DIGITAL LITERACY AND COMPUTER SCIENCE

Grades Kindergarten to 12

Massachusetts Curriculum Framework – 2016

MASSACHUSETTS DEPARTMENT OF ELEMENTARY AND SECONDARY EDUCATION

This document was prepared by the Massachusetts Department of Elementary and Secondary Education

The Massachusetts Department of Elementary and Secondary Education, an affirmative action employer, is committed to ensuring that all of its programs and facilities are accessible to all members of the public. We do not discriminate on the basis of age, color, disability, national origin, race, religion, sex, or sexual orientation.

Inquiries regarding the Department's compliance with Title IX and other civil rights laws may be directed to

the Human Resources Director, 75 Pleasant St., Malden, MA, 02148, 781-338-6105.

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

Massachusetts Department of Elementary and Secondary Education

75 Pleasant Street, Malden, Massachusetts 02148-4906 Telephone: (781) 338-3000
TTY: N.E.T. Relay 1-800-439-2370

Mitchell D. Chester
Commissioner

October 2016

Dear Colleagues,

In the last decade, changes in technology, communication, and the information life cycle have contributed to significant changes in our world. Increasingly, people are becoming technology creators as well as technology users. Meaningful participation in modern society requires fluency in the uses of, impact of, and ability to manipulate technology for living, learning, and working. Given this context, knowledge and skills included in the Digital Literacy and Computer Science (DLCS) standards are essential for all students. Student of all backgrounds should be prepared for personal and civic efficacy in the twenty-first century and should have the opportunity to consider innovative and creative technology-based careers of the future.

I am pleased to present to you the *Massachusetts Curriculum Framework for Digital Literacy and Computer Science*. The standards presented here:

- Address core concepts in four key domains: Computing and Society, Digital Tools and Collaboration, Computing Systems, and Computational Thinking.
- Integrate practices necessary to successfully act in a technological world.
- Present coherent progressions of core concepts and practices from grades K to 12.
- Complement other *Massachusetts Curriculum Frameworks*.

The *Framework* is a significant step forward from the state's prior *Instructional Technology* and *2008 Technological Literacy* standards. The *Framework* incorporates and updates expectations for Technological Literacy, reframes those as Digital Literacy, and adds expectations for Computer Science, which is now such a critical aspect of our daily lives. I would like to thank all of the individuals and groups that provided input, reviewed comments, and suggested edits to the standards.

I believe that the ability to effectively use and manipulate technology to solve complex problems is the new literacy skill of the twenty-first century. The *Framework* provides guidance and resources intended to help educators and professionals prepare students across the Commonwealth for such success.

Sincerely,

Mitchell D. Chester, Ed.D.
Commissioner of Elementary and Secondary Education

Table of Contents

Acknowledgements

The *Massachusetts Curriculum Framework for Digital Literacy and Computer Science* is the result of the contributions of many educators and professionals across the state. The Department of Elementary and Secondary Education (DESE) thanks all Massachusetts groups that contributed to the development of these standards and all individual teachers, administrators, professionals, faculty, and parents who took the time to provide thoughtful comments during the public comment periods.

DESE collaborated with the Massachusetts Computing Attainment Network (MassCAN), an alliance of organizations supporting computer science in school, and Massachusetts Computer Using Educators (MassCUE), the state's professional organizations for technology educators. A review panel, representing diverse perspectives relevant to both digital literacy and computer science, was essential in guiding this work.

Massachusetts Curriculum Framework for Digital Literacy and Computer Science Review Panel 2015–2016

Robert Ackerman, Principal, Lt Job Lane School, Bedford

W. Richard Adrion, Professor of Computer Science, University of Massachusetts Amherst

Kris Aho, Technology/Engineering Teacher, Collins Middle, Salem

Delcie Bean, President, Paragus Strategic IT, Springfield

Ayora Berry, Global Schools Manager, PTC Inc., Needham

David Birnbach, School Committee Member, Andover

Deborah Boisvert, Executive Director, Boston-Area Advanced Technological Education Connections (BATEC), University of Massachusetts Boston

George Bracket, Principal, George Brackett Associates; Co-founder, Codman Academy Charter Public School, Boston

Shelley Chamberlain, Executive Director, MassCUE, Waltham

Mark Check, Museum of Science, Boston

Kim Cochrane, Coordinator, C&I Technology and Education Technology, Framingham State University

Sandra Cole, Principal Systems Engineer, MITRE Corporation, Bedford

Benjamin Davison, Quantitative User Experience Researcher, Google, Inc., Cambridge

Lauren Dietz, Instructional Technology Specialist, Newton

David Digiammerino, Technology Teacher, Millis

Jennifer Dimmick, Library Teacher, Newton South High, Newton

Katheryn Fisler, Professor of Computer Science, Worcester Polytechnic Institute

Carla Fitzgerald, Social Studies Teacher, Marshall Middle School, Billerica

Robert Ford, Director of Technology, Lincoln

Rayna Freedman, Teacher and Instructional Technology Specialist, Jordan/Jackson Elementary, Mansfield

Jen Goree, Software Portfolio and Project Manager, The Concord Consortium

Wendy Haskell, Director of Technology and Library/Media, Falmouth

Diane Hauser, Senior Program Director, Educational Technology, Boston

John Heffernan, Technology Teacher, Williamsburg

Linda Herbison, Business Teacher, Montachusett Regional Vocational Technical, Fitchburg

Sue Howard, Instructional Technology Specialist, Concord

Eric Jewart, Software Developer, Ab Initio; Advanced Placement® Computer Science Teacher, Cambridge Rindge and Latin, Cambridge

Kimberle Koile, Research Scientist, Center for Educational Computing Initiatives, MIT, Cambridge

Nancy Lanza-Welsh, Computer Teacher, Justus C. Richardson Middle School, Dracut

Jeffery LeBoeuf, Teacher, Worcester Technical High, Worcester

Michael Looney, Curriculum Chair, Technology Department, Mashpee High, Mashpee

Fred Martin, Associate Professor, University of Massachusetts Lowell

Chad McGowan, Teacher, Ashland High, Ashland

Hamed Nilforoshan, CEO, AlNoor Games; Student, Wellesley Sr High, Wellesley

Jessica Parson Jarboe, Mathematics and Computer Science Teacher, Milton High, Milton

David Petty, Computer Science Teacher, STEM Coordinator, Winchester

Alec Resnick, Director, Sprout & Co. and Somerville STEAM Academy, Somerville

Vivian Siegel, Director of Education and Outreach, Broad Institute of MIT and Harvard, Cambridge

Susanne Steizer-Escobar, Computer Science Faculty and Department Chair, MassBay Community College, Wellesley

Peter Wong, Director of University Relations, Museum of Science, Boston

Department of Elementary and Secondary Education

Marilyn Decker, Director, Science, Technology/Engineering and Mathematics

Anne DeMallie, Computer Science and STEM Integration Specialist

Jake Foster, Director, Science, Technology/Engineering and Mathematics

Ken Klau, Director, Digital Learning

Andréa Cote, Coordinator of Professional Learning Services and Accessibility

Education Development Center

Joyce Malyn-Smith, Principal Investigator Focusing on Strategic Initiatives in STEM Workforce and Human Development

Massachusetts Computing Attainment Network

Kelly Powers, Project Director

Jim Stanton, Executive Director

Organizations and Key Documents Referenced

- K-12 Computer Science Standards (Computer Science Teachers Association, 2011)
- Massachusetts Technology Literacy Standards (DESE, 2008)
- Standards for the 21st Century Learner (American Association of School Librarians, 2007)
- ISTE Standards (International Society for Technology in Education, 2007)
- Computing at School (National Centre for Computing Education, 2013)
- AP® Computer Science Principles (College Board, 2016)

Digital Literacy and Computer Science Education for All Students

Vision

Digital literacy and computer science knowledge, reasoning, and skills are essential both to prepare students for personal and civic efficacy in the twenty-first century and to prepare and inspire a much larger and more diverse number of students to pursue the innovative and creative careers of the future. The abilities to effectively use and create technology to solve complex problems are the new and essential literacy skills of the twenty-first century.

Digital literacy and computer science standards in this *Framework* articulate critical learning outcomes for Kindergarten through Grade 12 to help prepare students for success in world. The standards represent the core elements of digital literacy and computer science and are intended to drive coherent, rigorous instruction, which results in the mastery and application of digital literacy and computer science knowledge, reasoning, and skills.

Key Features

1. **The standards include core concepts in four strands:** Computing and Society, Digital Tools and, Computing Systems, and Computational Thinking.

2. **The 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 skills needed to successfully use and create technology. When integrated with core concepts the practices in the standards define the types of performance students should be able to demonstrate as a result of learning digital literacy and computer science.

3. **The 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.

4. **The standards prepare students for post-secondary opportunities – civic, college, and career.** Digital literacy and computer science skills, knowledge and practices are essential to prepare all students for personal and civic efficacy, as well as the workplace. The importance of digital literacy and computer science knowledge and skills are pervasive in virtually all fields, from transportation to entertainment and the arts, from energy to the life sciences, from business and manufacturing to health care, from physics, chemistry and environmental sciences to all areas of research, and in many personal and civic decisions from telephones to garage doors to technological systems in communities.

5. **The standards complement other Massachusetts Curriculum Frameworks.** The standards in this *Framework* overlap in meaningful and substantive ways with standards from other academic disciplines 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

There are a few aspects of state learning standards that are important to keep in mind as curriculum and instruction is developed to help students learn the standards:

- 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 flexibility in arranging the standards in any order within a grade span to suit the needs of students.
- The practices articulate the dispositions and skills students acquire over time. Practices form the basis of analytical reasoning, specifically in the context of effectively using technology to support problem solving. Coupling practice with content gives the context for student performance.
- Curricula and assessment should be developed in a way, which builds students' knowledge and ability toward mastery of the standards. Effective instruction engages 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.
- Some standards have multiple functions or multiple components, the purpose of which is to convey the richness of expected outcomes.

Preparation for Career Options

Providing consistent exposure to curricula based on the standards during grades K-8 will create the necessary foundation for college- and career-focused learning in grades 9-12. In grades K-8, the standards afford opportunities to: (1) integrate core concepts and practices across the curriculum, and (2) more deeply explore digital literacy and computer science practices in contexts, such as introductory programming, data collection and analysis, robotics, etc., in specific subject areas or as exploratory courses. In grades 9-12, the standards provide opportunities for students to gain proficiency and incorporate substantive expectations of the AP® Computer Science Principles Curriculum Framework, the widely recognized benchmark for post–secondary preparation. With a strong foundation provided by this *Framework*, students will be well prepared for a variety of civic, college, and career options that include the use and creation of technology.

Guiding Principles

The vision of the *Massachusetts Curriculum Framework for Digital Literacy and Computer Science* is to engage students in digital literacy and computer science skills and concepts through the integration of practices, while making connections to what they know and the world they live in. The goal of the Guiding Principles is to help educators create relevant, rigorous, and coherent digital literacy and computer science programs that support student engagement, curiosity, computational thinking, and excitement for learning over time.

The following five Guiding Principles are intended to inform the development of programs that effectively engage students in learning the standards. They should guide the development and evaluation of programs in the schools and the broader community. Strong digital literacy and computer science programs effectively support student learning, so students are prepared for a dynamic world.

Learning

Digital literacy and computer science ideas should be explored in ways that stimulate curiosity, create enjoyment, and develop depth of understanding.

Students need to understand digital literacy and computer science concepts and use them effectively. The standards for digital literacy and computer science practice describe ways in which students increasingly engage with the subject matter as they grow in digital literacy and computer science maturity and expertise through the elementary, middle, and high school years.

Students should be actively engaged in designing, creating and inventing, discussing ideas, and applying their skills in interesting, thought-provoking situations. As students develop technology skills, it is important they apply these skills in their classroom, school, and life so that they will understand why these skills are important. For example, a student who needs to gather data in a science experiment and organize and manipulate the data in order to analyze the results will see a reason for learning about the features and function of a data collection tool and database. This is context-sensitive learning in which technology skills instruction is centered on the students' needs. Student understanding is further developed through ongoing reflection about cognitively demanding and worthwhile tasks.

Tasks should be designed to challenge students in multiple ways. Activities should build upon curiosity and prior knowledge and enable students to solve progressively deeper, broader, and more sophisticated problems. Digital literacy and computer science tasks reflecting sound and significant concepts should generate active classroom talk, promote the development of conjectures, and lead to an understanding of the necessity for digital literacy and computer science reasoning.

Teaching

An effective program is based on a carefully designed set of content standards that are clear and specific, focused, and articulated over time as a coherent sequence.

The sequence of topics and performances should be based on what is known about how students' knowledge, skill, and understanding develop over time. What and how students are taught should reflect not only the topics, but also the key ideas that determine how knowledge is organized and generated. Students should be asked to apply their learning and to show their thinking and understanding.

Creating and problem solving are the hallmark of computational thinking and an effective program. Skills in computational thinking require practice with a variety of problems, as well as a firm grasp of devices, tools, services, and techniques, and their underlying principles. Armed with this deeper knowledge, the student can then use digital literacy and computer science skills in a flexible way to create new products, attack various problems, and devise different ways of solving any particular problem. Problem solving calls for reflective thinking, persistence, learning from the ideas of others, and going back over one's own work with a critical eye. Students should be able to communicate their ideas and work collaboratively. They should analyze situations and justify their solutions.

As digital tools, computing devices, and services become an integral part of the learning environment, and as students gain the knowledge and skills to use, modify and create with them appropriately, new opportunities for learning open up. Dynamic geometric applets, for example, can help students visualize and understand complex mathematics concepts. Simulation software enables students to investigate models of real-world problems, such as climate change and population growth.

Success in creating and solving problems helps to create an abiding interest. Students learn to solve problems arising in everyday life, society, and the workplace.

For a program to be effective, it must also be taught by knowledgeable teachers.

Equity

All students should have a high-quality digital literacy and computer science program that prepares them for college and a career.

All Massachusetts students should have a high-quality digital literacy and computer science program that meets the goals and expectations of these standards and addresses students' individual interests and talents. The standards provide clear signposts along the way to the goal of college and career readiness for all students. The standards provide for a broad range of students, from those requiring tutorial support to those with talent in digital literacy and computer science. To promote achievement of these standards, teachers should encourage classroom talk, reflection, use of multiple problem-solving strategies, and a positive disposition toward digital literacy and computer science. They should have high expectations for all students. At every level of the education system, teachers should act on the belief that every child should learn challenging digital literacy and computer science concepts. Teachers and guidance personnel should advise students and parents about why it is important to take advanced courses in digital literacy and computer science and how this will prepare students for success in college and the workplace.

All students should have the benefit of quality instructional materials, good libraries, and adequate technology. All students must have the opportunity to learn and meet the same high standards. In order to meet the needs of the greatest range of students, digital literacy and computer science programs should provide the necessary intervention and support for those students who are below or above grade-level expectations. Practice and enrichment should extend beyond the classroom. Tutorial sessions, digital literacy and/or computer science clubs, competitions, and apprenticeships are examples of digital literacy and computer science activities that promote learning.

Because digital literacy and computer science is the cornerstone of our digital world, a comprehensive curriculum should include modeling activities that demonstrate the connections among disciplines.

Schools should also provide opportunities for communicating with experts in applied fields to enhance students' knowledge of these connections.

Literacy Across the Content Areas

An effective digital literacy and computer science program builds upon and develops students' literacy skills and knowledge.

Reading, writing, and communication skills are necessary elements of learning and engaging in digital literacy and computer science, as well as in other content areas. Supporting the development of students' literacy skills will allow them to deepen their understanding of digital literacy and computer science concepts and help them to determine the meanings of symbols, key terms and phrases, as well as develop reasoning skills that apply across the disciplines. In reading, teachers should consistently support students' ability to gain and deepen understanding of concepts from written material by helping them acquire comprehension skills and strategies, as well as specialized vocabulary and symbols. Digital literacy and computer science classrooms should make use of a variety of text materials and formats, including textbooks, notebook/journals, contextual problems, -Internet, and data presented in a variety of media.

In communicating, teachers should consistently support students' ability to reason and achieve deeper understanding of concepts, and to express their understanding in a focused, precise, and convincing manner.

In collaborating, teachers should facilitate opportunities for digital literacy and computer science discourse using precise language to convey ideas, communicate solutions, and support arguments.

Assessment

Assessment of student learning in digital literacy and computer science should take many forms to inform instruction and learning.

Assessment reflects classroom expectations and shows outcomes of student learning based on established knowledge and performance goals. The learning standards in this *Framework* are a key resource for setting such knowledge and performance objectives. Assessment assists teachers in improving classroom practice, planning curricula, developing self-directed learners, reporting student progress, and evaluating programs. It provides students with feedback about how their knowledge and skills are developing and what can be done to improve them. It lets parents know how well their children are doing and what needs to be done to help them do better.

Diagnostic information gained from different types of assessment enables teachers to adjust their day-to-day and week-to-week practices to foster greater student achievement. There are many types of assessment, such as paper-and-pencil testing, performance assessments, interviews, and portfolios, as well as less formal inventories, such as regular observation of student responses to instruction. Given the emphasis on practices in the standards, performance-based assessments should be developed that allow students to demonstrate what they have learned in the context of real-world problems and applications.

Planning and Support

An effective digital literacy and computer science program requires coherent district-wide planning and ongoing support for implementation.

An effective curriculum that addresses the learning standards of this *Framework* must be planned as a cohesive K–12 program. Teachers in different classrooms and at different levels should agree about what is to be taught in given grades. For example, middle school teachers should be able to expect that students coming from different elementary schools within a district share a common set of digital literacy and computer science understandings and skills and that the students they send on to high school will be well prepared for what comes next. In order for this expectation to be met, middle school teachers need to plan curricula in coordination with their elementary and high school colleagues and with district staff.

To facilitate planning, a district coordinator or administrator should be involved in articulating, coordinating, and implementing a district-wide (K–12) digital literacy and computer science curriculum. School districts should choose engaging, challenging, and accurate curriculum materials that are based on research into how children learn digital literacy and computer science.

When planning for the introduction of a new curriculum, it is important to explicitly identify how success will be measured. Indicators need to be determined and should be communicated to all stakeholders. Supervisors should monitor whether the curriculum is actually being used, how instruction has changed, and how student learning is being realized. Teacher teams, working across grade levels, should look at student work and other forms of assessment to determine whether there is evidence of achievement of the sought-for gains in student understanding.

Implementation of a new curriculum is accomplished over multiple years and requires opportunities for extensive professional development. Teachers must have both content knowledge and pedagogical expertise to use curricular materials in a way that enhances student learning. A well-planned program for professional development provides for both content learning and content-based pedagogical training. It is further recommended that middle and high school courses be taught by teachers who are certified in their areas and who are, therefore, very familiar with the safe use of materials, equipment, and processes.

Finally, students will be more likely to succeed in meeting the standards if they have the curricular and instructional support that encourages their interests in digital literacy and computer science. Further, students who are motivated to continue their studies and to persist in more advanced and challenging courses, are more likely to become STEM-engaged citizens and, in some cases, pursue careers in STEM fields. These affective goals should be an explicit focus of quality digital literacy and computer science programs.

Overview

The standards for Kindergarten to grade 12 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: remembering, understanding, applying, analyzing, evaluating, and creating.[1]

Vision

Digital literacy and computer science knowledge, reasoning, and skills are essential both to prepare students for personal and civic efficacy in the twenty-first century and to prepare and inspire a much larger and more diverse number of students to pursue the innovative and creative careers of the future. The abilities to effectively use and create technology to solve complex problems are the new and essential literacy skills of the twenty-first century.

<table>
<tr><th colspan="5">Learning Progression</th></tr>
<tr><th>Grade Spans</th><th colspan="4">Strands</th></tr>
<tr><td>K-2</td>
<td rowspan="4">Computing and Society [CAS]
a. Safety and Security
b. Ethics and Laws
c. Interpersonal and Societal Impact</td>
<td rowspan="4">Digital Tools and Collaboration [DTC]
a. Digital Tools
b. Collaboration and Communication
c. Research</td>
<td rowspan="4">Computing Systems [CS]
a. Computing Devices
b. Human and Computer Partnerships
c. Networks
d. Services</td>
<td rowspan="4">Computational Thinking [CT]
a. Abstraction
b. Algorithms
c. Data
d. Programming and Development
e. Modeling and Simulation</td></tr>
<tr><td>3-5</td></tr>
<tr><td>6-8</td></tr>
<tr><td>9-12</td></tr>
<tr><th colspan="5">Practices</th></tr>
<tr><td colspan="5">Connecting, Creating, Abstracting, Analyzing, Communicating, Collaborating, Research</td></tr>
</table>

1 Anderson, L.W. (Ed.), Krathwohl, D.R. (Ed.), Airasian, P.W., Cruikshank, K.A., Mayer, R.E., Pintrich, P.R., Raths, J., & Wittrock, M.C. (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's Taxonomy of Educational Objectives (Complete edition). New York: Longman.

Strands

Progression of Core Concepts

The Kindergarten through grade 12 standards 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. DLCS practices are integrated throughout the standards and help define performance expectations that specify what students should know and be able to do.

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. Society values many different computing innovations. 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). 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 do not have defined rules or are dynamic in nature, are best done by humans. Many tasks, however, 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. These skills and knowledge inform the decision to use technology in creating, innovating, or solving a problem or sub-problem.

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) **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.

e) **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.

Practices

Practices cultivate the internalization of dispositions and skills that students apply to solve digital literacy and computer science problems. 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 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 addressing 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" solutions 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 a 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 or 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; and
- Exchanging knowledge and feedback with a partner or team member.

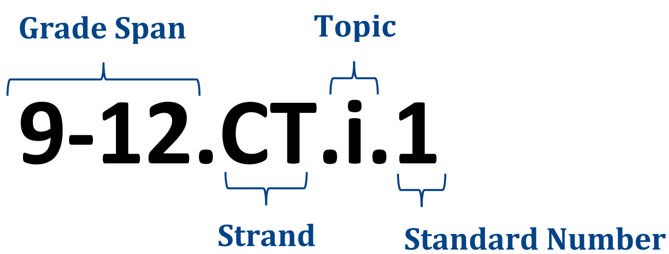
7. Researching

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 for the standards has four components: grade span (K-2, 3-5, 6-8, 9-12); strand (CAS, DTC, CS, CT); topic (a, b, c, etc.); and standard number (1, 2, 3, etc.).



Kindergarten to Grade 2

Early elementary school students are introduced to foundational concepts by integrating basic digital literacy skills with simple ideas about computational thinking. They learn that tools help people do things better, or more easily, or do some things that could otherwise not be done at all. Through the exploration of differences between humans, computing devices, and digital tools, students begin to understand if, when, and how they should use technology.

Kindergarten through grade 2 standards integrate all seven practices. Standards in this grade span ask students to demonstrate the ability to:

Digital Literacy and Computer Science Practices

1. Creating
2. Connecting
3. Abstracting
4. Analyzing
5. Communicating
6. Collaborating
7. Researching

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. The standards are designed with a focus on active learning, creativity, and exploration. Standards for the earliest grade span allow teacher flexibility in deciding when students are ready to use technology. Basic technology skills may be learned through the use of manipulatives, pencil-and-paper, and other manual methods through which children acquire basic skills. Many skills introduced in this grade span will be further developed in later grade spans.

Computing and Society [CAS]

- Understand basic safety and security concepts and basic understanding of safe information sharing.
- Explore what is means to be a good digital citizen.
- Observe and describe how people use technology and how technology can influence people.

Digital Tools and Collaboration [DTC]

- Develop basic use of digital tools and research skills to create simple artifacts.
- Develop basic use of digital tools to communicate or exchange information.

Computing Systems [CS]

- Understand that computing devices take many forms and have different components.
- Consider basic structures of computing systems and networks.
- Explore human and computer differences to determine when technology is beneficial.

Computational Thinking [CT]

- Explore abstraction through identification of common attributes.
- Create and enact a simple algorithm.
- Understand how information can be collected, used, and presented with computing devices or digital tools.
- Create a simple computer "program."
- Use basic models and simulations.

Kindergarten to Grade 2: Computing and Society [CAS]

Safety and Security [K-2.CAS.a]

1. Demonstrate proper ergonomics (e.g., body position, stretching) when using devices.
2. Use electrical devices safely and in moderation (e.g., unplug devices by pulling the plug rather than the cord, do not mix water/food and electric devices, avoid gaming and walking).
3. Care for devices appropriately (e.g., handling devices gently, completely shutting down devices when not in use, storing devices in the appropriate container).
4. Explain that a password helps protect the privacy of information.
5. Identify safe and unsafe examples of online communications.
6. Explain why we keep personal information (e.g., name, location, phone number, home address) private.
7. Identify which personal information (e.g., user name or real name, school name or home address) should and should not be shared online and with whom.
8. Explain why it is necessary to report inappropriate electronic content or contact.

Ethics and Laws [K-2.CAS.b]

1. Define good digital citizenship as using technology safely, responsibly, and ethically.
2. Demonstrate responsible use of computers, peripheral devices, and resources as outlined in school rules [Acceptable Use Policy (AUP) for K-2].
3. Explain that most digital artifacts have owners.
4. Explain the importance of giving credit to media creators/owners when using their work.

Interpersonal and Societal Impact [K-2.CAS.c]

5. Identify and describe how people (e.g., students, parents, police officers) use many types of technologies in their daily work and personal lives.
6. Recognize when the purpose of content is to provide information or to influence you to act.

Kindergarten to Grade 2: Digital Tools and Collaboration [DTC]

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).
2. Identify, locate, and use letters, numbers, and special keys on a keyboard (e.g., Space Bar, Shift, Delete).
3. Create a simple digital artifact.
4. Use appropriate digital tools individually and collaboratively to create, review, and revise simple artifacts that include text, images and audio.

Collaboration and Communication [K-2.DTC.b]

1. Collaboratively use digital tools and media resources to communicate key ideas and details in a way that informs, persuades, and/or entertains.
2. Use a variety of digital tools to exchange information and feedback with teachers.
3. Use a variety of digital tools to present information to others.

Research [K-2.DTC.c]

1. Conduct basic keyword searches to gather information from teacher-provided digital sources (e.g., online library catalog, databases).
2. Create an artifact individually and collaboratively that answers a research question, while clearly expressing thoughts and ideas.
3. Acknowledge and name sources of information or media (e.g., title of book, author of book, website).

Kindergarten to Grade 2: Computing Systems [CS]

Computing Devices [K-2CS.a]

1. Identify different kinds of computing devices in the classroom and other places (e.g., laptops, tablets, smart phones, desktops).
2. Identify visible components of computing devices (e.g., keyboard, screen, monitor, printer, pointing device).
3. Explain that computing devices function when applications, programs, or commands are executed.
4. 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).

Human and Computer Partnerships [K-2.CS.b]

1. Explain that computing devices are machines that are not alive but can be used to help humans with tasks.
2. Recognize that some tasks are best completed by humans and others by computing devices (e.g., a human might be able to rescue someone in a normal environment, but robots would be better to use in a dangerous environment).
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).

Networks [K-2.CS.c]

1. Explain that networks link computers and devices locally and around the world allowing people to access and communicate information.

Services [K-2.CS.d]

There are no standards in this strand for this grade span.

Kindergarten to Grade 2: Computational Thinking [CT]

Abstraction [K-2.CT.a]

1. List the attributes of a common object, for example, cars have a color, type (e.g., pickup, van, sedan), number of seats, etc.

Algorithms [K-2.CT.b]

1. Define an algorithm as a sequence of defined steps.
2. Create a 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).
3. Enact an algorithm using tangible materials (e.g., manipulatives, your body) or present the algorithm in a visual medium (e.g., storyboard).

Data [K-2.CT.c]

1. Identify different kinds of information (e.g., text, charts, graphs, numbers, pictures, audio, video, collections of objects.)
2. Identify, research, and collect information on a topic, issue, problem, or question using age-appropriate digital technologies.
3. Individually and collaboratively, propose a solution to a problem or question based on an analysis of information.
4. Individually and collaboratively, create information visualizations (e.g., charts, infographics).
5. Explain that computers can save information as data that can be stored, searched, retrieved, and deleted.

Programming and Development [K-2.CT.d]

1. Define a computer program as a set of commands created by people to do something.
2. Explain that computers only follow the program's instructions.
3. Individually or collaboratively, create a simple program using visual instructions or tools that do not require a textual programming language (e.g., "unplugged" programming activities, a block-based programming language).

Modeling and Simulation [K-2.CT.e]

1. Describe how models represent a real-life system (e.g., globe, map, solar system, digital elevation model, weather map).
2. Define simulation and identify the concepts illustrated by a simple simulation (e.g., growth and health, butterfly life cycle).

Grades 3 to 5

Upper elementary students learn to differentiate tasks that are best done by computing systems or digital tools and those best done by humans. Students explore a variety of computing devices and digital tools and further develop their computational thinking problem solving skills. As students progress through grades 3–5, they begin to evaluate the uses and limitations of existing artifacts and modify parts of existing artifacts to develop something new. Students are able to describe and document their computational work in writing, using presentation tools and through demonstrations of their work.

Grade 3 to 5 standards integrate all seven practices. Standards in this grade span ask students to demonstrate the ability to:

Digital Literacy and Computer Science Practices

1. Creating
2. Connecting
3. Abstracting
4. Analyzing
5. Communicating
6. Collaborating
7. Researching

With increased maturity, students in third through fifth grade are able to engage in learning in ways that are both more systematic and creative. Upper elementary is a critical time to engage students in the DLCS practices. Students' capabilities as creators and problem solvers build on their experiences in K–2. They continue to develop concepts through exploration, discovery, and creativity with the guidance, support, and encouragement of their educator. Standards for this grade span allow teacher flexibility in deciding when students are ready to use technology.

Computing and Society [CAS]

- Understand safety and security concepts, safe and appropriate use of technology, and how to deal with cyberbullying.
- Demonstrate responsible use of technology, digital content, and interactions.
- Observe and describe how technology can influence people.
- Basic understanding of digital media messaging and equity of access to technology.

Digital Tools and Collaboration [DTC]

- Use digital tools and keyboarding skills to publish multimedia artifacts.
- Use digital tools to communicate or exchange information.
- Develop intermediate research skills to create artifacts and attribute credit.

Computing Systems [CS]

- Understand different computing devices and their components.
- Use different computing devices and troubleshoot and solve simple problems.
- Differentiate tasks that are best done by computing systems and humans.
- Understand the components of a network and basic network authentication.
- Basic understanding of services.

Computational Thinking [CT]

- Create a new representation and breakdown a larger problem into sub problems.
- Write, debug, and analyze an algorithm.

- Understand databases and organizing and transforming data.
- Write, debug, and correct programs using successively sophisticated techniques.
- Create a model and use data from a simulation.

Grades 3 to 5: Computing and Society [CAS]

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.
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. Identify appropriate and inappropriate uses of technology when posting to social media, sending e-mail or texts, and browsing the Internet.
4. Explain the proper use and operation of security technologies (e.g., passwords, virus protection software, spam filters, popup blockers, cookies).
5. Describe ways to employ safe practices and avoid the potential risks/dangers associated with various forms of online communications, downloads, linking, Internet purchases, advertisements, and inappropriate content within constrained environments.
6. Identify different types of cyberbullying (e.g., harassment, flaming, excluding people, outing, and impersonation).
7. Explain that if you encounter cyberbullying or other inappropriate content, you should immediately tell a responsible adult (e.g., teacher, parent).

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 (AUP)].
2. Describe the difference between digital artifacts that are open or free and those that are protected by copyright.
3. Explain the guidelines for the fair use of downloading, sharing, or modifying of digital artifacts.
4. Describe the purpose of copyright and the possible consequences for inappropriate use of digital artifacts that are protected by copyright.
5. Explain that laws exist (e.g., Section 508, Telecommunication Act of 1996) that help ensure that people with disabilities can access electronic and information technology.

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, e-mail ads).
2. Explain why websites, digital resources, and artifacts may include advertisements and collect personal information.
3. Define the digital divide as unequal access to technology on the basis of differences, such as income, education, age, and geographic location.
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, allows them to buy and sell things).

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

Grades 3 to 5: Digital Tools and Collaboration [DTC]

Digital Tools [3-5.DTC.a]

1. Type five words-per-minute times grade level (e.g., for Grade 5, type 25 words/minute).
2. Navigate between local, networked, or online/cloud environments and transfer files between each (upload/download).
3. Use digital tools (local and online) to manipulate and publish multimedia artifacts.

Collaboration and Communication [3-5.DTC.b]

1. Communicate key ideas and details individually or collaboratively in a way that informs, persuades, and/or entertains using digital tools and media-rich resources.
2. Collaborate through online digital tools under teacher supervision.

Research [3-5.DTC.c]

1. Identify digital information sources to answer research questions (e.g., online library catalog, online encyclopedias, databases, websites).
2. Perform searches to locate information using two or more key words and techniques to refine and limit such searches.
3. Evaluate digital sources for accuracy, relevancy, and appropriateness.
4. Gather and organize information from digital sources by quoting, paraphrasing, and/or summarizing.
5. Create an artifact that answers a research question and clearly communicates thoughts and ideas.
6. Cite text-based sources using a school- or district-adopted format.
7. Provide basic source information [e.g., Uniform Resource Locator (URL), date accessed] for non-text-based sources (e.g., images, audio, video).

Grades 3 to 5: Computing Systems [CS]

Computing Devices [3-5.CS.a]

1. Identify a broad range of computing devices (e.g., computers, smart phones, tablets, robots, e-textiles) and appropriate uses for them.
2. Describe the function and purpose of various input and output devices (e.g., monitor, keyboard, speakers, controller, probes, sensors, Bluetooth transmitters, synthesizers).
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).
4. Identify and solve simple hardware and software problems that may occur during everyday use (e.g., power, connections, application window or toolbar).
5. Describe the differences between hardware and software.
6. Identify and explain that some computing functions are always active (e.g., locations function on smart phones).

Human and Computer Partnerships [3-5.CS.b]

1. Compare and contrast human and computer performance on similar tasks (e.g., sorting alphabetically, finding a path across a cluttered room) to understand which is best suited to the task.
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, to do things they could not do otherwise.
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’).

Networks [3-5.CS.c]

1. Describe how a network is made up of a variety of components and identify the common components (e.g., links, nodes, networking devices).
2. Describe the need for authentication of users and devices as it relates to access permissions, privacy, and security.
3. Define and explain why devices are numbered/labeled in networks [e.g., the World Wide Web Uniform Resource Locator (URL), the Internet Protocol (IP) address, the Machine Access Code (MAC)].
4. Recognize that there are many sources of and means for accessing information within a network (e.g., websites, e-mail protocols, search engines)

Services [3-5.CS.d]

1. Identify common services (e.g., driving directions apps that access remote map services, digital personal assistants that access remote information services).

Grades 3 to 5: Computational Thinking [CT]

Abstraction [3-5.CT.a]

1. Use numbers or letters to represent information in another form (e.g., secret codes, Roman numerals, abbreviations).
2. Organize information in different ways to make it more useful/relevant (e.g., sorting, tables).
3. Make a list of sub-problems to consider, while addressing a larger problem.

Algorithms [3-5.CT.b]

1. Define an algorithm as a sequence of instructions that can be processed by a computer.
2. Recognize that different solutions exist for the same problem (or sub-problem).
3. Use logical reasoning to predict outcomes of an algorithm.
4. Individually and collaboratively create an algorithm to solve a problem (e.g., move a character/robot/person through a maze).
5. Detect and correct logical errors in various algorithms (e.g., written, mapped, live action, or digital).

Data [3-5.CT.c]

1. Describe examples of databases from everyday life (e.g., library catalogs, school records, telephone directories, contact lists).
2. Individually and collaboratively collect and manipulate data to answer a question using a variety of computing methods (e.g., sorting, totaling, averaging) and tools (such as a spreadsheet) to collect, organize, graph, and analyze data.

Programming and Development [3-5.CT.d]

1. Individually and collaboratively create, test, and modify a program in a graphical environment (e.g., block-based visual programming language).
2. Use arithmetic operators, conditionals, and repetition in programs.
3. Use interactive debugging to detect and correct simple program errors.
4. Recognize that programs need known starting values (e.g., set initial score to zero in a game).

Modeling and Simulation [3-5.CT.e]

1. Individually and collaboratively create a simple model of a system (e.g., water cycle, solar system) and explain what the model shows and does not show.
2. Identify the concepts, features, and behaviors illustrated by a simulation (e.g., object motion, weather, ecosystem, predator/prey) and those that were not included.
3. Individually and collaboratively, use data from a simulation to answer a question.

Grades 6 to 8

The goal for middle school students is to define problems more precisely, to conduct a more thorough process of selecting the best devices, tools, and solutions. Students learn to differentiate problems or sub-problems that are best solved by computing systems or digital tools and those best solved by humans. Students further develop their computational thinking problem solving skills, which facilitates the use of technology.

Grade 6 to 8 standards integrate all seven practices. Standards in this grade span ask students to demonstrate the ability to:

Digital Literacy and Computer Science Practices

1. Creating
2. Connecting
3. Abstracting
4. Analyzing
5. Communicating
6. Collaborating
7. Researching

By the time students reach middle school, they should have had numerous experiences in using technology to create artifacts and solve problems. Active engagement of middle school students with the practices is critical: students generally make up their minds about whether they identify with science and engineering by the time they leave grade 8. Students should have opportunities to develop the skills necessary for a meaningful progression of development in order to engage in reasoning, which is critical to success in civic life, post-secondary education, and career.

Computing and Society [CAS]

- Understand safety and security concepts, online identity and privacy, and how to deal with cyberbullying and inappropriate content.
- Demonstrate responsible use of technology and laws regarding ownership of material/ideas, licensing, and fair use.
- Understand consequences of inappropriate technology use, including harassment and sexting.
- Examine the impact of emerging technology in schools, communities, and societies.
- Evaluate digital media bias and messaging.

Digital Tools and Collaboration [DTC]

- Use a variety of digital tools to create artifacts, online content, and online surveys.
- Understand that different digital tools have different uses.
- Communicate and publish online.
- Advance research skills.

Computing Systems [CS]

- Understand hardware and software components of a computing device; troubleshoot hardware and software problems.
- Use a variety of computing devices to manipulate data.
- Differentiate tasks/problems best solved by computing systems or by humans.
- Understand that network components carry out specific functions to connect computing devices, people, and services.
- Understand the capabilities services can provide.

Computational Thinking [CT]

- Create a new representation, define functions, and use decomposition.
- Write, debug, and analyze advanced algorithms and basic programs.
- Understand how computing devices represent and manipulate information.
- Create, modify, and manipulate databases.
- Use a variety of data collection devices.
- Create a model and use and modify a simulation for analysis.

Grades 6 to 8: Computing and Society [CAS]

Safety and Security [6-8.CAS.a]

1. Identify threats and actively protect devices and networks from viruses, intrusion, vandalism, and other malicious activities.
2. Describe how cyberbullying can be prevented and managed.
3. Explain the connection between the persistence of data on the Internet, personal online identity, and personal privacy.
4. Describe and use safe, appropriate, and responsible practices (netiquette) when participating in online communities (e.g., discussion groups, blogs, social networking sites).
5. Differentiate between appropriate and inappropriate content on the Internet.

Ethics and Laws [6-8.CAS.b]

1. Explain how copyright law and licensing protect the owner of intellectual property.
2. Explain possible consequences of violating intellectual property law and plagiarism.
3. Apply fair use for using copyrighted materials (e.g., images, music, video, text).
4. Identify the legal consequences of sending or receiving inappropriate content (e.g., cyberbullying, harassment, sexting).
5. Differentiate among open source and proprietary software licenses and their applicability to different types of software and media.
6. Demonstrate compliance with the school's Acceptable Use Policy (AUP).
7. Identify software license agreements and application permissions.
8. Explain positive and malicious purposes of hacking.
9. License original content and extend license for sharing in the public domain (e.g., creative commons).

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.
2. Identify and discuss the technology proficiencies needed in the classroom and the workplace, and how to meet the needs.
3. Relate the distribution of computing resources in a global society to issues of equity, access, and power.
4. Evaluate how media and technology can be used to distort, exaggerate, and misrepresent information.
5. Evaluate the bias of digital information sources, including websites.

Grades 6 to 8: Digital Tools and Collaboration [DTC]

Digital Tools [6-8.DTC.a]

1. Identify and explain the strengths, weaknesses, and capabilities of a variety of digital tools.
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 drawings.).
3. Integrate information from multiple file formats into a single artifact.
4. Individually and collaboratively, use advanced tools to design and create online content (e.g., digital portfolio, multimedia, blog, webpage).
5. Individually and collaboratively, develop and conduct an online survey.

Collaboration and Communication [6-8.DTC.b]

1. Communicate and publish key ideas and details individually or collaboratively in a way that informs, persuades, and/or entertains using a variety of digital tools and media-rich resources.
2. Collaborate synchronously and asynchronously through online digital tools.
3. Demonstrate ability to communicate appropriately through various online tools (e.g., e-mail, social media, texting, blog comments).

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).
2. Evaluate quality of digital sources for reliability, including currency, relevancy, authority, accuracy, and purpose of digital information.
3. Gather, organize, and analyze information from digital sources by quoting, paraphrasing, and/or summarizing.
4. Create an artifact, individually and collaboratively, that answers a research question and communicates results and conclusions.
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, video).

Grades 6 to 8: Computing Systems [CS]

Computing Devices [6-8.CS.a]

1. Describe the main functions of an operating system.
2. Recognize that there is a wide range of application software.
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)].
4. Identify and describe the use of sensors, actuators, and control systems in an embedded system (e.g., a robot, an e-textile, installation art, smart room).
5. Individually and collaboratively design and demonstrate the use of a device (e.g., robot, e-textile) to accomplish a task.
6. Use a variety of computing devices [e.g., probes, sensors, handheld devices, Global Positioning System (GPS)] to individually and collaboratively collect, analyze, and present information for content-related problems.
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, sound) that occur during everyday computer use.

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 excels.
2. Describe how humans and machines interact to solve problems that cannot be solved by either alone (e.g., "big data" experiments that involve drawing conclusions by analyzing vast amounts of data).

Networks [6-8.CS.c]

1. Explain the difference between physical (wired), local and wide area, wireless, and mobile networks.
2. Model the components of a network, including devices, routers, switches, cables, wires, and transponders.
3. Describe how information, both text and non-text, is translated and communicated between digital devices over a computer network.

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, a mobile device that uses location services for navigation).

Grades 6 to 8: Computational Thinking [CT]

Abstraction [6-8.CT.a]

1. Describe how data is abstracted by listing attributes of everyday items to represent, order and compare those items (e.g., street address as an abstraction for locations; car make, model, and license plate number as an abstraction for cars).
2. Define a simple function that represents a more complex task/problem and can be reused to solve similar tasks/problems.
3. Use decomposition to define and apply a hierarchical classification scheme to a complex system, such as the human body, animal classification, or in computing.

Algorithms [6-8.CT.b]

1. Design solutions that use repetition and conditionals.
2. Use logical reasoning to predict outputs given varying inputs.
3. Individually and collaboratively, decompose a problem and create a sub-solution for each of its parts (e.g., video game, robot obstacle course, making dinner).
4. Recognize that more than one algorithm can solve a given problem.
5. Recognize that boundaries need to be taken into account for an algorithm to produce correct results.

Data [6-8.CT.c]

1. Demonstrate that numbers can be represented in different base systems (e.g., binary, octal, and hexadecimal) and text can be represented in different ways [e.g., American Standard Code for Information Interchange (ASCII)].
2. Describe how computers store, manipulate, and transfer data types and files (e.g., integers, real numbers, Boolean Operators) in a binary system.
3. Create, modify, and use a database (e.g., define field formats, add new records, manipulate data), individually and collaboratively, to analyze data and propose solutions for a task/problem.
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 (scientific notation and percentages), charts, tables, and graphs.
5. Select and use data-collection technology (e.g., probes, handheld devices, geographic mapping systems) to individually and collaboratively gather, view, organize, analyze, and report results for content-related problems.

Programming and Development [6-8.CT.d]

1. Individually and collaboratively compare algorithms to solve a problem, based on a given criteria (e.g., time, resource, accessibility).
2. Use functions to hide the detail in a program.
3. Create a program, individually and collaboratively, that implements an algorithm to achieve a given goal.

4. Implement problem solutions using a programming language, including all of the following: looping behavior, conditional statements, expressions, variables, and functions.
5. Trace programs step-by-step in order to predict their behavior.
6. Use an iterative approach in development and debugging to understand the dimensions of a problem clearly.

Modeling and Simulation [6-8.CT.e]

1. Create a model of a real-world system and explain why some details, features and behaviors were required in the model and why some could be ignored.
2. Use and modify simulations to analyze and illustrate a concept in depth (e.g., light rays/mechanical waves interaction with materials, genetic variation).
3. Select and use computer simulations, individually and collaboratively, to gather, view, analyze, and report results for content-related problems (e.g., migration, trade, cellular function).

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 collaboratively 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 Literacy and Computer Science Practices

1. Creating
2. Connecting
3. Abstracting
4. Analyzing
5. Communicating
6. Collaborating
7. Researching

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.

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.

Grades 9 to 12: Computing and Society [CAS]

Safety and Security [9-12.CAS.a]

1. Evaluate and design an ergonomic work environment.
2. Explain safe practices when collaborating online, including how to anticipate potentially dangerous situations.
3. Construct strategies to combat cyberbullying/harassment.
4. Identify the mental health consequences of cyberbullying/harassment.
5. Explain how peer pressure in social computing settings influences choices.
6. Apply strategies for managing negative peer pressure and encouraging positive peer pressure.

Ethics and Laws [9-12.CAS.b]

1. Model mastery of the school's Acceptable Use Policy (AUP).
2. Identify computer-related laws and analyze their impact on digital privacy, security, intellectual property, network access, contracts, and consequences of sexting and harassment.
3. Discuss the legal and ethical implications associated with malicious hacking and software piracy.
4. Interpret software license agreements and application permissions.

Interpersonal and Societal Impact [9-12.CAS.c]

1. Explain the impact of the digital divide on access to critical information.
2. Discuss the impact of computing technology on business and commerce (e.g., automated tracking of goods, automated financial transaction, e-commerce, cloud computing).
3. Describe the role that assistive technology can play in people's lives.
4. Create a digital artifact that is designed to be accessible (e.g., closed captioning for audio, alternative text for images).
5. Analyze the beneficial and harmful effects of computing innovations (e.g., social networking, delivery of news and other public media, intercultural communication).
6. Cultivate a positive web presence (e.g., digital resume, portfolio, social media).
7. Identify ways to use technology to support lifelong learning.
8. Analyze the impact of values and points of view that are presented in media messages (e.g., racial, gender, political).
9. Discuss the social and economic implications associated with malicious hacking, software piracy, and cyber terrorism.

Grades 9 to 12: Digital Tools and Collaboration [DTC]

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).
2. Select digital tools or resources based on their efficiency and effectiveness to use for a project or assignment and justify the selection.

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.
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).

Research [9-12.DTC.c]

1. Generate, evaluate, and prioritize questions that can be researched through digital resources or tools.
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).
3. Evaluate digital sources needed to solve a given problem (e.g., reliability, point of view, relevancy).
4. Gather, organize, analyze, and synthesize information using a variety of digital tools.
5. Create an artifact that answers a research question, communicates results and conclusions, and cites sources.

Grades 9 to 12: Computing Systems [CS]

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.
2. Examine how the components of computing devices are controlled by and react to programmed commands.
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).
4. Explain and demonstrate how specialized computing devices can be used for problem solving, decision-making and creativity in all subject areas.
5. Describe how computing devices manage and allocate shared resources [e.g., memory, Central Processing Unit (CPU)].
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.

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).

Networks [9-12.CS.c]

1. Explain how network topologies and protocols enable users, devices, and systems to communicate with each other.
2. Examine common network vulnerabilities (e.g., cyberattacks, identity theft, privacy) and their associated responses.
3. Examine the issues (e.g., latency, bandwidth, firewalls, server capability) that impact network functionality.

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).
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 to 12: Computational Thinking [CT]

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.

Algorithms [9-12.CT.b]

1. Recognize that the design of an algorithm is distinct from its expression in a programming language.
2. Represent algorithms using structured language, such as pseudocode.
3. Explain how a recursive solution to a problem repeatedly applies the same solution to smaller instances of the problem.
4. Describe that there are ways to characterize how well algorithms perform and that two algorithms can perform differently for the same task.
5. Explain that there are some problems, which cannot be computationally solved.

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).
2. Create an appropriate multidimensional data structure that can be filtered, sorted, and searched (e.g., array, list, record).
3. Create, evaluate, and revise data visualization for communication and knowledge.
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).
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.

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).
2. Decompose a problem by defining functions, which accept parameters and produce return values.
3. Select the appropriate data structure to represent information for a given problem (e.g., records, arrays, lists).
4. Analyze trade-offs among multiple approaches to solve a given problem (e.g., space/time performance, maintainability, correctness, elegance).
5. Use appropriate looping structures in programs (e.g., FOR, WHILE, RECURSION).
6. Use appropriate conditional structures in programs (e.g., IF-THEN, IF-THEN-ELSE, SWITCH).
7. Use a programming language or tool feature correctly to enforce operator precedence.
8. Use global and local scope appropriately in program design (e.g., for variables).

9. Select and employ an appropriate component or library to facilitate programming solutions [e.g., turtle, Global Positioning System (GPS), statistics library].
10. Use an iterative design process, including learning from making mistakes, to gain a better understanding of the problem domain.
11. Engage in systematic testing and debugging methods to ensure program correctness.
12. Demonstrate how to document a program so that others can understand its design and implementation.

Modeling and Simulation [9-12.CT.e]

1. Create models and simulations to help formulate, test, and refine hypotheses.
2. Form a model from a hypothesis generated from research and run a simulation to collect and analyze data to test that hypothesis.

Glossary

A

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 allows one to focus on the problem.

Abstraction (product). 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). 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 AUP.

Algorithm. A set of unambiguous rules or instructions to achieve a particular objective.[2]

Alphanumeric. A combination of alphabetic and numeric characters, including other symbols, such as punctuation and mathematical symbols.[2] Example: ABC123

Annotated bibliography. A summary and/or evaluation of each of the sources in a bibliography.

Application. Software (program) used by people to accomplish a task.

Array. A data structure comprising a collection of values of the same type accessible through an **index**.[2] 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 piece of information which determines the properties of a field or **tag** in a **database** or a **string** of **characters** in a display.[2] 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, what, they or it claim(s) to be. Example: Some websites use a combination of e-mail address and password as a means of authentication.

[2] Computing At School, accessed April 2019: https://www.computingatschool.org.uk/

[3] Individuals with Disabilities Education Act (IDEA), accessed April 2019: https://sites.ed.gov/idea/

B

Bandwidth. 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]

Big data. Extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions.[5]

Binary. A method of **encoding** data using two symbols, 1 and 0.[2]

Binary number. A number written in the Base-2 Number System.[2] Example: the number 4 written in binary is 100.

Bit. A basic unit of data that stores one **binary** value, 1 or 0.[2]

Boolean. A data type with only two values: TRUE or FALSE.[2]

Bridge. A device that creates a single network from multiple networks or network segments.[6]

Browser cookie. A small piece of text recording activity about websites one visits stored on one's computer.[2]

C

Class. In **object-oriented programming**, an extensible program-code template for creating objects, including associating variables with objects ("member variables") and the processes objects perform ("methods").

Code. Any set of instructions expressed in a **programming language**.[2]

Coding. The act of writing computer programs in a **programming language**.[2]

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**.[2]

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.

[4] TechTerms, accessed May 2015: http://techterms.com/

[5] Breur, T. (2016). Statistical power analysis and the contemporary "crisis" in social sciences. *Journal of Marketing Analytics*. 4 (2–3): 61–65.

[6] Mishra, S., Jena, L., & Pradhan, A. (2012). Networking devices and topologies: A succinct study. *International Journal*, *2*(11).

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.[7]

Computing. Any goal-oriented activity requiring, benefiting from, or creating **algorithmic** processes.[8]

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 statement/conditional expression/conditional construct. Features of a **programming language** which perform different computations or actions depending on whether a programmer-specified **Boolean** condition evaluates to true or false.[9]

Constant. In computer programming, a fixed **value** in a program. Minimum and maximum amounts, dates, prices, headlines and error messages are examples.[19]

Central Processing Unit (CPU). The device within a computer that executes instructions.[2]

Character. A single alphabetic letter, numeric digit, or special symbol such as a decimal point or comma. A character is equivalent to a byte; for example, 50,000 characters take up 50,000 bytes. The word “character” takes up nine bytes.[19]

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.

[7] Hubwieser, P., Armoni, M., Brinda, T., Dagiene, V., Diethelm, I., Giannakos, M. N., ... & Schubert, S. (2011, June). Computer science/informatics in secondary education. In *Proceedings of the 16th annual conference reports on Innovation and technology in computer science education-working group reports* (pp. 19-38). ACM.

[8] Denning. P.J., et al. (1989). Computing as a discipline. Communications of the ACM. *Association for Computing Machinery*. 32: 9–23.

[9] Stump, A. (2013). *Programming language foundations*. John Wiley & Sons.

D

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.[10]

Data structure. A particular way to store and organize data within a computer program.[2]

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 computer programs. The rules are:

NOT (A and B) = NOT A or NOT B

NOT (A or B) = NOT A and NOT B

Debugging. The process of finding and correcting errors in **programs**.[2]

Decimal. The base 10 number system.[2]

Decomposition. Breaking a problem or system down into its components.[2]

Device. See **computing device**.

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.[2] Example: computer animation, LED infused clothing, interactive sculpture, 3-D printed objects, songs.

Digital citizenship. The norms of appropriate, responsible behavior with regard to the use of technology.

Digital creator. A person who makes **digital artifacts**.[2]

Digital device. An electronic device that can receive, store, process, or send digital information.[27]

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.

[10] Ullman, J. D. (2001). *A first course in database systems*. Pearson Education India.

Digital literacy.

> The ability to use digital technology, communication tools or **networks** to locate, evaluate, use, and create information.[11]
>
> The ability to understand and use information in multiple formats from a wide range of sources when it is presented via computers.[12]
>
> 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.[13]

Digital media. Media encoded in a computer-readable form.[2]

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).

E

Encode. To assign a code to represent data.[19]

Ergonomics. Designing and arranging things people use so the people and things interact most efficiently and safely.

Expression. In a **programming language**, a combination of explicit values, **constants**, **variables**, **operators**, and **functions** interpreted according to the particular rules of precedence and of association which computes and then produces (returns, in a **stateful** environment) another value.[2] Example: b = a + 2.

[11] Murray, J. (2018). *Technology in the classroom: What is digital literacy*. http://www.teachhub.com/technology-classroom-what-digital-literacy

[12] Gilster, P., & Glister, P. (1997). *Digital literacy*. New York: Wiley Computer Pub.

[13] Jones, B., & Flannigan, S. L. (2006). Connecting the digital dots: Literacy of the 21st century. *Educause Quarterly*, *29*(2), 8-10.

F

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 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.[14]

H

Hacking.

Appropriately applying ingenuity[15]

Cleverly solving a programming problem[16]

Using a computer to gain unauthorized access to data within a system

Hardware. The physical components that make up a computer.[2]

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.[17]

HTML. HyperText Mark-up Language; the language used to create web pages.[2]

Hub. In a computer **network**, a device to which all other devices are wired.[18]

I

Index. A common method for keeping track of data so that it can be accessed quickly. Like an index in a book, it is a list in which each entry contains the name of the item and its location. However, computer-based indexes may point to a physical location on a disk or to a logical location that points elsewhere to the actual location.[19]

[14] Knuth, D. E. (2011). *Art of Computer Programming*. Addison-Wesley.

[15] Kidder, T. (2011). *The soul of a new machine*. Back Bay Books.

[16] SpeedGuide.net, accessed April 2019: https://www.speedguide.net/glossary.php?seek=hack

[17] Google.com, accessed April 2019: https://www.google.com/search?q=define+hexadecimal&ie=&oe=

[18] PCMag.com, accessed April 2019: https://www.pcmag.com/encyclopedia

[19] PCMag.com, accessed April 2019: https://www.pcmag.com/encyclopedia

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**).[20][21]

Input (noun). A data value passed from the outside world to a computer.[2]

Input (verb). To send data from the outside world into a computer system.[2]

Internet. A network of interconnected **networks**.[2]

Intellectual property. Something (such as an idea, invention, or process) that comes from a person's mind.[22]

Internet Protocol (IP) Address. A unique numeric value assigned to a computer or other device connected to the **Internet** so that it may be identified and located.23 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.[4]

L

Latency. The delay before a transfer of data begins following an instruction for its transfer.[3]

Library/code library. A collection of **programs**, **applications**, or resource files, the goal of which is to provide students with sample **applications** and supplemental information to help them create or customize their own 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.[24]

Lists. A **data structure** for storing ordered values.[2] Data are of arbitrary/unfixed size.

Local Area Network (LAN). A computer **network** limited to a small area, such as an office building, university, or even a residential home.[3]

Loop. A sequence of **statements** which is specified once, but which may be carried out several times in succession.[25]

[20] Newsom, D., & Haynes, J. (2007). *Public relations writing: Form & style*. Cengage Learning.

[21] Tufte, E. R. (1983). *The visual display of quantitative information*. Graphics Press. Cheshire, Connecticut.

[22] Merrian-Webster, accessed April 2019: https://www.merriam-webster.com/

[23] Postel, J. (1980). *RFC 760-DoD standard internet protocol*. Information Sciences Institute of the University of Southern California.

[24] Commission of the European Communities. (2001). *Making a European area of lifelong learning a reality*. Commission of the European Communities.

[25] Watt, D. A. (2004). *Programming language design concepts*. John Wiley & Sons.

M

Machine code/machine language. a computer programming language consisting of **binary** or **hexadecimal** instructions which a computer can respond to directly.[33]

Media Access Control (MAC) Address. 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.[2]

Modeling (verb). The act of creating a **model**.[2]

N

Network. A collection of **nodes** connected to one another by **networking devices** and links (cables or by wireless media) and arranged so data may be sent between devices either directly or via other devices.[2]

Networking devices. Network devices are units that mediate data in a **network**.[2] Networking hardware such as **hubs**, **switches**, **routers**, and **bridges** are used to connect **nodes** on a network, so that they can share data or resources.

Node. Computational devices (e.g., personal computers, printer, smart phones, servers) on a **network**.

O

Object code. The **machine language** representation of programming **source code**. Software called a "compiler" converts the source code into object code and links it to other object code **libraries** to become an executable program. Not related to **object-oriented programming**.[18]

Object-oriented programming. Computer programming based on the concept of “objects”, which can contain data, in the form of fields (often known as attributes), and code, in the form of procedures (often known as methods).[26]

[26] Kindler, E.; Krivy, I. (2011). Object-Oriented Simulation of systems with sophisticated control. *International Journal of General Systems*: 313–343.

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.[27]

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**.[27]

Operating System (OS). A set of programs that manage the functioning of, and other programs' access to, hardware.[2]

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.[28]

Output (noun). A response from a system.[2] Example: a program that adds could have inputs of 2 and 2 with an output of 4.

Output (verb). To generate an **output**.[2]

P

Parameter. 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.[4]

Peripheral (device). A peripheral device 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.[29]

Pixel. The smallest controllable element of a picture/display.[2]

Pop-up. Appearing suddenly on a computer screen.[22]

Procedure. Lists of programs to be executed. In programming, a procedure is another term for a **module**, **subroutine** or **function**. 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.[19]

[27] Smith, M. S., Moteff, J. D., Kruger, L. G., McLoughlin, G. J., & Seifert, J. W. (2001). *Internet: an overview of key technology policy issues affecting its use and growth* (pp. 41-72). Nova Science Publishers, Inc.

[28] WhatIs.com, accessed May 2015: http://whatis.techtarget.com/definition/operator

[29] Laplante, P. A. (Ed.). (2000). *Dictionary of computer science, engineering and technology*. CRC Press.

Process (noun). A running program.[2]

Process (verb). The act of using data to perform a calculation or other operation.[2]

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.

Program. A set of instructions a computer executes in order to achieve a particular objective.[2]

Programming (computer programming). The craft of analyzing problems and designing, writing, testing, and maintaining **programs** to solve them.[2]

Programming language. Formal language used to give a computer instructions.[2]

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.[30]

Proprietary software. **Software** distributed in **object code** form. The developers or distributors reserve all freedoms and rights.[27]

Pseudocode. An informal high-level description of the operating principle of a computer program or other **algorithm**.

R

Recursive. A **procedure** or **subroutine**, implemented in a programming language, whose implementation references itself.[31]

Repetition. The process of repeating a task a set number of times or until a **condition** is met.[2]

Resolution. A measurement of the number of **pixels** needed to display an image.[2]

Router. A device that connects **networks** to one another.

S

Safety. The awareness of personal, physical, and psychological well-being in a digital society.

Selection. Using **conditions** to control the flow of a program.[2]

Sequence (noun). An ordered set of instructions.[2]

[30] Sahoo, R. & Sahool, G. (2016). *Computer science with C++*. New Saraswati House (India) Pvt. Ltd.

[31] Graham, R. L., Knuth, D. E., & Patashnik, O. (1990). *Concrete mathematics, 7th printing*.

Sequence (verb). To arrange instructions in a particular order.[2]

Server. A computer or **program** dedicated to a particular set of tasks that provides **services** to other computers or programs on a **network**.[2]

Services. **Software** and **hardware** that provide some capability that can be accessed by another program or device remotely or through a defined, discoverable interface.

Sexting. The act of sending, receiving, or forwarding sexually explicit messages, photos, or images via cell phone, computer, or other digital device.[32]

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.[2]

Software piracy. Illegal copying, distribution, or use of **software**.

Source code. Programming statements and instructions written by a computer programmer. Source code is what a programmer writes, but it is not directly executable by the computer.[33]

Spam. Unsolicited commercial advertisements distributed online. Most spam comes to people via e-mail, but spam can be found in online chat rooms and message boards.[22]

Stateful. Capable of maintaining the status of a process or transaction.

Statement. A descriptive phrase that generates one or more instructions in a computer program.[22]

String. A set of adjoining alphanumeric characters. Strings are text, such as names, addresses and descriptions. Although a string may include numeric digits, the digits cannot be calculated within the string. They have to be copied out of the string into a numeric structure.[19]

Subroutine. See **procedure**.

Switch. A device that connects devices on a computer **network** by receiving, processing, and forwarding data to the destination device.[34]

Tag. A field that identifies the contents of a data record.[19]

[32] Mass.Gov, accessed May 2016: http://www.mass.gov/berkshireda/crime-awareness-andprevention/sexting/sexting.html

[33] Google.com, accessed April 2019: https://www.google.com/search?q=machine+language

[34] Kohlhepp, R. J. (2000). The 10 most important products of the decade. *Network Computing*.

V

Variable. In computer programming, a storage location paired with an associated symbolic name (an identifier), which contains some known or unknown quantity of data referred to as a "value." The variable name is the usual way to reference the stored value.[35]

[35] Aho, A. V., Sethi, R., & Ullman, J. D. (1986). *Compilers: Principles, techniques, and tools*. Addison-Wesley.