software into other engineered artifacts.

Practices

Practices cultivate the internalization of dispositions that skillful people in digital literacy and computer science apply to solve problems. The ***Practices*** speak to the types of performance students should be able to demonstrate in the standards. As students progress through their education, they should acquire increasingly sophisticated practices. Effective instruction couples practices with digital literacy and computer science content to provide a context for performance.

1. **Creating:**
   Digital literacy and computer science are disciplines in which students demonstrate creative thinking, construct knowledge, and develop innovative artifacts and processes using technology. Students engage in the creative aspects of computing by designing and developing interesting computational artifacts and by applying techniques to creatively solve problems. Skills include:
   - Creating artifacts, or computational projects with practical, personal, and /or social intent;
   - Selecting appropriate methods, paths or techniques to develop artifacts;
   - Using appropriate algorithmic and information-management principles and/or digital tools;
   - Applying critical thinking, digital tools and technology to solve problems;
   - Making ethical and responsible choices in selecting tools, information, and media to create and share artifacts; and
   - Reviewing, revising, and iterating on work to create high-quality artifacts.

2. **Connecting:**
   Developments in computing have far-reaching effects on society and have led to significant innovations. The developments have implications for individuals, society, commercial markets, and innovation. Students study their effects and draw connections between different computing concepts. Skills include:
   - Describing the impact of computing on society (humanity), economies, laws, and histories; and
   - Distinguishing between ethical and unethical practices with respect to safe and responsible use of information, data, media, and computing devices.

3. **Abstracting**:
   Computational thinking requires understanding and applying abstraction at multiple levels. Students use abstraction to develop models and to classify and manage information. Skills include:
   - Identifying abstractions;
   - Describing modeling in a computational context;
   - Using abstraction and decomposition when attacking complex tasks or designing complex systems;
   - Classifying data into groups and hierarchies; and
   - Identifying attributes (properties) of the data groups.

4. **Analyzing**:
   Students use critical thinking and analytical skills to locate, evaluate and analyze information, information sources, their own computational artifacts, and the computational artifacts others have produced. Skills include:
   - Asking questions to define a problem or information need;
   - Describing and articulating a problem or information need;
   - Evaluating information sources, research, data, proposed solutions, models, or prototypes;

- Identifying ways to improve solutions or information quality; and
- Selecting and justifying appropriateness, precision, or quality of “best” solution and information sources.

5. **Communicating**:
   Communication is the expression and exchange of information between two or more people. Communication includes written and oral mediums, as well as tangible representations supported by graphs, visualizations, demonstrations, stories, and analysis. Effective communication is accurate, clear, concise, persuasive, and responsible. Skills include:
   - Evaluating various digital tools for best expression of a particular idea or set of information;
   - Selecting and using digital media and tools to communicate effectively;
   - Communicating to or with different audiences;
   - Describing computation with accurate and precise language, notations, or visualizations where relevant;
   - Summarizing the purpose of a proposed solution, model, prototype, or computational artifact;
   - Justifying the design, appropriateness of choices and selection of solution; and
   - Communicating responsibly, such as respecting intellectual property.

6. **Collaborating**:
   People working collaboratively in teams, locally or globally, can often achieve more than individuals working alone. Effective collaboration draws on diverse perspectives, skills, knowledge, and dispositions to address complex and open-ended problems to accomplish goals. Skills include:
   - Collaborating with others to conduct research, solve a computational problem, or developing digital artifacts;
   - Collaborating with others to create computational artifacts, computational projects, or digital by-products;
   - Exchanging knowledge and feedback with a partner or team member.

7. **Researching**:
   Students apply digital tools to gather, evaluate, and use information in a legal, safe, and ethical manner. Skills include:
   - Defining a problem, research question, or goal;
   - Identifying information needs, whether primary (e.g., raw data/experimentation/collection), or secondary (e.g., existing information);
   - Employing research strategies to locate all possible sources;
   - Evaluating and selecting the best sources of information for credibility, accuracy, and relevance, which may include original data, creating a prototype, or conducting other tangible work;
   - Using information ethically: attributing sources of information (text, written, images, other media) using the appropriate citation format for the discipline;
   - Organizing and analyzing information;
   - Synthesizing and inferring information and data; and
   - Creating a thesis that addresses the research question.

Standards

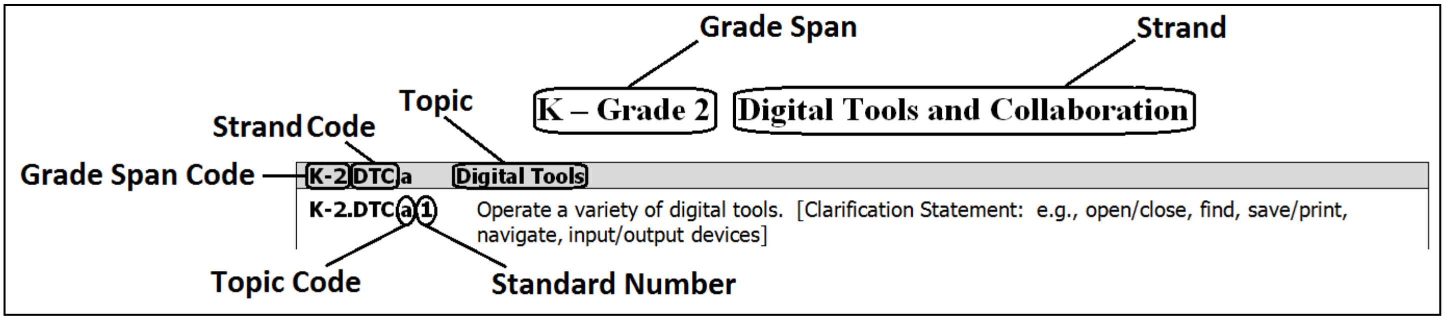
The coding system used for the standards identifies the strand and topic of the standard.

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 and does not imply teaching order.



K-2.DTC a 1 Operate a variety of digital tools. [Clarification Statement: e.g., open/close, find, save/print, navigate, input/output devices]

Kindergarten to Grade 2

Students in this grade span develop concepts through exploration, discovery, and creativity with the guidance, support, and encouragement of their educator. They design, build, and test inventions and solutions through exploration and play. They learn that tools help people do things better or more easily or do some things that could otherwise not be done at all.

Kindergarten – Grade 2: Computing and Society (CAS)

K-2.CAS.a	**Safety and Security**
K-2.CAS.a.1	Demonstrate proper ergonomics (e.g., body position) when using devices.
K-2.CAS.a.2	Use electrical devices safely (e.g., unplug devices by pulling the plug, not the cord; water/food and electric devices do not mix).
K-2.CAS.a.3	Care for devices appropriately (e.g., handling devices gently; completely shutting down devices when not in use; storing devices in the appropriate container).
K-2.CAS.a.4	Explain that a password helps protect the privacy of information.
K-2.CAS.a.5	Identify safe and unsafe examples of online communications.
K-2.CAS.a.6	Identify the importance of keeping personal information (e.g., name, location, phone number, address, school, etc.) private.
K-2.CAS.a.7	Identify which personal information (e.g., user name or real name, school name or home address) can and cannot be shared online and with whom.
K-2.CAS.a.8	Explain why it is necessary to report inappropriate electronic content or contact.
K-2.CAS.b	**Ethics and Laws**
K-2.CAS.b.1	Define good digital citizenship as using technology safely, responsibly, and ethically.
K-2.CAS.b.2	Demonstrate responsible use of computers, peripheral devices, and resources as outlined in school rules (Acceptable Use Policy (AUP) K-2).
K-2.CAS.b.3	Explain who owns a digital artifact
K-2.CAS.b.4	Explain the importance of giving credit to media creators/owners when using their work.
K-2.CAS.c	**Interpersonal and Societal Impact**
K-2.CAS.c.1	Identify and describe how people use many types of technologies in their daily work and personal lives.
K-2.CAS.c.2	Explain that the purpose of advertisements is to sell things to people, while content provides information to people.

Kindergarten – Grade 2: Digital Tools and Collaboration (DTC)

K-2.DTC.a	**Digital Tools**
K-2.DTC.a.1	Operate a variety of digital tools (e.g., open/close, find, save/print, navigate, use input/output devices).
K-2.DTC.a.2	Identify, locate, and use letters, numbers, and special keys on a keyboard (e.g., space bar, Shift, Delete).
K-2.DTC.a.3	Define a digital artifact as digital content.
K-2.DTC.a.4	Use appropriate digital tools to create, review, and revise artifacts that include text, images and audio, individually and collaboratively.
K-2.DTC.b	**Collaboration and Communication**
K-2.DTC.b.1	Collaboratively use digital tools and media-rich resources to communicate key ideas and details in a way that informs, persuades, and/or entertains.
K-2.CAS.b.2	Use a variety of digital tools to exchange information and feedback with teachers and other students (e.g., e-mail, text messaging).
K-2.CAS.b.3	Use a variety digital tools to present information to others (e.g., use a Liquid Crystal Display (LCD) projector or screen-sharing application to give a presentation or display images, audio, or video)
K-2.DTC.c	**Research**
K-2.DTC.c.1	Conduct basic keyword searches to gather information from teacher-provided digital sources (e.g., online library catalog, databases).
K-2.DTC.c.2	Create an artifact individually and collaboratively that answers a research question while clearly expressing thoughts and ideas.
K-2.DTC.c.3	Cite sources using a developmentally appropriate format (e.g., title of book, author of book, name of database, Uniform Resource Locator (URL) of a website).

Kindergarten – Grade 2: Computing Systems (CS)

K-2.CS.a	**Computing Devices**
K-2.CS.a.1	Identify different kinds of computing devices in the classroom and other places (e.g., laptops, tablets, smart phones, desktops, printers).
K-2.CS.a.2	Explain that a computing device runs when a program or command is executed.
K-2.CS.a.3	Operate a variety of computing systems (e.g., turn on, use input/output devices such as a mouse, keyboard, or touch screen; find, navigate, launch a program).
K-2.CS.b	**Human and Computer Partnerships**
K-2.CS.b.1	Explain that computers are different from living things because computers rely on electricity to operate and do not grow, reproduce, or need food, air, or water to operate.
K-2.CS.b.2	Discriminate between tasks that are best completed by humans or by computers (e.g., a human can solve simple mathematical problems involving a few numbers; machines can solve very complicated mathematical problems involving hundreds, thousands, or millions of numbers).
K-2.CS.b.3	Recognize that different tools can solve the same problem (e.g., pen and paper, calculators, and smart phones can all be used to solve simple mathematical problems)
K-2.CS.c	**Networks**
K-2.CS.c.1	Explain that the Internet links computers and devices locally and around the world allowing people to access and communicate information.
K-2.CS.d	**Services – There are no standards in this strand for this grade span.**

Kindergarten – Grade 2: Computational Thinking (CT)

K-2.CT.a	**Abstraction**
K-2.CT.a.1	List the attributes of a common object, for example, cars have a color, type (pickup, van, or sedan), engine size, etc.
K-2.CT.b	**Algorithms**
K-2.CT.b.1	Define an algorithm as a sequence of defined steps.
K-2.CT.b.2	Create simple algorithm, individually and collaboratively, without using computers to complete a task (e.g., making a sandwich, getting ready for school, checking a book out of the library).
K-2.CT.b.3	Enact an algorithm using tangible materials (e.g., manipulatives, your body) or present the algorithm in a visual medium (e.g., storyboard).
K-2.CT.c	**Data**
K-2.CT.c.1	Explain that computers can save information as data that can be stored, searched, retrieved, and deleted.
K-2.CT.c.2	Identify different kinds of data (e.g., text, charts, graphs, numbers, pictures, audio, video, collections of objects.)
K-2.CT.c.3	Identify, research, and collect a data set on a topic, issue, problem, or question using age-appropriate digital technologies.
K-2.CT.c.4	Propose a developmentally appropriate solution to a problem or question based on an analysis of the data and critical thinking, individually and collaboratively.
K-2.CT.c.5	Create data visualizations (e.g., charts and infographics), individually and collaboratively.
K-2.CT.d	**Programming and Development**
K-2.CS.d.1	Define a computer program as a set of commands created by people to do something.
K-2.CS.d.2	Explain that computers only follow the program's instructions.
K-2.CS.d.3	Construct a simple program using tools that that do not require a textual programming language (e.g., a block-based programming language).
K-2.CT.e	**Modeling and Simulation**
K-2.CT.e.1	Define simulation and identify the concepts illustrated by a simple simulation (e.g., growth and health, butterfly life cycle).
K-2.CT.e.2	Describe how models represent a real-life system (e.g., globe, map).

Grades 3 to 5

Grades 3 – 5: Computing and Society (CAS)

3-5.CAS.a	**Safety and Security**
3-5.CAS.a.1	Describe how to use proper ergonomics (e.g., body position, lighting, positioning of equipment, taking breaks) when using devices.
3-5.CAS.a.2	Describe the threats to safe and efficient use of devices (e.g., SPAM, spyware, phishing, viruses) associated with various forms of technology use (e.g., downloading and executing software programs, following hyperlinks, opening files).
3-5.CAS.a.3	Identify appropriate and inappropriate uses of technology when posting to social media, sending e-mail, and browsing the Internet
3-5.CAS.a.4	Explain the proper use and operation of security technologies (e.g., passwords, virus protection software, spam filters, popup blockers, cookies).
3-5.CAS.a.5	Describe the potential risks/dangers, and employ safe practices associated with various forms of online communications, downloads, linking, Internet purchases, advertisements, and inappropriate content within constrained environments.
3-5.CAS.a.6	Identify different types of cyberbullying (e.g., harassment, flaming, excluding people, outing, and impersonation)
3-5.CAS.a.7	Explain that if you encounter cyberbullying, you should immediately tell a responsible adult (e.g., teacher or parent).
3-5.CAS.b	**Ethics and Laws**
3-5.CAS.b.1	Demonstrate responsible use of computers, peripheral devices, and resources as outlined in school rules (Acceptable Use Policy 3-5).
3-5.CAS.b.2	Describe the difference between digital artifacts that are open or free and those that are protected by copyright.
3-5.CAS.b.3	Explain the guidelines for the fair use of downloading, sharing, or modifying of digital artifacts.
3-5.CAS.b.4	Describe the purpose of copyright and the possible consequences for inappropriate use of digital artifacts that are protected by copyright.
3-5.CAS.b.5	Explain the laws (e.g., section 508, Telecommunication Act of 1996) that help ensure that people with disabilities can access electronic and information technology.
3-5.CAS.c	**Interpersonal and Societal Impact**
3-5.CAS.c.1	Explain the different forms of web advertising (e.g., search ads, pay-per-click ads, banner ads, targeted ads, in-game ads, and e-mail ads)
3-5.CAS.c.2	Explain why websites, digital resources, and artifacts may include advertisements and collect personal information.
3-5.CAS.c.3	Define the digital divide as unequal access to technology on the basis of differences, such as income, education, age, and geographic location.
3-5.CAS.c.4	Use critical thinking to explain how access to technology helps empower individuals and groups (e.g., gives them access to information, the ability to communicate with others around the world, and allows them to buy and sell things).
3-5.CAS.c.5	Identify resources in the community that can give people access to technology (e.g., libraries, community centers, education programs, schools, hardware/software donation programs, etc.)

3-5.CAS.c.6	Identify ways in which people with disabilities access and use technology (e.g., audio players and recorders, FM listening systems, magnifiers, etc.)
3-5.CAS.c.7	Identify the impact of social media and cyberbullying on individuals, families, and society.

Grades 3 – 5: Digital Tools and Collaboration (DTC)

3-5.DTC.a	**Digital Tools**
3-5.DTC.a.1	Type five words-per-minute times grade level (e.g., Grade 5 = 25 words/minute).
3-5.DTC.a.2	Navigate between local, networked, or online/cloud environments and transfer files between each (upload/download).
3-5.DTC.a.3	Use digital tools (local and online) to manipulate and publish multimedia artifacts.
3-5.DTC.b	**Collaboration and Communication**
3-5.DTC.b.1	Communicate key ideas and details in a way that informs, persuades, and/or entertains working individually or collaboratively using digital tools and media-rich resources.
3-5.DTC.b.2	Collaborate through online digital tools under teacher supervision.
3-5.DTC.c	**Research**
3-5.DTC.c.1	Identify digital information sources to answer research questions (e.g., online library catalog, online encyclopedias, databases, websites, etc.).
3-5.DTC.c.2	Perform searches to locate information using two or more key words and techniques to refine and limit such searches.
3-5.DTC.c.3	Evaluate digital sources for accuracy, relevancy, and appropriateness.
3-5.DTC.c.4	Gather and organize information from digital sources by quoting, paraphrasing, and/or summarizing.
3-5.DTC.c.5	Create an artifact that answers a research question clearly communicating thoughts and ideas.
3-5.DTC.c.6	Cite text-based sources using a school- or district-adopted format (e.g., Modern Language Association (MLA)).
3-5.DTC.c.7	Provide basic source information (e.g., Uniform Resource Locator (URL), date accessed) for non-text-based sources (e.g., images, audio, video, etc.).

Grades 3 – 5: Computing Systems (CS)

3-5.CS.a	**Computing Devices**
3-5.CS.a.1	Identify a broad range of computing devices (e.g., computers, phones, tablets, robots, e-textiles, etc.) and appropriate uses for them.
3-5.CS.a.2	Describe the function and purpose of various input and output devices (e.g., monitor, screen, keyboard, controller, speakers).
3-5.CS.a.3	Demonstrate an appropriate level of proficiency (connect and record data, print, send command, connect to internet, search) in using a range of computing devices (e.g., probes, sensors, printers, robots, computers, cell phones).
3-5.CS.a.4	Identify and solve simple hardware and software problems that may occur during everyday use (e.g., power, connections, application window or toolbar).
3-5.CS.a.5	Describe the differences between hardware and software.
3-5.CS.b	**Human and Computer Partnerships**
3-5.CS.b.1	Compare and contrast human and computer performance on similar tasks (e.g., sorting alphabetically or finding a path across a cluttered room) to understand which is best suited to the task.
3-5.CS.b.2	Explain how hardware and applications (e.g., Global Positioning System (GPS) navigation for driving directions, text-to-speech translation, language translation) can enable everyone, including people with disabilities, do things they could not do otherwise.
3-5.CS.b.3	Explain advantages and limitations of technology (e.g., a spell-checker can check thousands of words faster than a human could look them up, however, a spell-checker might not know whether ‘underserved’ is correct or if the author’s intent was to type ‘undeserved.’).
3-5.CS.c	**Networks**
3-5.CS.c.1	Describe how a network is made up of a variety of components and identify the common components.
3-5.CS.c.2	Describe the need for authentication (e.g., identification of users and devices as it relates to access permissions, privacy, and security).
3-5.CS.c.3	Define and explain what the Internet is by using common terms (e.g., website, Uniform Resource Locator (URL), Internet Protocol (IP) address).
3-5.CS.c.4	Recognize that there are many sources of and means for accessing information within a network.
3-5.CS.d	**Services**
3-5.CS.d.1	Identify common services (e.g., search engines or document storage) and describe how they differ from locally installed applications.

Grades 3 – 5: Computational Thinking (CT)

3-5.CT.a	**Abstraction**
3-5.CT.a.1	Use numbers or letters to represent information in another form (e.g., secret codes, Roman numerals, abbreviations).
3-5.CT.a.2	Explain how a variable (a storage location associated with a name) can be used to store changing data/values (e.g., the score in video game, account balance, numbers to add).
3-5.CT.a.3	Organize information in different ways to make it more useful/relevant (e.g., sorting, tables).
3-5.CT.a.4	Make a list of sub-problems to consider while addressing a larger problem.
3-5.CT.b	**Algorithms**
3-5.CT.b.1	Define an algorithm as a sequence of instructions that can be processed by a computer.
3-5.CT.b.2	Recognize that different solutions exist for the same problem.
3-5.CT.b.3	Use logical reasoning to predict outcomes of an algorithm.
3-5.CT.b.4	Write an algorithm to solve a grade-level appropriate problem (e.g., move a character/robot/person through a maze, instruct a character/robot/person to draw a specific shape, have character/robot/person start, repeat, or end activity as required or upon a specific event), individually and collaboratively.
3-5.CT.b.5	Detect and correct logical errors in algorithms; written, mapped, live action, or digital.
3-5.CT.c	**Data**
3-5.CT.c.1	Use a Base-2 System to demonstrate how a string of bits can be used to represent alphanumeric information (e.g., letters, numbers, and symbols).
3-5.CT.c.2	Describe examples of databases from everyday life (e.g., library catalogs, school records, telephone directories, and contact lists).
3-5.CT.c.3	Collect and manipulate data using a variety of computing methods (e.g., sorting, totaling, and averaging).
3-5.CT.c.4	Utilize a database, such as a spreadsheet, to collect, organize, graph, and analyze data to answer a question.
3-5.CT.d	**Programming and Development**
3-5.CT.d.1	Create, test, and modify a program in a graphical environment (e.g., block-based visual programming language), individually and collaboratively.
3-5.CT.d.2	Use arithmetic operators, conditionals, and repetition in programs.
3-5.CT.d.3	Use interactive debugging to detect and correct simple program errors.
3-5.CT.d.4	Recognize that programs need known initial conditions (e.g., set initial score to zero in a game, initialize variables, or initial values set by hardware input).
3-5.CT.e	**Modeling and Simulation**
3-5.CT.e.1	Identify the concepts illustrated by a simulation (e.g., ecosystem, predator/prey, invasive species).
3-5.CT.e.2	Create a simple model of a system (e.g., cell, solar system) and explain what the model shows and does not show, individually and collaboratively.
3-5.CT.e.3	Use data from a simulation to answer a question, individually and collaboratively.

Grades 6 to 8

Grades 6 – 8: Computing and Society (CAS)

6-8.CAS.a	**Safety and Security**
6-8.CAS.a.1	Identify threats and actively protect devices from viruses, intrusion, vandalism, and other malicious activities.
6-8.CAS.a.2	Describe how cyberbullying can be prevented and managed.
6-8.CAS.a.3	Explain the connection between the persistence of data on the Internet, online identity, and privacy.
6-8.CAS.a.4	Describe and use safe and appropriate practices when participating in online communities (e.g., discussion groups, blogs, social networking sites).
6-8.CAS.a.5	Discriminate between appropriate and inappropriate content on the Internet.
6-8.CAS.b	**Ethics and Laws**
6-8.CAS.b.1	Explain how copyright law and licensing protect the owner of intellectual property.
6-8.CAS.b.2	Explain possible consequences of violating intellectual property law.
6-8.CAS.b.3	Apply fair use for using copyrighted materials (e.g., images, music, video, text).
6-8.CAS.b.4	Identify the legal consequences of cyberbullying/harassment in social computing.
6-8.CAS.b.5	Differentiate among open source and proprietary software licenses and their applicability to different types of software and media.
6-8.CAS.b.6	Demonstrate compliance with the school’s Acceptable Use Policy.
6-8.CAS.b.7	Identify software license agreements and application permissions.
6-8.CAS.b.8	Explain positive and malicious purposes of hacking.
6-8.CAS.b.9	License original content and extend license for sharing in the public domain (e.g., creative commons).
6-8.CAS.c	**Interpersonal and Societal Impact**
6-8.CAS.c.1	Describe current events and emerging technologies in computing and the effects they may have on education, the workplace, individuals, communities, and global society.
6-8.CAS.c.2	Identify and discuss the technology proficiencies needed in the workplace, as well as ways to prepare to meet these demands.
6-8.CAS.c.3	Relate the distribution of computing resources in a global society to issues of equity, access, and power.
6-8.CAS.c.4	Evaluate how media and technology can be used to distort, exaggerate, and misrepresent information.
6-8.CAS.c.5	Evaluate the bias of digital information sources, including websites.

Grades 6 – 8: Digital Tools and Collaboration (DTC)

6-8.DTC.a	**Digital Tools**
6-8.DTC.a.1	Identify and explain the strengths, weaknesses, and capabilities of a variety of digital tools.
6-8.DTC.a.2	Identify the kinds of content associated with different file types and why different file types exist (e.g., formats for word processing, images, music, three-dimensional (3-D) drawings, etc.).
6-8.DTC.a.3	Integrate information from multiple file formats into a single artifact.
6-8.DTC.a.4	Use advanced tools to design and create online content (e.g., webpage, blog, digital portfolio, multimedia), individually and collaboratively.
6-8.DTC.a.5	Develop and conduct an online survey, individually and collaboratively.
6-8.DTC.b	**Collaboration and Communication**
6-8.DTC.b.1	Communicate and publish key ideas and details in a way that informs, persuades, and/or entertains working individually or collaboratively using a variety of digital tools and media-rich resources.
6-8.DTC.b.2	Collaborate synchronously and asynchronously through online digital tools (e.g., not necessarily in the same time and place).
6-8.DTC.b.3	Demonstrate ability to communicate appropriately through various online tools (e.g., e-mail, social media, texting, or blog comments).
6-8.DTC.c	**Research**
6-8.DTC.c.1	Perform advanced searches to locate information using a variety of digital sources (e.g., Boolean Operators, limiters like reading level, subject, media type).
6-8.DTC.c.2	Evaluate quality of digital sources for reliability e.g., currency, relevancy, authority, accuracy, and purpose of digital information.
6-8.DTC.c.3	Gather, organize, and analyze information from digital sources by quoting, paraphrasing, and/or summarizing.
6-8.DTC.c.4	Create an artifact, individually and collaboratively, that answers a research question and communicates results and conclusions.
6-8.DTC.c.5	Use digital citation tools to cite sources using a school- or district-adopted format (e.g., Modern Language Association (MLA)), including proper citation for all text and non-text sources (e.g., images, audio, and video).

Grades 6 – 8: Computing Systems (CS)

6-8.CS.a	**Computing Devices**
6-8.CS.a.1	Describe the main functions of an operating system.
6-8.CS.a.2	Recognize that there is a wide range of application software.
6-8.CS.a.3	Identify and describe the function of the main internal parts of a basic computing device (e.g., motherboard, hard drive, Central Processing Unit (CPU)).
6-8.CS.a.4	Identify and describe the use of sensors, actuators, and control systems in an embodied system (e.g., a robot, an e-textile, installation art, smart room).
6-8.CS.a.5	Design and demonstrate the use of a device (e.g., robot, e-textile) to accomplish a task, individually and collaboratively.
6-8.CS.a.6	Use a variety of computing devices (e.g., probes, sensors, handheld devices, Global Positioning System (GPS), etc.) to collect, analyze, and present information for content-related problems, individually and collaboratively.
6-8.CS.a.7	Identify steps involved in diagnosing and solving routine hardware and software problems (e.g., power, connections, application window or toolbar, cables, ports, network resources, video, and sound) that occur during everyday computer use.
6-8.CS.b	**Human and Computer Partnerships**
6-8.CS.b.1	Explain why some problems can be solved more easily by computers or humans based on a general understanding of types of tasks at which each student excels.
6-8.CS.b.2	Describe how humans and machines interact to solve problems that cannot be solved by either alone (e.g., scientific experiments that involve drawing conclusions by analyzing vast amounts of data).
6-8.CS.c	**Networks**
6-8.CS.c.1	Explain the difference between physical (wired), local area wireless, and mobile networks.
6-8.CS.c.2	Model the components of a network.
6-8.CS.c.3	Describe how information, both text and non-text, is translated and communicated between digital computers over a computer network.
6-8.CS.d	**Services**
6-8.CS.d.1	Identify capabilities of devices that are enabled through services (e.g., a wearable device that stores fitness data in the cloud or a mobile device that uses location services for navigation).

Grades 6 – 8: Computational Thinking (CT)

6-8.CT.a	**Abstraction**
6-8.CT.a.1	Demonstrate how data is abstracted (e.g., mouse's position and x and y coordinates, image and screen pixels, Internet Protocol (IP) addresses and domain names) by listing the attributes of common objects (e.g., address, position, color, shape).
6-8.CT.a.2	Define a simple function that hides the complexity of a task/problem and can be reused to solve similar tasks/problems.
6-8.CT.a.3	Define and apply a hierarchical classification scheme to a complex system, such as the human body, or in computing (e.g., objects in high level object oriented languages, animal classification).
6-8.CT.b	**Algorithms**
6-8.CT.b.1	Design solutions that use repetition and two-way selection (e.g., IF/ELSE, FOR, WHILE).
6-8.CT.b.2	Use logical reasoning to predict outputs while showing an understanding of inputs.
6-8.CT.b.3	Decompose a problem and create a sub-solution for each of its parts (e.g., video game, robot obstacle course, making dinner), individually and collaboratively.
6-8.CT.b.4	Recognize that more than one algorithm can solve a given problem.
6-8.CT.b.5	Recognize that boundaries need to be taken into account for an algorithm to produce correct results.
6-8.CT.c	**Data**
6-8.CT.c.1	Demonstrate that numbers can be represented in different number systems (e.g., binary, octal, and hexadecimal) and text can be represented in different ways such as American Standard Code for Information Interchange (ASCII).
6-8.CT.c.2	Describe how computers store, manipulate, and transfer data types and files (e.g., integers, real numbers, and Boolean Operators) in a binary system.
6-8.CT.c.3	Create, modify, and use a database (e.g., define field formats, adding new records, manipulate data) to analyze data and propose solutions for a task/problem, individually and collaboratively.
6-8.CT.c.4	Perform a variety of operations such as sorting, filtering, and searching in a database to organize and display information in a variety of ways such as number formats (e.g., scientific notation, percentages, and exponents), charts, tables, and graphs.
6-8.CT.c.5	Select and use data-collection technology (e.g., probes, handheld devices, geographic mapping systems) to gather, view, organize, analyze, and report results for content-related problems, individually and collaboratively.
6-8.CT.d	**Programming and Development**
6-8.CT.d.1	Select the 'best' algorithm based on a given criteria (e.g., time, resource, accessibility) to solve a problem, individually and collaboratively.
6-8.CT.d.2	Use functions to hide the detail in a program (e.g., perform abstraction).
6-8.CT.d.3	Create a program that implements an algorithm to achieve a given goal, individually and collaboratively.
6-8.CT.d.4	Implement problem solutions using a programming language, including all of the following: looping behavior, conditional statements, expressions, variables, and functions.
6-8.CT.d.5	Engage in collaborative program development (e.g., pair programming).
6-8.CT.d.6	Perform program tracing to predict the behavior of programs.
6-8.CT.d.7	Use iterative development and debugging to explore the problem domain.
6-8.CT.e	**Modeling and Simulation**
6-8.CT.e.1	Create/modify and use a simulation to analyze and illustrate a concept in depth (e.g., use a

	simulation to illustrate genetic variation.), individually and collaboratively.
6-8.CT.e.2	Create a model of a real-world system and explain why some details were required in the model and why some could be ignored, individually and collaboratively.
6-8.CT.e.3	Modify existing simulations by introducing new parameter(s).
6-8.CT.e.4	Select and use computer simulations to gather, view, analyze, and report results for content-related problems (e.g., immigration, international trade, invasive species, dissection), individually and collaboratively.

Grades 9 to 12

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	Use safe and appropriate practices when creating digital artifacts.
9-12.CAS.a.3	Explain safe practices when collaborating online, including how to anticipate potentially dangerous situations.
9-12.CAS.a.4	Construct strategies to combat cyberbullying/harassment, individually and collaboratively.
9-12.CAS.a.5	Identify the mental health consequences of cyberbullying/harassment.
9-12.CAS.a.6	Explain how peer pressure in social computing settings influences choices.
9-12.CAS.a.7	Apply strategies for managing negative peer pressure and encouraging positive peer pressure; individually and collaboratively.
9-12.CAS.b	**Ethics and Laws**
9-12.CAS.b.1	Model mastery of the school's Acceptable Use Policy.
9-12.CAS.b.2	Identify computer-related laws and analyze their impact on digital privacy, security, intellectual property, network access, contracts, 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 meets accessibility requirements.
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, and intercultural communication); individually and collaboratively.
9-12.CAS.c.6	Cultivate a positive web presence (e.g., digital resume, portfolio, social media, etc.).
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 to use for a real-world task and justify the selection based on their efficiency and effectiveness, individually and collaboratively.
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 projects, 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; individually and collaboratively.
9-12.DTC.c.5	Create an artifact that answers a grade appropriate research question, communicating results and conclusions, and citing sources, individually and collaboratively.

Grades 9 - 12: Computing Systems (CS)

9-12.CS.a	**Computing Devices**
9-12.CS.a.1	Select computing devices to accomplish a real-world task and justify the selection based on cost, efficiency, and effectiveness; individually and collaboratively.
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., power, connections, application window or toolbar, cables, ports, network resources, video, and sound; update software patches; virus scan; empty trash; run Windows Utility®; closing all programs; rebooting, using 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, individually and collaboratively.
9-12.CS.a.5	Describe how computing devices manage and allocate shared resources (e.g., memory, Central Processing Unit (CPU), etc.).
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, individually and collaboratively.
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 (partition a task into sub-problems better suited for human or machine solutions, e.g., a human-computer team playing chess, forecasting weather, piloting airplanes), individually and collaboratively.
9-12.CS.c	**Networks**
9-12.CS.c.1	Explain how network 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, etc.) and their associated responses.
9-12.CS.c.3	Examine the issues (e.g., latency, bandwidth, firewalls, and 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.
9-12.CS.d.2	Explain the concept of quality of service for services as security, availability, performance, etc., which are configured based on required Service Level Agreements (SLAs) or Operational Level Agreements (OLAs).

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 abstraction to manage problem complexity.
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 and structures in programs affect data storage**.**
9-12.CT.c.2	Explain how digital image file sizes are affected by resolution and color depth.
9-12.CT.c.3	Demonstrate how data compression affects data representation and quality.
9-12.CT.c.4	Create an appropriate multidimensional data structure that can be filtered, sorted, and searched (data structure, e.g., array, list, and record), individually and collaboratively.
9-12.CT.c.5	Create, evaluate, and revise data visualization for communication and knowledge, individually and collaboratively.
9-12.CT.c.6	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), individually and collaboratively.
9-12.CT.c.7	Identify 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 that leads to 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, individually and collaboratively, that leads to a minimum viable product followed by reflection, analysis, and iteration (e.g., data-set analysis program for 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, individually and collaboratively.
9-12.CT.d.3	Select the appropriate data structure to represent information for a given problem (e.g., records, arrays, lists), individually and collaboratively.
9-12.CT.d.4	Analyze trade-offs among multiple approaches to solve a given problem (e.g., space/time performance, maintainability, correctness, elegance), individually and collaboratively.
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) component, statistics library, Scratch).

9-12.CT.d.10	Use the 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, individually and collaboratively.
9-12.CT.e	**Modeling and Simulation**
9-12.CT.e.1	Create models and simulations to help formulate, test, and refine hypotheses, individually and collaboratively.
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, individually and collaboratively.

Glossary

The following terms and definitions were developed using various resources listed in the resource section of this document.

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.
abstraction (product)	An abstraction is a new representation of a thing, a system or a problem which helpfully reframes a problem by hiding details irrelevant to the question at hand.
Acceptable Use Policy (AUP)	An acceptable use policy (AUP) is a document stipulating constraints and practices that a user must agree to for access to a district/corporate network or the Internet. Many businesses and educational facilities require that employees or students sign an acceptable use policy before being granted a network ID.
algorithm	A set of unambiguous rules or instructions to achieve a particular objective.[1]
alphanumeric	A combination of alphabetic and numeric characters, including other symbols, such as punctuation and mathematical symbols. [2] Example: ABC123
annotated bibliography	An annotated bibliography includes a summary and/or evaluation of each of the sources in a bibliography.
application	Software (program) that is used by people to accomplish a task.
array	A data structure comprising a collection of values of the same type accessible through an index. [1] Fixed size. Example: [A, B, C, D). is an array of letters. The second element of the array is B.
assistive technology	In general, the term `"assistive technology device" means any item, piece of equipment, or product system, whether acquired commercially off-the-shelf, modified, or customized, that is used to increase, maintain, or improve functional capabilities of a person with a disability. Exception: The term does not include a medical device that is surgically implanted, or the replacement of such device. [3]
asynchronous	Not necessarily in the same time and place.
attribute	A specification that defines a property of an object, element, or file. It may also refer to or set the specific value for a given instance of such. For clarity, attributes should more correctly be considered metadata. An attribute is frequently and generally a property of a property. (Wikipedia) Example: The "color" attribute of a red car would have the value "red."
authentication	Any process by which you verify someone or something (a device) is who or what or they or it claim(s) to be. Example: Some websites use a combination of e-mail address and password as a means of authentication.
bandwidth	Describes the maximum data transfer rate of a network or Internet connection. It measures how much data can be sent over a specific connection in a given amount of time. [4]
binary	A method of encoding data using two symbols, 1 and 0. [1]
binary number	A number written in the Base-2 Number System. [1] Example: the number 4 written in binary is 100.
bit	A basic unit of data that stores one binary value, 1 or 0. [1]
Boolean	A data type with only two values, TRUE or FALSE. [1]
browser cookie	A small piece of text recording activity about websites one visits stored on one's computer. [1]
class	In object-oriented programming, a class is an extensible program-code template for creating objects, providing initial values for state (member variables) and implementations of behavior (member functions, methods). (Wikipedia)
code	Any set of instructions expressed in a programming language. [1]
coding	The act of writing computer programs in a programming language. [1]

color depth	The number of different colors that may be used in an image dictated by the number of bits used to represent the color of each pixel. [1]
component	A reusable element with a specification for how it is to be used (with inputs and outputs, as appropriate).
computational artifact	Inventions, creations, final products, and development byproducts, created by the act or process of computing.
computational devices	See computing devices.
computational thinking	A way of thinking when computing that uses decomposition, pattern recognition, abstraction, pattern generalization, and algorithm design.
Computer Science (CS)	The study of computers and algorithmic processes, including their principles, hardware and software designs, applications, and their impact on society. [5]
computing	Any goal-oriented activity requiring, benefiting from, or creating algorithmic processes.[2]
computing artifacts	Any creation facilitated by a computer, such as digital documents, digital videos, databases, computer program including computational models and simulations.
computing devices	A machine that can be programmed to carry out a set of logical or arithmetic operations (e.g., a laptop computer, a mobile phone, a computer chip inside an appliance).
conditional/conditional statement	In computer science, conditional statements, conditional expressions, and conditional constructs are features of a programming language which perform different computations or actions depending on whether a programmer-specified *Boolean Condition* evaluates to true or false. [2]
Central Processing Unit (CPU)	The device within a computer that executes instructions. [1]
cyberbullying	The use of electronic communication to bully a person typically by sending messages of an intimidating or threatening nature.
cyber harassment	The use of the Internet or other electronic means to harass an individual, a group, or an organization.
database	An integrated and organized collection of logically related records or files or data that are stored in a computer system which consolidates records previously stored in separate files into a common pool of data records which provides data for many applications. [2]
data structure	A particular way to store and organize data within a computer program. [1]
data visualization	The presentation of data in a static or dynamic graphical format. Example: chart, table, or infographic.
De Morgan's Laws	A pair of logical rules that are used to help build conditions in computers programs. The rules are: NOT (A and B) = NOT A and NOT B NOT (A or B) = NOT A or NOT B
debugging	The process of finding and correcting errors in programs. [1]
decimal	The base 10 number system. [1]
decomposition	Breaking a problem or system down into its components. [1]
digital	Created in a form that is the 1s and 0s a computer uses to store information.
digital artifact	Digital content made by a human with intent and skill. [1] Example: computer animation, LED infused clothing, interactive sculpture, 3D printed objects or songs.
digital citizenship	The norms of appropriate, responsible behavior with regard to the use of technology.
digital creator	A person who makes digital artifacts. [1]
digital device	An electronic device that can receive, store, process, or send digital information.[6]
digital divide	The gap between those who have access to digital technology and those who do not, which is influenced by social, cultural and economic factors.
Digital Literacy	The ability to use digital technology, communication tools or networks to locate, evaluate, use, and create information. [7] The ability to understand and use information in multiple formats from a wide range of sources when it is presented via computers. [8] A person's ability to perform tasks effectively in a digital environment. Literacy includes the ability to read and interpret media, reproduce data and images through digital manipulation, and evaluate and apply new knowledge gained from digital environments. [9]

digital media	Media encoded in a computer-readable form. [1]
digital privacy	The protection of personal information on the Internet.
digital safety	The knowledge of maximizing the user's personal safety and security risks to private information and property associated with using the Internet and the self-protection from computer crime in general.
digital tools	An application that produces, manipulates, or stores data in a digital format (e.g., word processors, drawing programs, image/video/music editors, simulators, 3D-design sketchers, publishing programs).
ergonomics	Designing and arranging things people use so the people and things interact most efficiently and safely.
expression	An expression in a programming language is a combination of explicit values, constants, variables, operators, and functions that are interpreted according to the particular rules of precedence and of association for a particular programming language which computes and then produces (returns, in a stateful environment) another value.[2] Example: b = a + 2.
fair use	The legal concept that allows brief excerpts of copyrighted material to be used for purposes such as review, news reporting, teaching, scholarship, or art.
firewall	A network security system with rules to control incoming and outgoing traffic.
function	A function is 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. [10]
hacking	Appropriately applying ingenuity (from "The Meaning of Hack"), cleverly solving a programming problem (the New Hacker's Dictionary), and using a computer to gain unauthorized access to data within a system.
hardware	The physical components that make up a computer. [1]
hexadecimal	A positional numeral system with a radix or base of 16. It uses sixteen distinct symbols, most often the symbols 0–9 to represent values zero to nine, and A, B, C, D, E, F (or alternatively a, b, c, d, e, f) to represent values of ten to fifteen. Hexadecimal numerals are widely used by computer system designers and programmers. [2]
HTML	HyperText Mark-up Language; he language used to create web pages. [1]
infographics	A static data visualization used to condense large amounts of information that is more easily understood by the reader (e.g., maps, hierarchies, networks (Adaptation of definition by Edward Rolf Tufte)).
input (noun)	An input is a data value passed from the outside world to a computer. [1]
input (verb)	To input is to send data from the outside world into a computer system. [1]
Internet	A network of interconnected networks. [1]
intellectual property	Something (such as an idea, invention, or process) that comes from a person's mind. (Miriam Webster online)
IP address	Internet Protocol address is a unique numeric value that is assigned to a computer or other device connected to the Internet so that it may be identified and located. [1] Example: 127.0.0.1.
iterative	The act of repeating a process with the aim of approaching a desired goal, target, or result, such as a grammatical rule that can be repeatedly applied.. (Wikipedia)
latency	The amount of time it takes a packet of data to move across a network connection. [3]
LAN	Local Area Network is a computer network limited to a small area, such as an office building, university, or even a residential home. [3]
library / code library	A collection of programs, applications, or resources files. The goal of the Code Library is to provide students with sample applications and supplemental information to help them create or customize their own program or application.
Lifelong learning	All learning activity undertaken throughout life, with the aim of improving knowledge, skills and competences within a personal, civic, social and/or employment-related perspective. [11]
lists	A data structure for storing ordered values. [1] Data is of arbitrary/unfixed size.
loop	A loop is a sequence of statements which is specified once but which may be carried out several times in succession. (Wikipedia)

MAC Address	Media Access Control Address is a hardware identification number that uniquely identifies each device on a network. The MAC address is manufactured into every network card and, therefore, cannot be changed.
memory	Temporary storage used by computing devices.
minimum viable product	A prototype that embodies an initial set of design goals and facilitates live testing and revision.
model (noun)	A representation of (some part of) a problem or a system. [1]
modeling (verb)	The act of creating a model. [1]
network	More precisely, a computer network is a collection of computational devices (personal computers, phones, servers, switches, routers, and so on) connected to one another by cables or by wireless media and arranged so data may be sent between devices either directly or via other devices. [1]
open source software	Publishers of open source software provide copies of both the source code and the object code when they distribute computer programs to the public. In addition, they establish the terms of use of the software by means of a license.[6]
open source license	A contract that provides users with a sufficient set of privileges to access and modify the open source software's source code**.**[6]
operating system	A set of programs that manage the functioning of, and other programs' access to, hardware. [1]
operator	A character that represents a specific action (e.g., x is an arithmetic operator that represents multiplication). In computer programs, one of the most familiar sets of operators, the Boolean Operators, is used to work with true/false values. (Google)
output (noun)	A response from a system. [1] Example: a program that adds could have inputs of 2 and 2 with an output of 4.
output (verb)	To generate an output. [1]
parameter	A parameter is a special kind of variable used in a subroutine to refer to one of the pieces of data provided as input to the subroutine. 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. (Wikipedia)
peripheral device	A computer peripheral is any external device that provides input and output for the computer. For example, a keyboard and mouse are input peripherals, while a monitor and printer are output peripherals. Computer peripherals, or peripheral devices, are sometimes called "I/O devices," because they provide input and output for the computer. Some peripherals, such as external hard drives, provide both input and output for the computer.[3]
pixel	The smallest controllable element of a picture/display. [1]
Pop-up	Appearing suddenly on a computer screen. (Webster)
process (noun)	A process is a running program. [1]
process (verb)	The act of using data to perform a calculation or other operation. [1]
problem domain	The area of expertise or application that needs to be examined to solve a problem. Simply, looking at only the topics of an individual's interest and excluding everything else. https://en.wikipedia.org/wiki/Problem_domain
program	A set of instructions that the computer executes in order to achieve a particular objective.[1]
programming (computer programming)	The craft of analyzing problems and designing, writing, testing, and maintaining programs to solve them. [1]
programming language	Formal language used to give a computer instructions. [1]
proprietary software license	Proprietary software is licensed under legal right of the copyright holder, with the intent that the licensee is given the right to use the software *only* under certain conditions, and restricted from other uses, such as modification, sharing, studying, redistribution, or reverse engineering. https://en.wikipedia.org/wiki/Proprietary_software

proprietary software	Software distributed in object code form. The developers or distributors reserve all freedoms and rights.[6]
Pseudocode	An informal high-level description of the operating principle of a computer program or other algorithm.
recursive	A recursive function refers to a procedure or subroutine, implemented in a programming language, whose implementation references itself. (Wikipedia).
repetition	The process of repeating a task a set number of times or until a condition is met.[1]
resolution	A measurement of the number of pixels needed to display an image. [1]
router	A device that connects networks to one another.
safety	The awareness of personal, physical, and psychological well-being in a digital society.
selection	Using conditions to control the flow of a program. [1]
sequence (noun)	An ordered set of instructions. [1]
sequence (verb)	To arrange a set of instructions in a particular order. [1]
server	A computer or program dedicated to a particular set of tasks that provides services to other computers or programs on a network. [1]
services	Software and hardware that provide some capability that can be accessed by another program or device remotely or through a defined, discoverable interface.
simulation	Imitation of the operation of a real world process or system over time.
social computing	An umbrella term for communications and collaboration via the Internet.
software	The programs that run on the hardware/computer system. [1]
software piracy	Illegal copying, distribution, or use of software.
spam	Unsolicited commercial advertisements distributed online. Most spam comes to people via e-mail, but spam can also be found in online chat rooms and message boards. (http://compnetworking.about.com/library/glossary/bldef-spam.htm)
subroutine	A sequence of program instructions that perform a specific task packaged as a unit. This unit can then be used in programs wherever that particular task should be performed. In different programming languages, a subroutine may be called a procedure, function, routine, method, or subprogram. (Wikipedia)
switch	More precisely, a network switch. This is a device that connects multiple computers to one another on a single local area network (LAN) and directs packets from machine to machine. [1]
synchronous	Occurring at the same time.
table	A data type storing organized sets of data under column headings. [1]
trace	to follow it (program) to its origin or destination.
variable	A data store used in a program. [1]
visualization	The process of representing data graphically and interacting with these representations in order to gain insight into the data. Any technique for creating images, diagrams, or animations to communicate a message. (Wikipedia)
WAN	Wide area networks span long distances via telephone lines, fiber optic cables, or satellite links. They can also be composed of smaller LANs that are interconnected. [8]
web presence	All aspects of one's personal or corporate online identity (e.g., social media profiles, and personal and business websites).
World Wide Web	A service made of connected hypertext documents linked together across the Internet.

----

[1] Computing At School accessed May 2015: www.computingatschool.org.uk

[2] Wikipedia accessed June 2015

[3] IDEA.ed.gov accessed May 2015: http://idea.ed.gov

[4] TechTerms.com -The Tech Terms Computer Dictionary Accessed May 2015

[5] ACM/CSTA Model Curriculum for K–12 Computer Science, Revised 2011

[6] IT Law Wiki accessed June 2015: http://itlaw.wikia.com

[7] Digital Strategy Glossary of Key Terms http://www.digitalstrategy.govt.nz/Media-Centre/Glossary-of-Key-Terms/ accessed August 21, 2008

[8] Paul Gilster, *Digital Literacy*, New York: Wiley and Computer Publishing, 1997, p. 1.

[9] Barbara R. Jones-Kavalier and Suzanne L. Flannigan: *Connecting the Digital Dots: Literacy of the 21st-Century*

[10] *Webopedia, accessed June 2015: www.webopedia.com*

[11] European Commision, Making a European Area of Lifelong Learning a Reality, COM Brussels, Commision of European Communities, 2001, p. 9

Resources

Resources:

- CSTA K-12 Computer Science Standards, 2011: http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf
- Massachusetts Technology Literacy Standards – April 2008
- The American Association of School Librarians (AASL) released in October 2007 the *Standards for the 21st Century Learner:* http://www.ala.org/aasl/standards.
- ISTE National Standards, 2007: http://www.iste.org/standards/iste-standards
- Computing At School, 2013: www.computingatschool.org.uk
- College Board's AP Computer Science Principles, Computational Thinking Practices, 2016: http://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-curriculum-framework.pdf