DIGITAL LITERACY AND COMPUTER SCIENCE CURRICULUM FRAMEWORK

Developed 2018-2019
Implemented 2019-2020

Mountain Brook Schools
32 Vine Street
Mountain Brook, AL 35213

DIGITAL LITERACY AND COMPUTER SCIENCE CURRICULUM FRAMEWORK

Mountain Brook Schools
Board of Education

Elizabeth Dunn, Chairman
Brad Sklar, Vice Chairman
Tommy Luckie
Richard Murray
Nicky Barnes

Dr. Richard Barlow
Superintendent

Dr. Missy Brooks
Director of Instruction

Digital Literacy and Computer Science
Curriculum Committee

Brookwood Forest Elementary
Sarah Katherine Nichols – K
Jennifer Jinnette - K
Tanishia Sims – K
Tara Smith – K
Perry Wright – K
Sammye Davis – 1
Jennifer Hunt – 1
Stacey Kirkpatrick – 1
Carrie Knight – 1
Travis Morgan – 1
Cindy Burns – 2
Caroline Peek – 2
Ashley Scott – 2
Eleanor Walker – 2
Dawn Elsberry – 3
Linda Mason – 3
Caroline Pridgen – 3
Kelly Stout – 3
Natalie Borland – 4
Ann Scott Cohen – 4
Laura Frenz – 4
Lane Walker – 4
Carla Dudley – 5
Jolie Welner – 5
Audrey Williams – 5
John Woolard – 5
Marion Bray – 6
Tracy Cole – 6
Jessica Meriwether – 6
Virginia Moore – 6
Kathy Snyder – 6
Sarah Katherine Nichols - Parent
Sharon Mumm – Technology Coordinator
Christy Christian – Assistant Principal

Cherokee Bend Elementary
Heather Brown – K
Cynthia Echols – K
Leah Saab – K
Hannah Umphrey – K
Suzy Brown – 1
Roby Gaut – 1
Trish Humphries – 1
Bethny Lee – 1
Kelly Anderson – 2
Kelley Finley – 2
Emily Griner – 2
Meredith Lusco – 2
Samantha Austin – 3
Danean Davis – 3
Maggie Helms – 3
Lyndsi Kirk – 3
Beth Dean – 4
Shelley Hunt – 4
Sally Till – 4
Karan Bush – 5
Anna DeBell – 5
Stacy Freeman – 5
Barbara Parker – 5
Alexandra Andrews – 6
Laurie Fuentes – 6
Lane Tucker – 6
Claire Puckett - Parent
Kenneth Camp – Technology Coordinator
Blair Inabinet – Assistant Principal

Crestline Elementary
Jenny Azar – K
Jayne Euwer – K
Phyllis Farrar – K
Melanie Hennessy – K
Johnna Noles – K
Sarah Norris – K
Michele Ramsey – K
Rachel Anderson – 1
Greer Black – 1
Marlyss Green – 1
Deborah Holder – 1
Chelsey Summerrow – 1
Lindsey Westlake – 1
Tracey Barringer – 2
Kendra Bierbrauer – 2
Allison Davis – 2
Kay Haley – 2
Christy Neely – 2
Sarah Stradley – 2
Susan Weston – 2
Tara Davis – 3
Kelsey Long – 3
Kelly Mitchell – 3
Carly Morgan – 3

Becca Pigg – 3
Laura Rives – 3
Sally Baker – 4
Caroline Ferrarone – 4
Scott McKerley – 4
Bradley O'Neill – 4
Jennifer Preston – 4
Jenny Wood – 4
Sarah Neale Bearden – 5
Cindy Carlisle – 5
Bill Garner – 5
Melinda Howe – 5
Amanda Johnson – 5
Lori Sullivan – 5
Lawson Hollans – 6
Teresa Howell – 6
Karen Scott – 6
Elizabeth Studinka – 6
Jamie Underwood – 6
Jackie Woodall – Parent
Paula Stanbridge – Technology Coordinator
Catherine Waters – Assistant Principal
Josh Watkins – Assistant Principal

Moutain Brook Elementary
Kelsey Frey – K
Mitchel Nelson – K
Katie Potts – K
Julie Summers – K
Joy Bohringer – 1
Julie Cox – 1
Connie Liddle – 1
Paige Ward – 1
Katherine Brown – 2
Katherine Cochran - 2
Bethany McCandless – 2
Julie Tuck – 2
Judy Dyess – 3
Kim Hall – 3
Paulina Haskins – 3
Cindy Peavy – 3
Meredith Collins – 4
Ashley Margaritis – 4
Loretta Rowan – 4
Jennifer Wilson – 4
Bill Andrews – 5
Suzanne Andrews – 5
Ruthie Gravlee – 5
Alex McCain – 6
Lauren Merrill – 6
Bethany Tompkins – 6
Missy Wright – 6
Suzanne Perkins - Parent
Thea Patrick – Technology Coordinator
Brannon Aaron – Assistant Principal

Mountain Brook Jr. High
Ruth Beenken - English
Andrew Cotton - English
Anne Carter Finch - English
Julie Garret - English
Andrew Grayson - English
Trip Hubbard - English
Anna McCain - English
Mary Phillips - English
Pam Pugh - English
Theresa Shadrix - English
Leslie Stephenson - English
Kimberly Wilder - English
Renee Collingwood - Math
Mary Riley Deer - Math
Nidia Fernandez-Lee - Math
Denise Grisham - Math
Brittany Henegar - Math
Madeline Hunt - Math
Drew Jackson - Math
Adam Johnson - Math
Cathy Laswell - Math
Lars Porter - Math
Wendy Spiller - Math
Priscilla Stokes - Math
Brittany Wilson – Math
Mariya Breaux - Science
Marisa Burns - Science
Susan Haggard - Science
Bruce Henricks - Science
Daniel Sipes - Science
Pearle Smith - Science
Debbie Stump - Science
Vicki Webb - Science
Courtney Burger - Social Studies
Leslie Carlson - Social Studies
D. C. Hall - Social Studies
Paul Hnizdil - Social Studies
Derek Kennedy - Social Studies
Lisa Lewis - Social Studies
Beth Lippeatt - Social Studies
John Pledger - Social Studies
Ben Smith - Social Studies

Tami Genry - Librarian
Suzan Brandt – Technology Coordinator
Brook Gibbons – Assistant Principal
Holly Martin – Staff Development Specialist

Mountain Brook High School
Krissie Allen - English
Melinda Cammarata - English
Nancy Glaub - English
Julie Kash - English
Angela Knox - English
Catherine Lowe - English
Jane Major - English
Shannon Marks - English
Christina McGovern - English
Mattie Newson - English
Jeff Roberts - English
Denise Trimm - English
Summer Upton - English
Greg Wald - English
Wanda Burns - Math
Morgan Chatham - Math
Jacqueline Cotter - Math
Amy Kathryn Gannon - Math
Rhonda Guillory - Math
Nancy Hollis - Math
Paul Kustos - Math
Fred Major - Math
Kristina Noto - Math
Christy Stamps - Math
Casey Truesdale - Math
Lauren Wright – Math
Rhonda Aust - Science
Christi Elias - Science
Lynn Faulk - Science
Toula Froemelt - Science
Barry Hartley - Science
Michelle Holdbrooks - Science
Marcy Jordan - Science
Michael McGovern - Science
Melissa Scott Palmer - Science
Bryan Rosenstiel - Science
Walt Rogers - Science
Ashley Van Beek - Science
Ginny Bakken - Social Studies
Ben Callaway - Social Studies
Jake Collins - Social Studies
Missy Cunningham - Social Studies
Matt Ferguson - Social Studies
Pete Giangrosso - Social Studies
Leah Kilfoyle - Social Studies
Glenn Lamar - Social Studies
Shane Martin - Social Studies
Brock Rotter - Social Studies
Alessia Sartorio - Social Studies
Sherri Traffica - Social Studies
Joe Webb - Social Studies
Holly Alexander – World Languages
John Binet – World Languages
Jessie Creech – World Languages
Heather Fitch – World Languages
Audrey Laird – World Languages
Drew Lasater – World Languages
Lori Leopard – World Languages
Allison Price – World Languages
Crawford Bumgarner – Parent
Joani Kay – Technology Coordinator
Carrie Busby – Assistant Principal

Central Office
Dr. Missy Brooks – Director of Instruction
Lanie Kent – Assistant Director of Instruction
Donna Williamson – Director of Technology

Preface

The *Mountain Brook Digital Literacy and Computer Science Curriculum Framework* was developed using the *Alabama Course of Study: Digital Literacy and Computer Science*. Content standards in this document define minimum requirements, in accordance with provisions of the Code of Alabama (1975, §16-35-4). The standards are fundamental and specific but not exhaustive. This document provides an overview and learning goals for each grade band and outlines minimum standards for each grade.

Alabama Course of Study:
Digital Literacy and Computer Science
General Introduction

Technology allows educators and students to transform teaching and learning and to develop crucial skills for communicating, creating, and interacting with each other in a global society. Although technology is not a panacea for all instructional problems, it equips students with tools that have not existed in the past. Technology allows digitally and computationally literate students to transition from being simply consumers of information and media to being producers as well.

Attaining digital and computational literacy strengthens life skills such as solving problems creatively, thinking critically, and working cooperatively in teams. Because technology is at the center of almost every aspect of daily life, the digitally literate person is more likely to face the challenges of a dynamic global society with confidence.

Digitally literate students can use technology responsibly and appropriately to create, collaborate, think critically, and apply algorithmic processes. They can access and evaluate information to gain lifelong knowledge and skills in all subject areas.

The *Mountain Brook Digital Literacy and Computer Science (DLCS) Curriculum Framework* defines the minimum required content that students should know and be able to do in order to learn effectively and become capable, responsible, and self-reliant citizens in this information-based global society. Content standards in this document are minimum and required, as specified in the Code of Alabama (1975), §16-35-4. They are fundamental but not exhaustive.

This plan draws upon the requirements of nationally recognized programs. The International Society for Technology in Education (ISTE) Standards for Students emphasize the skills and qualities we want to foster in students, enabling them to engage and thrive in a connected, digital world. The Course of Study standards are designed for use by educators across the curriculum at every grade level, so that these skills are cultivated throughout a student's academic career (2016 ISTE Standards for Students). The K-12 Computer Science Framework illuminates the big ideas of computer science through the lenses of concepts (what students should know) and practices (what students should do), representing the behaviors that computationally literate students use to engage with the core concepts of computer science.

The DLCS standards will enable students to employ cognitive and technical skills to find, evaluate, create, and communicate information via existing and emerging technologies. The standards introduce the study of computers and algorithmic processes, including computer science principles, hardware and software designs, applications, networks, and societal impact, and lay the groundwork for students to use their increasingly valuable knowledge and skills in college and careers.

Students will use digital tools to create, communicate, and collaborate. These tools provide powerful, engaging learning experiences which pervade their daily lives and impact the future. Technological understanding prepares students to be productive citizens.

Digital Literacy and Computer Science Conceptual Framework

Conceptual Framework

The Conceptual Framework graphic exemplifies the purpose of the *Mountain Brook Digital Literacy and Computer Science Curriculum Framework,* which is to enhance students' lives by providing them the knowledge and skills to be innovators and positive contributors to the society in which they live. An Alabama student, a citizen of the world, is depicted as the epicenter from which six strands radiate around the globe. The student's heart is a prominent feature because communication, collaboration, creativity, and critical thinking all require empathy. Empathy begins with understanding the human condition and opening the mind to new perspectives and ideas. Without understanding and openness, progress cannot be made.

Technology has the potential to amplify students' capacity to collaborate, create, and communicate in an increasingly global economy. In order to improve the world, one must understand how technology shapes the landscape and reshapes our institutions at an ever-increasing speed. To employ and produce new technologies, a global citizen not only needs to be proficient in the use of digital tools but must also understand how and why these tools work. Global citizens must utilize technological tools, algorithmic thinking, and digital strategies as means to acquire knowledge, to communicate and collaborate locally and globally, to identify and solve complex problems, and to share solutions and ideas with the world.

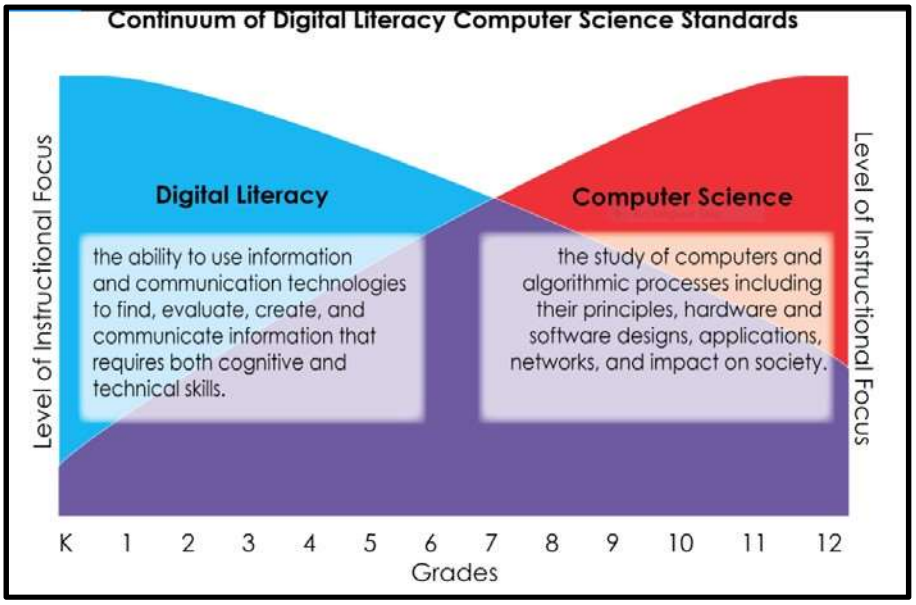
The conceptual framework graphic succinctly summarizes the structure and goals of digital literacy and computer science education in Alabama. The strands emerging from the student to encircle the globe represent digital connection to the world and specify the roles filled by students of today and tomorrow: **Computational Thinker, Citizen of the Digital Culture, Global Collaborator, Computing Analyst,** and **Innovative Designer.** These titles indicate that digital citizens should not merely connect, but responsibly work together to improve the world. The careful observer notices a sixth ribbon, currently unlabeled to indicate that new and emerging technologies will require openness to future changes.

In the background, underpinning the strands, are two elements that are key to their implementation. The map of Alabama is depicted by a circuit board, which represents tangible hardware. The continents on the globe are marked with binary code, the language of software and computer science. Students will not only interact with both of these on a daily basis but also take part in their construction in order to become the innovative citizens the world needs today and tomorrow.

The goal of the Digital Literacy and Computer Science standards is to enable students to use cognitive and technical skills responsibly in finding, evaluating, creating, and communicating information. Standards will also introduce students to the study of computers and algorithmic processes, including computer science principles, hardware and software design, applications, networks, and societal impacts, so that students will be fully equipped with the important, increasingly valuable knowledge and skills needed in college and careers.

Position Statements

A Vision for K-12 Computer Science:



In the early grades, the continuum focuses more on digital literacy, the skills that students must learn with the introduction of computer science standards. In the later grades, the instructional focus transitions toward computer science while continuing to address more advanced digital literacy skills. While both focus areas are present along the entire continuum, this graph represents the transition in the level of instructional focus as students progress along the continuum.

Digital Literacy

A digitally literate student is able to work with digital tools both alone and in networked environments. Students must also have the skills to adapt to new tools throughout their lifetimes as resources and platforms continue to evolve. The operating systems, interfaces, resources, and collaborative technology of today require students to advance with the latest innovations in collaboration and creation as new systems appear.

Computer Science

Computing is essential for today's students to possess the computational thinking skills required for the workforce both now and in the future. A computational mindset helps students engage the digital world in which they live. An understanding of Internet protocols, data representation, and solution-based and algorithmic processes allows students to meet the challenges of computational thinking confidently.

Global Collaboration

Students need opportunities to connect with others locally and globally, giving each the opportunity to learn together, share knowledge, and develop cultural understandings and relationships. Technology is the conduit that provides easy-to-implement experiences and opportunities for teaching and learning. It is imperative that students be provided with opportunities to exercise these skills in an authentic environment without respect to physical boundaries.

Assessment

Students must be digitally competent if they are to be successful in academic, professional, and personal arenas. Their competence must be effectively assessed to guide classroom planning. Digital literacy and computer science cannot be adequately measured using traditional, paper-and-pencil objective tests. These skills are best assessed through problem- and/or project-based assignments, preferably as content-embedded tasks that solve authentic problems.

Directions for Interpreting the Minimum Required Content

1. **Content Standards** are statements that define what students should know and be able to do at the conclusion of a course or grade. Content standards in this document contain minimum required content. The order in which standards are listed within a course or grade is not intended to convey
a sequence for instruction. Each content standard completes the phrase "*Students can*."

> *Students can:*
>
> Create a research-based product collaboratively using online digital tools.
>
> (Grade 1 – Content Standard 13)

2. **Lettered Subtitles** denote content that is related to the standards and required for instruction. Subtitles are listed under standards and identify additional minimum required content.

> *Students can:*
>
> Differentiate between a generalized expression of an algorithm in pseudocode and its concrete implementation in a programming language.
> a. Explain that some algorithms do not lead to exact solutions in a reasonable amount of time and thus approximations are acceptable.
>
> (Grades 9 – 12 – Content Standard 3)

3. **Examples** clarify certain components of content standards and are not required content. They are illustrative but not exhaustive.

> *Students can:*
>
> Explain social engineering, including countermeasures, and its impact on a digital society.
> Examples: Phishing, hoaxes, impersonation, baiting, spoofing.
>
> (Grade 7 – Content Standard 12)

Recurring Standards for the Mountain Brook Digital Literacy and Computer Science Curriculum Framework

Recurring standards are key practices or concepts that recur at grade levels along the K - 12 continuum with progressive complexity. Rather than repeating these standards at multiple grade levels in this document, the standards are outlined below.

Safety, Privacy, and Security

1. Identify, demonstrate, and apply personal safety use of digital devices.

Legal and Ethical Behavior

2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules.

Impact of Computing

3. Analyze the potential impact of computing.

Systems

4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.

Collaborative Research

5. Locate, curate, and evaluate information from digital sources to answer research questions.

Digital Tools

6. Produce, review, and revise authentic artifacts using appropriate digital tools.

Grades K-2 Overview

In the primary grades (Grades K-2), students begin their formal study of digital literacy and computer science skills. As they are introduced to the digital world, students explore concepts by integrating basic digital literacy skills with simple ideas about computational thinking. At this level, the focus is on learning with digital tools, enhancing the process and student outcomes. Students begin to choose the best tool to meet a need or solve a problem. They discover ways to think and to use digital tools to complete tasks more easily, collaboratively, and efficiently.

Students in kindergarten through second grade will meet the following learning goals:

- As ***Computational Thinkers,*** students explain how computing is an integral part of our world.
- As ***Citizens of a Digital Culture,*** students demonstrate ways to be good digital citizens.
- As ***Global Collaborators,*** students collaborate with other learners and contribute ideas to their joint projects.
- As ***Computing Analysts,*** students use their growing knowledge of computers to create artifacts systematically and efficiently.
- As ***Innovative Designers,*** students undertake challenges and create new ways to address existing problems.

By the end of second grade, students understand the importance of perseverance as they create plans, collect data, and analyze data to make informed decisions.

Kindergarten Overview

Kindergarten content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade.

Content Standard Strands and Topics	Recurring Standards
Computational Thinker Abstraction Algorithms Programming and Development **Citizen of a Digital Culture** Safety, Privacy, and Security Legal and Ethical Behavior Digital Identity Impact of Computing **Global Collaborator** Communication Digital Tools Collaborative Research **Computing Analyst** Data Systems **Innovative Designer** Human/Computer Partnerships Design Thinking	**Safety, Privacy, and Security** 1. Identify, demonstrate, and apply personal safe use of digital devices. **Legal and Ethical Behavior** 2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules. **Impact of Computing** 3. Assess the validity and identify the purpose of digital content. **Systems** 4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues. **Collaborative Research** 5. Locate and curate information from digital sources to answer research questions. **Digital Tools** 6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Kindergarten

Students in kindergarten explore ways they relate to their world and to digital environments. They start to learn that certain information should be confidential. As a class, students begin to collaborate beyond the walls of their classroom by learning from others, exploring new ideas, collecting data, and analyzing data to make decisions. Kindergartners learn to use digital tools to express ideas, complete tasks, solve problems, and begin to comprehend how technology can help them understand and relate to others. Underlined words appear in the glossary.

Students can:

Computational Thinker

Algorithms

1. List the sequence of events required to solve problems.
   Examples: Tying shoes, making a sandwich, brushing teeth.

Programming and Development

2. Demonstrate use of input devices.
   Examples: Mouse, touch screen, keyboard.

Citizen of a Digital Culture

Safety, Privacy, and Security

3. Distinguish between private and public information.
   Example: Your birth date is private; your shirt color is public.
4. Identify age-appropriate methods for keeping personal information private.
   Example: Keeping passwords, name, address, and phone number confidential.

Legal and Ethical Behavior

5. Demonstrate appropriate behaviors for working with others responsibly and kindly.
   Examples: Face-to-face collaborative groups or interactions, online interactions, role play.

Impact of Computing

6. Recognize ways in which computing devices make certain tasks easier.
   Examples: Communication, doctor's visits/medical records, maps and directions.

Global Collaborator

Digital Tools

7. Locate letters and numbers on the keyboard.

Collaborative Research

8. Present information from a variety of digital resources.
9. Create a research-based product collaboratively using online digital tools, given specific guidance.
   Examples: Find simple facts about a specific topic, create a slide that contains facts located in trade books or other sources as a group or with a partner.

Computing Analyst

Data

10. Collect data and organize it in a chart or graph collaboratively.
11. Describe how digital devices save information.

Systems

12. Use a variety of digital devices, in both independent and collaborative settings.
    Examples: Interactive boards, tablets, laptops, other handheld devices.

Innovative Designer

Design Thinking

13. Use a design process in a guided setting to create an artifact or solve a problem.
    Example: Problem - understanding locations on the school campus. Solution - draw paper or digital maps of the school.

Grade 1 Overview

Grade 1 content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade level.

Content Standard Strands and Topics

Computational Thinker

- Abstraction
- Algorithms
- Programming and Development

Citizen of a Digital Culture

- Safety, Privacy, and Security
- Legal and Ethical Behavior
- Digital Identity
- Impact of Computing

Global Collaborator

- Communication
- Digital Tools
- Collaborative Research

Computing Analyst

- Data
- Systems

Innovative Designer

- Human/Computer Partnerships
- Design Thinking

Recurring Standards

Safety, Privacy, and Security
1. Identify, demonstrate, and apply personal safe use of digital devices.

Legal and Ethical Behavior
2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules.

Impact of Computing
3. Assess the validity and identify the purpose of digital content.

Systems
4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.

Collaborative Research
5. Locate and curate information from digital sources to answer research questions.

Digital Tools
6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Grade 1

Students in first grade describe and utilize the basic functions of computing devices. They begin to create algorithms collaboratively and start learning keyboarding skills. First graders explore and identify the appropriateness of specific online behaviors. As a class, students communicate and collaborate with people outside their immediate environment to understand how others use technology in their daily lives. Students use digital tools to demonstrate their knowledge to others and use feedback to solve problems.

Underlined words appear in the glossary.

Students can:

Computational Thinker

Abstraction

1. Classify and sort information into logical order with and without a computer.
   Examples: Sort by shape, color, or other attribute; sort A-Z.

Algorithms

2. Order events into a logical sequence or algorithm.
   Examples: Unplugged coding activities, sequence of instruction.

Programming and Development

3. Construct elements of a simple computer program in collaboration with others.
   Examples: Block programming, basic robotics, unplugged programming.

Citizen of a Digital Culture

Safety, Privacy, and Security

4. Demonstrate age-appropriate methods for keeping personal information private.
   Example: Keep passwords confidential, use anonymous profile picture or avatar, develop user names that are non-identifying or do not include actual name.

Legal and Ethical Behavior

5. Differentiate between prior knowledge and ideas or thoughts gained from others.
6. Identify appropriate and inappropriate behaviors for communicating in a digital environment.
   Examples: Cyberbullying, online etiquette.

Digital Identity

7. Recognize that a person has a digital identity.

Impact of Computing

8. Identify ways in which computing devices have impacted people's lives.
   Example: Location services, instantaneous access to information.

Global Collaborator

Communication

9. Use a variety of digital tools collaboratively to connect with other learners.
   Examples: Video calling, blogs, collaborative documents.

Digital Tools

10. Identify an appropriate tool to complete a task when given guidance and support.
    Examples: Choosing a word processing tool to write a story, choosing a spreadsheet for a budget.
11. Type five words per minute minimum with 95% accuracy using appropriate keyboarding techniques.

Collaborative Research

12. Identify keywords in a search and discuss how they may be used to gather information.
13. Create a research-based product collaboratively using online digital tools.
    Examples: Find simple facts about a specific topic, create a slide that contains facts located in trade books or other sources

Computing Analyst

Data

14. Discuss the purpose of collecting and organizing data.
15. Interpret data displayed in a chart.
    Example: Using charts which depict data students interpret the data either verbally or in written form (which has more, less, are equal).
16. Demonstrate how digital devices can save information as data that can be stored, searched, retrieved, and deleted.

Systems

17. Use digital devices with a variety of operating systems.
    Examples: Interactive boards, tablets, laptops, other handheld devices
18. Label visible components of digital devices.
    Examples: Visible input and output components such as USB, touch screen, keyboard, audio and video connectors, speakers.

Innovative Designer

Design Thinking

19. Identify and revise problem-solving strategies to solve a simple problem.
    Examples: Scientific method, visual images or mind pictures, look for patterns, systematic list.

Grade 2 Overview

Grade 2 content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade level.

Content Standard Strands and Topics

Computational Thinker

- Abstraction
- Algorithms
- Programming and Development

Citizen of a Digital Culture

- Safety, Privacy, and Security
- Legal and Ethical Behavior
- Digital Identity
- Impact of Computing

Global Collaborator

- Communication
- Digital Tools
- Collaborative Research

Computing Analyst

- Data
- Systems

Innovative Designer

- Human/Computer Partnerships
- Design Thinking

Recurring Standards

Safety, Privacy, and Security
1. Identify, demonstrate, and apply personal safe use of digital devices.

Legal and Ethical Behavior
2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules.

Impact of Computing
3. Assess the validity and identify the purpose of digital content.

Systems
4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.

Collaborative Research
5. Locate and curate information from digital sources to answer research questions.

Digital Tools
6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Grade 2

Students in second grade take proper care of computing devices and use them responsibly, gaining benefits from various digital tools as they find ways to use them in their daily tasks. Students research meaningful topics using appropriate sources and acknowledge their sources properly. Students exchange information through various media and present their ideas to diverse audiences. Second graders demonstrate their knowledge of computational thinking by creating multi-step algorithms to solve problems.

Underlined words appear in the glossary.

Students can:

Computational Thinker

Abstraction

1. Create and sort information into useful order using digital tools.
   Examples: Sort data spreadsheets A-Z, simple filters, and tables.

Algorithms

2. Create an algorithm for other learners to follow.
   Examples: Unplugged coding activities, illustrate sequence of a process such as baking a cake.

Programming and Development

3. Construct elements of a simple computer program using basic commands.
   Examples: Digital block-based programming, basic robotics.
4. Identify bugs in basic programming.
   Examples: Problem-solving, trial and error.

Citizen of a Digital Culture

Legal and Ethical Behavior

5. Cite media and/or owners of digital content at an age-appropriate level.
   Example: Basic website citation.
6. Demonstrate appropriate behaviors for communicating in a digital environment.
   Example: netiquette.

Digital Identity

7. List positive and negative impacts of digital communication.
   Example: Anything posted or communicated electronically may be easily reproduced and could remain a positive or negative part of your digital identity/footprint.

Impact of Computing

8. Interpret ways in which computing devices have influenced people's lives.
   Example: Discuss tasks completed daily in which some type of device is used to make the tasks easier (calculator, microwave to quickly heat food, mobile phone for instant communication).

Global Collaborator

Communication

9. Use a variety of digital tools to connect with other learners.
   Examples: Online conferences, blogs, collaborative documents.

Digital Tools

10. Identify multiple tools which could be used to complete a task.
11. Type 10 words per minute with 95% accuracy using appropriate keyboarding techniques.

Collaborative Research

12. Conduct basic keyword searches to gather information.
13. Create a research-based product using online digital tools.

Computing Analyst

Data

14. Collect, create, and organize data in a digital chart or graph.
15. Explain how users control the ways digital devices save information in an organized manner.
    Examples: Folders, cloud-based, pictures, chronologically, naming files.

Systems

16. Compare the different operating systems used on digital devices.
17. Explain the purposes of visible input and output components of digital devices.
    Examples: Purpose of keyboard, mouse, ports, printers, etc.

Innovative Designer

Design Thinking

18. Investigate the design process and use digital tools to illustrate potential solutions to a problem, given guidance and support.
    Examples: Create a presentation, drawing or graphic, audio tool, or video.

Grades 3-5 Overview

In Grades 3-5, students explore diverse computing devices and digital tools while developing their problem-solving and computational thinking skills. These skills are necessary across the curriculum. Third- through fifth- grade students are able to engage in learning in ways that are methodical and imaginative. Students' capabilities as problem solvers, innovators, and creators build on their K–2 experiences.

Students in third, fourth, and fifth grades will meet the following learning goals:

- As ***Computational Thinkers,*** students use problem-solving processes to understand how to write and debug an algorithm and to evaluate and create new informational representations which successfully reframe an issue.
- As ***Citizens of a Digital Culture,*** students demonstrate an understanding of concepts involving safety and security, responsible use of technology, and the influence of technology on its users.
- As ***Global Collaborators,*** students collaboratively utilize intermediate research skills to create artifacts and use digital tools to communicate or exchange information.
- As ***Computing Analysts,*** students understand and use various computing devices strategically to solve a problem and accomplish a task in the most effective way.
- As ***Innovative Designers,*** students pioneer new solutions, products, and processes through design thinking and be familiar with the advantages and limitations of technology.

When these learning goals are mastered in a student-centered environment, students will become proficient global citizens who are able to deal with a rapidly changing world. Alabama's students will be able to solve both intermediate and complex problems and find desirable solutions for both local and global issues. The design thinking process will allow students to use logic, intuition, imagination, and systematic reasoning to explore what could be and create innovative solutions that benefit themselves and others.

Grade 3 Overview

Grade 3 content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade level.

Content Standard Strands and Topics

Computational Thinker

- Abstraction
- Algorithms
- Programming and Development

Citizen of a Digital Culture

- Safety, Privacy, and Security
- Legal and Ethical Behavior
- Digital Identity
- Impact of Computing

Global Collaborator

- Communication
- Digital Tools
- Collaborative Research

Computing Analyst

- Data
- Systems

Innovative Designer

- Human/Computer Partnerships
- Design Thinking

Recurring Standards

Safety, Privacy, and Security
1. Identify, demonstrate, and apply personal safe use of digital devices.

Legal and Ethical Behavior
2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules.

Impact of Computing
3. Assess the validity and identify the purpose of digital content.

Systems
4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.

Collaborative Research
5. Locate and curate information from digital sources to answer research questions.

Digital Tools
6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Grade 3

In third grade, students build on K-2 foundations by looking at basic troubleshooting and whole-class problem-solving. Students will identify appropriate uses of technology and a broad range of computing systems. Third grade standards focus on student collaboration and communication.

Underlined words appear in the glossary.

Students can:

Computational Thinker

Abstraction

1. Use numbers or letters to represent information in another form.
   Examples: Secret codes/encryption, Roman numerals, or abbreviations.
2. Analyze a given list of sub-problems while addressing a larger problem.
   Example: Problem - making a peanut butter sandwich; sub-problem - opening jar, finding a knife, getting the bread.
   Problem - design and share a brochure; sub-problem - selecting font, choosing layout.

Algorithms

3. Explain that different solutions exist for the same problem or sub-problem.
   Example: Multiple paths exist to get home from school; one may be a shorter distance while one may encounter less traffic.
4. Examine logical reasoning to predict outcomes of an algorithm.
5. Create an algorithm to solve a problem as a collaborative team.
   Examples: Move a character/robot/person through a maze. List steps to build a sandwich.
6. Describe the function of a flowchart.

Programming and Development

7. Test and debug a given program in a block-based visual programming environment using arithmetic operators, conditionals, and repetition in programs, in collaboration with others.
   Examples: Sequencing cards for unplugged activities, online coding practice.

Citizen of a Digital Culture

Safety, Privacy, and Security

8. Describe how to use proper ergonomics when using devices.
   Examples: Body position, lighting, positioning of equipment, taking breaks.
9. Identify the proper use and operation of security technologies.
   Examples: Passwords, virus protection software, spam filters, pop-up blockers.
10. Describe ways web advertising collects personal information.
    Examples: Search ads, banner ads, in-game ads, email ads.

Impact of Computing

11. Identify resources in the community that offer technology access.
    Examples: Libraries, community centers, restaurants, education programs, schools, or hardware/software donation programs.

12. Identify and discuss ways that access to technology helps empower individuals and groups.
    Examples: Gives access to information; provides the ability to communicate with others around the world; enables people to buy and sell things.

Global Collaborator

Communication

13. Communicate key ideas and details collaboratively in a way that informs, persuades, and/or entertains, using digital tools.
    Example: Create a digital presentation to persuade school administrators to allow additional time for lunch.

Digital Tools

14. Type 15 words per minute with 95% accuracy using appropriate keyboarding techniques.
15. Describe local, networked, and online or cloud environments.

Collaborative Research

16. Conduct basic keyword searches to produce valid, appropriate results, and evaluate results for accuracy, relevance, and appropriateness.
    Examples: Use search techniques, check for credibility and validity.

Computing Analyst

Data

17. Describe examples of data sets or databases from everyday life.
    Examples: Library catalogs, school records, telephone directories, or contact lists.

Systems

18. Identify a broad range of digital devices, the services they provide, and appropriate uses for them.
    Examples: Computers, smartphones, tablets, robots, e-textiles, driving directions apps that access remote map services, digital personal assistants that access remote information services.
19. Describe the differences between hardware and software.

Innovative Designer

Human/Computer Partnerships

20. Compare and contrast human and computer performance on similar tasks to understand which is better suited to the task.
    Examples: Sorting alphabetically, finding a path across a cluttered room.
21. Explain advantages and limitations of technology.
    Example: 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*.

Design Thinking

22. Discuss the design process and use digital tools to illustrate potential solutions.
23. Implement the design process to solve a simple problem.
    Examples: Uneven table leg, noise in the cafeteria, tallying the collection of food drive donations.

Grade 4 Overview

Grade 4 content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade level.

Content Standard Strands and Topics

Computational Thinker

- Abstraction
- Algorithms
- Programming and Development

Citizen of a Digital Culture

- Safety, Privacy, and Security
- Legal and Ethical Behavior
- Digital Identity
- Impact of Computing

Global Collaborator

- Communication
- Digital Tools
- Collaborative Research

Computing Analyst

- Data
- Systems

Innovative Designer

- Human/Computer Partnerships
- Design Thinking

Recurring Standards

Safety, Privacy, and Security
1. Identify, demonstrate, and apply personal safe use of digital devices.

Legal and Ethical Behavior
2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules.

Impact of Computing
3. Assess the validity and identify the purpose of digital content.

Systems
4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.

Collaborative Research
5. Locate and curate information from digital sources to answer research questions.

Digital Tools
6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Grade 4

Fourth graders will delve into more intricate processes of digital literacy and computer science through small group collaboration under the supervision and instruction of the teacher as a facilitator. Working with partners, students will identify and describe the different aspects of computational thinking and global collaboration using various devices.

Underlined words appear in the glossary.

Students can:

Computational Thinker

Abstraction

1. Construct a basic system of numbers, letters, or symbols to represent information as a cipher.
   Examples: Combine data from multiple sources, sorting multi-level.
2. Formulate a list of sub-problems to consider while addressing a larger problem.
   Examples: Problem - a multi-step math problem; sub-problem - steps to solve.
   Problem - light bulb does not light; sub-problem - steps to resolve why.

Algorithms

3. Show that different solutions exist for the same problem or sub-problem.
4. Detect and debug logical errors in various basic algorithms.
   Example: Trace the path of a set of directions to determine success or failure.
5. Use flowcharts to create a plan or algorithm.
6. Define a simple pseudocode.

Programming and Development

7. Create a working program in a block-based visual programming environment using arithmetic operators, conditionals, and repetition in programs, in collaboration with others.

Citizen of a Digital Culture

Safety, Privacy, and Security

8. Demonstrate the proper use and operation of security technologies.
   Examples: Passwords, virus protection software, spam filters, pop-up blockers.

Legal and Ethical Behavior

9. Identify laws and tools which help ensure that users of varying abilities can access electronic and information technology.
   Examples: ADA Laws

Digital Identity

10. Identify the different forms of web advertising and why websites, digital resources, and artifacts may include advertisements and collect personal information.
    Examples: Search ads, pay-per-click ads, banner ads, targeted ads, in-game ads, email ads.

Impact of Computing

11. Discuss the digital divide as unequal access to technology based on differences such as income, education, age, or geographic location and locate resources in the community that can give people access to technology.

Global Collaborator

Communication

12. Use basic features of digital tools to communicate key ideas and details in a way that informs and/or persuades.
13. Synthesize complex information from multiple sources in different ways to make it more useful and/or relevant.

Digital Tools

14. Type 20 words per minute with 95% accuracy using appropriate keyboarding techniques.

Collaborative Research

15. Conduct complex keyword searches to produce valid, appropriate results and evaluate results for accuracy, relevance, and appropriateness.
    Examples: Search techniques, check for credibility and validity.

Computing Analyst

Data

16. Gather and organize data to answer a question using a variety of computing and data visualization methods.
    Examples: Sorting, totaling, averaging, charts, and graphs.

Systems

17. Demonstrate an appropriate level of proficiency in performing tasks using a range of digital devices.
    Examples: Collect and record data, print, use send command, connect to Internet, or search; use probes, sensors, printers, robots, or computers.

Modeling and Simulation

18. Create a simple digital model of a system, individually and collaboratively, and explain what the model shows and does not show.
    Examples: Create a model of the water cycle and indicate that it shows how precipitation forms but does not indicate how pesticides get into rivers.
19. Use data from a simulation to answer a question collaboratively.

Innovative Designer

Human/Computer Partnerships

20. Explain how hardware and applications can enable everyone, including people with disabilities, to do things they could not do otherwise.
    Examples: Global Positioning System [GPS] to navigate, text-to-speech feature to read aloud from a digital resource, translate a digital resource to a different language.

Design Thinking

21. Develop, test, and refine prototypes as part of a cyclical design process to solve a simple problem.

Grade 5 Overview

Grade 5 content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade level.

Content Standard Strands and Topics

Computational Thinker

- Abstraction
- Algorithms
- Programming and Development

Citizen of a Digital Culture

- Safety, Privacy, and Security
- Legal and Ethical Behavior
- Digital Identity
- Impact of Computing

Global Collaborator

- Communication
- Digital Tools
- Collaborative Research

Computing Analyst

- Data
- Systems

Innovative Designer

- Human/Computer Partnerships
- Design Thinking

Recurring Standards

Safety, Privacy, and Security
1. Identify, demonstrate, and apply personal safe use of digital devices.

Legal and Ethical Behavior
2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules.

Impact of Computing
3. Assess the validity and identify the purpose of digital content.

Systems
4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.

Collaborative Research
5. Locate and curate information from digital sources to answer research questions.

Digital Tools
6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Grade 5

During fifth grade, students will progress toward independence while continuing to collaborate on local and global issues. Students learn to be creators, not only consumers, who can effectively utilize digital tools and understand the influence of technology. These standards are written to encourage student-centered learning through teacher facilitation and creative, hands-on activities.

Underlined words appear in the glossary.

Students can:

Computational Thinker

Abstraction

1. Construct a complex system of numbers or letters to represent information.
   Example: Student-created complex secret codes using more than one form to solve a problem or answer a question.

Algorithms

2. Create an algorithm to solve a problem while detecting and debugging logical errors within the algorithm.
   Examples: Program the movement of a character, robot, or person through a maze.
   Define a variable that can be changed or updated.
3. Create an algorithm that is defined by simple pseudocode.
4. Create a simple pseudocode.
5. Develop and recommend solutions to a given problem and explain the process to an audience.

Programming and Development

6. Create a working program in a block-based visual programming environment using arithmetic operators, conditionals, and repetition in programs.
7. Identify variables.
8. Demonstrate that programs require known starting values that may need to be updated appropriately during the execution of programs.
   Examples: Set initial value of a variable, updating variables.

Citizen of a Digital Culture

Safety, Privacy, and Security

9. Explain the proper use and operation of security technologies.
   Examples: Passwords, virus protection software, spam filters, pop-up blockers, cookies.
10. Identify appropriate and inappropriate uses of communication technology and discuss the permanence of actions in the digital world.

Legal and Ethical Behavior

11. Explain that laws and tools exist to help ensure that people of varying abilities can access electronic and information technology.
    Examples: Section 508, Telecommunication Act of 1996, Braille, closed captioning, text to speech.

Digital Identity

12. Explain the different forms of web advertising and why websites, digital resources, and artifacts may include advertisements that may collect personal information.
    Examples: personalized web experiences based on tailored web searches, maintaining search history, quicker access to relevant information.

Impact of Computing

13. Share knowledge of resources in the community that can give people access to technology.
    Example: student created print and/or digital resource to share WiFi or other connectivity opportunities within the community.
14. Analyze the impact of social media on individuals, families, and society.
15. Explore and predict how advances in computing technologies affect job opportunities and/or processes now and in the future.

Global Collaborator

Communication

16. Use advanced features of digital tools and media-rich resources to communicate key ideas and details in a way that informs, persuades, and/or entertains.
17. Publish organized information in different ways to make it more useful or relevant.
    Examples: Infographic, student created website.

Digital Tools

18. Type 25 words per minute with 95% accuracy using appropriate keyboarding techniques.

Collaborative Research

19. Conduct advanced keyword searches to produce valid, appropriate results and evaluate results for accuracy, relevance, and appropriateness.
    Examples: Search techniques, check for credibility and validity.

Social Interactions

20. Collaborate locally and globally using online digital tools under teacher supervision.

Computing Analyst

Data

21. Manipulate data to answer a question using a variety of computing methods and tools to collect, organize, graph, analyze, and publish the resulting information.

Systems

22. Identify computing services that may be initially turned on by default.
    Examples: Geolocations, geotagging.
23. Identify the key components of a network.
    Examples: Links, nodes, networking devices.
24. Describe the need for authentication of users and devices as it relates to access permissions, privacy, and security.
    Examples: Logging in at school, logging personal devices to public networks.

Modeling and Simulations

25. Analyze the concepts, features, and behaviors illustrated by a simulation.
    Examples: Object motion, weather, ecosystem, predator/prey.
26. Connect data from a simulation to real-life events.

Innovative Designer

Human/Computer Partnerships

27. Define social engineering and discuss possible defenses.
    Examples: Phishing, impersonating

Design Thinking

28. Develop, test, and refine prototypes as part of a cyclical design process to solve a complex problem.
    Examples: Design backpack for a specific user's needs; design a method to collect and transport water without the benefit of faucets; design boats that need to hold as much payload as possible before sinking; design models of chairs based on specific user needs.

Grades 6-8 Overview

Students in Grades 6-8 are developing more independence as they seek their places in an increasingly digital and global society. Many of these students will begin developing their global online presence for the first time. In these grades, students are becoming proficient digital citizens, while continuing to build on a strong foundation in computer science principles. The goals of the content strands at this level demonstrate this balance.

Sixth, seventh, and eighth grade students will meet the following learning goals:

- As ***Computational Thinkers***, students break problems into component parts, identify key pieces of information, and use that information to solve problems.
- As ***Citizens of a Digital Culture***, students verbalize the impact of computing in a global society while safely, securely, ethically, and legally interacting with digital environments and protecting their digital identities.
- As ***Global Collaborators***, students use appropriate digital tools to communicate data that informs, persuades, and entertains to collaborate with society locally and globally.
- As ***Computing Analysts***, students utilize computing systems efficiently in the management and interpretation of data and information.
- As ***Innovative Designers***, students leverage human and computer partnerships within a design process, creating useful and thoughtful solutions to problems.

The content standards for Grades 6-8 encourage analysis, synthesis, and evaluation in digital literacy and computer science as themes within all areas of the academic curriculum. Furthermore, students in Grades 6-8 will work collaboratively to explore, employ, and develop digital tools.

Grade 6 Overview

Grade 6 content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade level.

Content Standard Strands and Topics

Computational Thinker

- Abstraction
- Algorithms
- Programming and Development

Citizen of a Digital Culture

- Safety, Privacy, and Security
- Legal and Ethical Behavior
- Digital Identity
- Impact of Computing

Global Collaborator

- Communication
- Digital Tools
- Collaborative Research

Computing Analyst

- Data
- Systems

Innovative Designer

- Human/Computer Partnerships
- Design Thinking

Recurring Standards

Safety, Privacy, and Security
1. Identify, demonstrate, and apply personal safe use of digital devices.

Legal and Ethical Behavior
2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules.

Impact of Computing
3. Assess the validity and identify the purpose of digital content.

Systems
4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.

Collaborative Research
5. Locate and curate information from digital sources to answer research questions.

Digital Tools
6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Grade 6

During sixth grade, students will continue to develop the foundation of computer science. They will expand their problem-solving skills and progress toward independence while continuing to collaborate on local and global issues. Students must be creators, not just consumers, who can effectively utilize digital tools and understand the influence of technology. These standards are written to encourage student-centered learning through innovative and engaging activities.

Underlined words appear in the glossary.

Students can:

Computational Thinker

Abstraction

1. Remove background details from an everyday process to highlight essential properties.
   Examples: When making a sandwich, the type of bread, condiments, meats, and/or vegetables do not affect the fact that one is making a sandwich.
2. Define a process as a function.
   Example: Functions or sets of steps combined to produce a process: turning off your alarm + getting out of bed + brushing your teeth + getting dressed = morning routine.

Algorithms

3. Create pseudocode that uses conditionals.
   Examples: Using if/then/else (If it is raining then bring an umbrella else get wet).
4. Differentiate between flowcharts and pseudocode.
   Example: Flowcharts use shapes to indicate what to do at each step while pseudocode uses text.
5. Identify algorithms that make use of sequencing, selection or iteration.
   Examples: Sequencing is doing steps in order (put on socks, put on shoes, tie laces); selection uses a Boolean condition to determine which of two parts of an algorithm are used (hair is dirty? True, wash hair; false, do not); iteration is the repetition of part of an algorithm until a condition is met (if you're happy and you know it clap your hands, when you're no longer happy you stop clapping).

Programming and Development

6. Identify steps in developing solutions to complex problems using computational thinking.
7. Describe how automation works to increase efficiency.
   Example: Compare the amount of time/work to hand wash a car *vs*. using an automated car wash.
8. Create a program that initializes a variable.
   Example: Create a flowchart in which the variable or object returns to a starting position upon completion of a task.

Citizen of a Digital Culture

Safety, Privacy, and Security

9. Differentiate between a secure and a non-secure website including how they affect personal data.
   Example: HTTP *vs.* HTTPS.

Legal and Ethical Behavior

10. Describe the causes and effects of illegal use of intellectual property as it relates to print and digital media, considering copyright, fair use, licensing, sharing, and attribution.
11. Differentiate between appropriate and inappropriate digital content and the use of that content.

Digital Identity

12. Define digital permanence.
13. Define personal privacy, digital footprint, and open communication.

Impact of Computing

14. Discuss digital globalization and Internet censorship.
    Examples: Software that scans a website for posts about potential threats; a person's ability to order a product directly from a manufacturer in another part of the world; a student in Africa can take an online math course created in the United States; web-hosting company prevents posting of content.
15. Identify emerging technologies in computing.

Global Collaborator

Creative Communications

16. Communicate and/or publish collaboratively to inform others from a variety of backgrounds and cultures about issues and problems.

Digital Tools

17. Type 30 words per minute with 95% accuracy using appropriate keyboarding techniques.

Social Interactions

18. Define censorship.

Computing Analyst

Data

19. Track data change from a variety of sources.
    Example: Use editing or versioning tools to track changes to data.
20. Identify data transferring protocols, visualization, and the purpose of data and methods of storage.
    Examples: Using an online collection tool or form to collect data that is then stored in a spreadsheet or database.
21. Identify varying data structures/systems and methods of classification, including decimal and binary.
    Examples: Difference between a bit and a byte, bit representation, pixels.
22. Summarize the purpose of the American Standard Code for Information Interchange (ASCII).

Systems

23. Discuss how digital devices may be used to collect, analyze, and present information.
24. Compare and contrast types of networks.
    Examples: Wired, wireless (WiFi), local, wide area, mobile, Internet, and intranet.
25. Differentiate between secure and non-secure systems.

Modeling and Simulation

26. Explain why professionals may use models as logical representations of physical, mathematical, or logical systems or processes.
    Example: Students will discuss why an engineer may build a model of a building before actually constructing the building.
27. Explain how simulations serve to implement models.

Innovative Designer

Human/Computer Partnerships

28. Define assistive technologies and state reasons they may be needed.
29. Define artificial intelligence and identify examples of artificial intelligence in the community.
    Examples: Image recognition, voice assistants.

Design Thinking

30. Discuss and apply the components of the problem-solving process.
    Example: Students will devise a plan to alleviate traffic congestion around the school during drop-off and pick-up.

Grade 7 Overview

Grade 7 content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade level.

Content Standard Strands and Topics

Computational Thinker

- Abstraction
- Algorithms
- Programming and Development

Citizen of a Digital Culture

- Safety, Privacy, and Security
- Legal and Ethical Behavior
- Digital Identity
- Impact of Computing

Global Collaborator

- Communication
- Digital Tools
- Collaborative Research

Computing Analyst

- Data
- Systems

Innovative Designer

- Human/Computer Partnerships
- Design Thinking

Recurring Standards

Safety, Privacy, and Security
1. Identify, demonstrate, and apply personal safe use of digital devices.

Legal and Ethical Behavior
2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules.

Impact of Computing
3. Assess the validity and identify the purpose of digital content.

Systems
4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.

Collaborative Research
5. Locate and curate information from digital sources to answer research questions.

Digital Tools
6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Grade 7

During seventh grade, students will become independent thinkers while developing their global online presence. Students must be creators, not just consumers, who will effectively utilize digital tools, understand technology's impact on a global society, and integrate principles of computer science. These standards are written for student-centered learning with teacher mentoring.

Underlined words appear in the glossary.

Students can:

Computational Thinker

Abstraction

1. Create a function to simplify a task.
   Example: Get a writing utensil, get paper, jot notes can collectively be named "note taking".

Algorithms

2. Create complex pseudocode using conditionals and Boolean statements.
   Example: Automated vacuum pseudocode – drive forward until the unit encounters an obstacle; reverse 2"; rotate 30 degrees to the left, repeat.
3. Create algorithms that demonstrate sequencing, selection or iteration.
   Examples: Debit card transactions are approved until the account balance is insufficient to fund the transaction = iteration, do until.
4. Design a complex algorithm that contains sequencing, selection or iteration.
   Examples: Lunch line algorithm that contains parameters for bringing your lunch and multiple options available in the lunch line.

Programming and Development

5. Solve a complex problem using computational thinking.
6. Create and organize algorithms in order to automate a process efficiently.
   Example: Set of recipes (algorithms) for preparing a complete meal.
7. Create a program that updates the value of a variable in the program.
   Examples: Update the value of score when a coin is collected (in a flowchart, pseudocode or program).
8. Formulate a narrative for each step of a process and its intended result, given pseudocode or code.

Citizen of a Digital Culture

Safety, Privacy, and Security

9. Identify common methods of securing data.
   Examples: Permissions, encryption, vault, locked closet.

Legal and Ethical Behavior

10. Explain social engineering, including countermeasures, and its impact on a digital society.
    Examples: Phishing, hoaxes, impersonation, baiting, spoofing.
11. Demonstrate positive, safe, legal, and ethical habits when creating and sharing digital content and identify the consequences of failing to act responsibly.

Digital Identity

12. Discuss the impact of data permanence on digital identity including best practices to protect personal digital footprint.

Impact of Computing

13. Compare and contrast information available locally and globally.
    Example: Review an article published in the United States and compare to an article on the same subject published in China.
14. Discuss current events related to emerging technologies in computing and the effects such events have on individuals and the global society.
15. Discuss unique perspectives and needs of a global culture when developing computational artifacts, including options for accessibility for all users.
    Example: Would students create a webpage aimed at reaching a village of users that have no way access to the Internet?

Global Collaborator

Creative Communications

16. Construct content designed for specific audiences through an appropriate medium.
    Examples: Design a multi-media children's e-book with an appropriate readability level.
17. Publish content to be available for external feedback.

Digital Tools

18. Type 35 words per minute with 95% accuracy using appropriate keyboarding techniques.

Social Interactions

19. Discuss the benefits and limitations of censorship.
20. Evaluate the validity and accuracy of a data set.

Computing Analyst

Data

21. Compare common transfer protocols.
    Examples: FTP, HTTP
22. Compare data storage structures.
    Examples: Stack, array, queue, table, database.

Systems

23. Demonstrate the use of a variety of digital devices individually and collaboratively to collect, analyze, and present information for content-related problems.
24. Diagram a network given a specific setup or need.
    Examples: Home network, public network, business network.
25. List common methods of system cybersecurity.
    Examples: Various password requirements, two factor authentication, biometric, geolocation.

Modeling and Simulation

26. Categorize models based on the most appropriate representation of various systems.
27. Identify data needed to create a model or simulation of a given event.
    Examples: When creating a random name generator, the program needs access to a list of possible names.

Innovative Designer

Human/Computer Partnerships

28. Classify types of assistive technologies.
    Examples: Hardware, software, stylus, sticky keys.
29. Compare and contrast human intelligence and artificial intelligence.

Design Thinking

30. Apply the problem-solving process to solve real-world problems.

Grade 8 Overview

Grade 8 content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade level.

Content Standard Strands and Topics

Computational Thinker

- Abstraction
- Algorithms
- Programming and Development

Citizen of a Digital Culture

- Safety, Privacy, and Security
- Legal and Ethical Behavior
- Digital Identity
- Impact of Computing

Global Collaborator

- Communication
- Digital Tools
- Collaborative Research

Computing Analyst

- Data
- Systems

Innovative Designer

- Human/Computer Partnerships
- Design Thinking

Recurring Standards

Safety, Privacy, and Security
1. Identify, demonstrate, and apply personal safe use of digital devices.

Legal and Ethical Behavior
2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules.

Impact of Computing
3. Assess the validity and identify the purpose of digital content.

Systems
4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues.

Collaborative Research
5. Locate and curate information from digital sources to answer research questions.

Digital Tools
6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Grade 8

During eighth grade, students will expound upon computer science and global collaboration experiences. Students will be designers, not just consumers, who will effectively utilize digital tools and articulate the impact of technology on a global society. These standards are written to provide student-centered learning with minimal guidance from the teacher.

Underlined words appear in the glossary.

Students can:

Computational Thinker

Abstraction

1. Design a function using a programming language that demonstrates abstraction.
   Example: Create a program that utilizes functions in an effort remove repetitive sequences of steps.
2. Explain how abstraction is used in a given function.
   Example: Examine a set of block-based code and explain how abstraction was used.

Algorithms

3. Create an algorithm using a programming language that includes the use of sequencing, selections, or iterations.
   Example: Use a block-based or script programming language

```
Step 1: Start
Step 2: Declare variables a, b and c.
Step 3: Read variables a, b and c.
Step 4: If a>b
            If a>c
                    Display a is the largest number.
            Else
                    Display c is the largest number.
        Else
            If b>c
                    Display b is the largest number.
            Else
                    Display c is the greatest number.
Step 5: Stop
```

4. Create a function to simplify a task.
   Example: 3^8 = 3*3*3*3*3*3*3*3; =(Average) used in a spreadsheet to average a given list of grades.

Programming and Development

5. Discuss the efficiency of an algorithm or technology used to solve complex problems.
6. Describe how algorithmic processes and automation increase efficiency.
7. Create a program that includes selection, iteration, or abstraction, and initializes, and updates, at least two variables.

Examples: Make a game, interactive card, story, or adventure game.

Citizen of a Digital Culture

Safety, Privacy, and Security

8. Compare and contrast common methods of securing data.
9. Secure a file or other data.
   Examples: lock spreadsheet cell(s), password protect, encrypt.

Legal and Ethical Behavior

10. Analyze different modes of social engineering and their effectiveness.
    Examples: Phishing, hoaxes, impersonation, baiting, spoofing.
11. Advocate for positive, safe, legal, and ethical habits when creating and sharing digital content.
    Example: Students create a brochure that highlights the consequences of illegally downloading media.

Digital Identity

12. Cite evidence of the positive and negative effects of data permanence on personal and professional digital identity.

Impact of Computing

13. Evaluate the impact of digital globalization on public perception and ways Internet censorship can affect free and equitable access to information.
14. Analyze current events related to computing and their effects on education, the workplace, individuals, communities, and global society.
15. Critique computational artifacts, including options for accessibility for all users, with respect to the needs of a global culture.

Global Collaborator

Creative Communications

16. Present content designed for specific audiences through an appropriate medium.
    Example: Create and share a help video for a senior's center that provides tips for online safety.
17. Communicate and publish individually or collaboratively to persuade peers, experts, or community about issues and problems.

Digital Tools

18. Type 40 words per minute with 95% accuracy using appropriate keyboarding techniques.

Social Interactions

19. Critique the impacts of censorship as it impacts global society.
    Example: Create a presentation outlining the social implications of limiting access to web content by favoring or blocking particular products or websites.
20. Examine an artifact that demonstrates bias through distorting, exaggerating, or misrepresenting data and redesign it using factual, relevant, unbiased content to more accurately reflect the truth.

Computing Analyst

Data

21. Differentiate types of data storage and apply most efficient structure.
    Examples: Stack, array, queue, table, database.
22. Encrypt and decrypt various data.
    Example: Create and decipher a message sent in a secret code.

Systems

23. Design a digital artifact to propose a solution for a content-related problem.
    Example: Create a presentation outlining how to create a cost-efficient method to melt snow on roads during the winter.
24. Compare and contrast common methods of cybersecurity.
    Example: Discuss how password protections and encryption are similar and different.

Modeling and Simulation

25. Create a model that represents a system.
    Example: Food chain, supply and demand.
26. Create a simulation that tests a specific model.
    Examples: Demonstrate that pressure changes with temperature in a controlled environment; demonstrate that rocket design affects the height of a rocket's launch; demonstrate that the amount of water changes the height of a plant.

Innovative Designer

Human/Computer Partnerships

27. Analyze assistive technologies and how they improve the quality of life for users.
    Example: Research multiple speech to text technologies and write a persuasive essay in favor of one over another.
28. Develop a logical argument for and against artificial intelligence.
    Examples: Students debate the use of artificial intelligence in self-driving vehicles.
    Students write a persuasive essay to argue for or against digital personal assistants.

Design Thinking

29. Create an artifact to solve a problem using ideation and iteration in the problem-solving process.
    Examples: Create a public service announcement or design a computer program, game, or application.

Grades 9 – 12 Overview

Students in Grades 9-12 experience significant growth and development as they assume more complex responsibilities. They continue to develop unique personalities and begin to make important life decisions. In both school and community, high school students are strengthening and practicing the leadership and communication skills that facilitate entrance into adulthood. They continue to seek opportunities for realizing independence and individuality.

Grades 9-12 students have broadened their perspective regarding the importance of existing and developing technologies and have an understanding of the scope of technology in today's world. As students progress through their high school years, they are able to address a variety of problems on a range of topics in a logical manner. Technology offers students an efficient means for solving many types of problems.

Many students have opportunities to interact with people whose backgrounds are different from their own because of the cultural and ideological diversity of a technologically advanced global society. As the use of technology brings humankind closer together, concepts and skills utilizing digital literacy and computer science will assist students in becoming productive adults.

Grades 9-12 students will meet the following learning goals:

- As ***Computational Thinkers,*** students demonstrate how to simplify complex problems by developing algorithms that define the systematic processes.
- As ***Citizens of a Digital Culture,*** students demonstrate an understanding of concepts involving safety and security, responsible use of technology, and ways it can influence people through social interactions.
- As ***Global Collaborators,*** students utilize digital tools to collaborate and communicate with others to solve problems presented in today's technical world.
- As ***Computing Analysts,*** students analyze and create solutions to problems and challenges presented in the use of computer systems and data.
- As ***Innovative Designers,*** students make decisions and create solutions using the various digital tools available in today's technical environments.

Grades 9-12 Overview

Grades 9-12 content for digital literacy and computer science is organized into five strands of focused study outlined below in the column on the left and identified by bold print in shaded bars. Related content standards are grouped by topic below each strand.

The Recurring Standards for Digital Literacy and Computer Science are listed below in the column on the right. These recurring standards should be incorporated into classroom instruction at the appropriate level of rigor in each grade level.

Content Standard Strands and Topics	Recurring Standards
Computational Thinker Abstraction Algorithms Programming and Development **Citizen of a Digital Culture** Safety, Privacy, and Security Legal and Ethical Behavior Digital Identity Impact of Computing **Global Collaborator** Communication Digital Tools Collaborative Research **Computing Analyst** Data Systems **Innovative Designer** Human/Computer Partnerships Design Thinking	**Safety, Privacy, and Security** 1. Identify, demonstrate, and apply personal safe use of digital devices. **Legal and Ethical Behavior** 2. Recognize and demonstrate age-appropriate responsible use of digital devices and resources as outlined in school/district rules. **Impact of Computing** 3. Analyze the potential impact of computing. **Systems** 4. Identify and employ appropriate troubleshooting techniques used to solve computing or connectivity issues. **Collaborative Research** 5. Locate, curate, and evaluate information from digital sources to answer research questions. **Digital Tools** 6. Produce, review, and revise authentic artifacts that include multimedia using appropriate digital tools.

Underlined words appear in the glossary.

Students can:

Computational Thinker

Abstraction

1. Decompose problems into component parts, extract key information, and develop descriptive models to understand the levels of abstractions in complex systems.
2. Explain how computing systems are often integrated with other systems and embedded in ways that may not be apparent to the user.
   Examples: Millions of lines of code control the subsystems within an automobile (e.g., antilock braking systems, lane detection, and self-parking).

Algorithms

3. Differentiate between a generalized expression of an algorithm in pseudocode and its concrete implementation in a programming language.
   a. Explain that some algorithms do not lead to exact solutions in a reasonable amount of time and thus approximations are acceptable.
   b. Compare and contrast the difference between specific control structures such as sequential statements, conditional, iteration, and explain the benefits and drawbacks of choices made.
      Examples: Tradeoffs involving implementation, readability, and program performance.
   c. Distinguish when a problem solution requires decisions to be made among alternatives, such as selection constructs, or when a solution needs to be iteratively processed to arrive at a result, such as iterative "loop" constructs or recursion.
   d. Evaluate and select algorithms based on performance, reusability, and ease of implementation.
   e. Explain how more than one algorithm may solve the same problem and yet be characterized with different priorities.
      Examples: All self-driving cars have a common goal of taking a passenger to a designation but may have different priorities such as safety, speed, or conservation; web search engines have their own algorithms for search with their own priorities.
4. Use and adapt classic algorithms to solve computational problems.
   Examples: Sorting, searching, shortest path, and data compression.

Programming and Development

5. Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using current events.
6. Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects, with parameters, and which return a result.
7. Compare and contrast fundamental data structures and their uses.
   Examples: Strings, lists, arrays, stacks, queues.
8. Demonstrate code reuse by creating programming solutions using libraries and Application Programming Interfaces.
9. Demonstrate the ability to verify the correctness of a program.
   a. Develop and use a series of test cases to verify that a program performs according to its design specifications

b. Collaborate in a code review process to identify correctness, efficiency, scalability and readability of program code.

10. Resolve or debug errors encountered during testing using iterative design process.
    Examples: Test for infinite loops, check for bad input, check edge-cases.

Citizen of a Digital Culture

Safety, Privacy, and Security

11. Model and demonstrate behaviors that are safe, legal, and ethical while living, learning, and working in an interconnected digital world.
    a. Recognize user tracking methods and hazards.
       Examples: Cookies, WiFi packet sniffing.
    b. Understand how to apply techniques to mitigate effects of user tracking methods.
    c. Understand the ramifications of end-user license agreements and terms of service associated with granting rights to personal data and media to other entities.
    d. Explain the relationship between online privacy and personal security.
       Examples: Convenience and accessibility, data mining, digital marketing, online wallets, theft of personal information.
    e. Identify physical, legal, and ethical consequences of inappropriate digital behaviors.
       Examples: Cyberbullying/harassment, inappropriate sexual communications.
    f. Explain strategies to lessen the impact of negative digital behaviors and assess when to apply them.
12. Describe how sensitive data can be affected by malware and other attacks.
13. Compare various security measures of a computer system.
    Examples: Usability, security, portability, and scalability.
14. Compare ways to protect devices, software, and data.

Legal and Ethical Behavior

15. Explain the necessity for the school's Acceptable Use Policy.
16. Identify laws regarding the use of technology and their consequences and implications.
    Examples: Unmanned vehicles, net neutrality/common carriers, hacking, intellectual property, piracy, plagiarism.
17. Discuss the ethical ramifications of malicious hacking and its impact on society.
    Examples: Dissemination of privileged information, ransomware.
18. Explain the beneficial and harmful effects that intellectual property laws can have on innovation.

Digital Identity

19. Prove that digital identity is a reflection of persistent, publicly available artifacts.
20. Evaluate strategies to manage digital identity and reputation with awareness of the permanent impact of actions in a digital world.

Impact of Computing

21. Explain how technology facilitates the disruption of traditional institutions and services.
    Examples: Digital currencies, ridesharing, autonomous vehicles, retail, Internet of Things.
22. Research the impact of computing technology on possible career pathways.
    Examples: Government, business, medicine, entertainment, education, transportation.

23. Debate the positive and negative effects of computing innovations in personal, ethical, social, economic, and cultural spheres.
    Examples: Artificial Intelligence/machine learning, mobile applications, automation of traditional occupational skills.

Global Collaborator

Creative Communication

24. Compare and contrast Internet publishing platforms, including suitability for media types, target audience, and feedback mechanism.
    a. Apply version control capabilities within a digital tool to understand the importance of managing historical changes across suggestions made by a collaborative team.

Digital Tools

25. Utilize a variety of digital tools to create digital artifacts across content areas.

Collaborative Research

26. Use collaborative technologies to work with others including peers, experts, or community members to examine local, national, and global issues and problems from multiple viewpoints.

Social Interactions

27. Apply tools and methods for collaboration on a project to increase connectivity among people in different cultures and career fields.
    Examples: Collaborative documents, webinars, teleconferencing, and virtual fieldtrips

Computing Analyst

Data

28. Develop a model that reflects the methods, procedures and concepts used by computing devices in translating digital bits as real-world phenomena, such as print characters, sound, images, and video.
29. Summarize the role of compression and encryption in modifying the structure of digital artifacts and the varieties of information carried in the metadata of these artifacts.
30. Evaluate the tradeoffs involved in choosing methods for the organization of data elements and the location of data storage, including the advantages and disadvantages of networked computing.
    Examples: Client server, peer-to-peer, cloud computing.
31. Create interactive data visualizations using software tools to help others understand real-world phenomena.
32. Use data analysis tools and techniques to identify patterns in data representing complex systems.

Systems

33. Evaluate the scalability and reliability of networks by describing the relationship between routers, switches, servers, topology, packets, or addressing, as well as the issues that impact network functionality.
    Examples: Bandwidth, load, delay.
    a. Explain the purpose of Internet Protocol addresses and how domain names are resolved to IP addresses through a Domain Name System server.
    b. Understand the need for networking protocols and examples of common protocols.
       Examples: HTTP, SMTP, and FTP
34. Categorize the roles of operating system software.
35. Appraise the role of artificial intelligence in guiding software and physical systems.
    Examples: predictive modeling, self-driving cars.
36. Explain the tradeoffs when selecting and implementing cybersecurity recommendations.
    Examples: Two-factor authentication, password requirements, geolocation requirements.

Modeling and Simulation

37. Evaluate the ability of models and simulations to test and support the refinement of hypotheses.
    a. Create and utilize models and simulations to help formulate, test, and refine a hypothesis.
    b. Form a model of a hypothesis, testing the hypothesis by the collection and analysis of data generated by simulations.
       Examples: Science lab, robotics lab, manufacturing, space exploration.
    c. Explore situations where a flawed model provided an incorrect answer.

Innovative Designer

Human/Computer Partnerships

38. Systematically design and develop programs for broad audiences by incorporating feedback from users.
    Examples: Games, utilities, mobile applications.
39. Identify a problem that cannot be solved by either humans or machines alone and discuss a solution for it by decomposing the task into sub-problems suited for a human or machine to accomplish.
    Examples: Forecasting weather, piloting airplanes.

Design Thinking

40. Use an iterative design process, including learning from mistakes, to gain a better understanding of a problem domain.

Appendix A

ISTE Standards for Students

1. Empowered Learner

Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences. Students:

a. articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes.
b. build networks and customize their learning environments in ways that support the learning process.
c. use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.
d. understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2. Digital Citizen

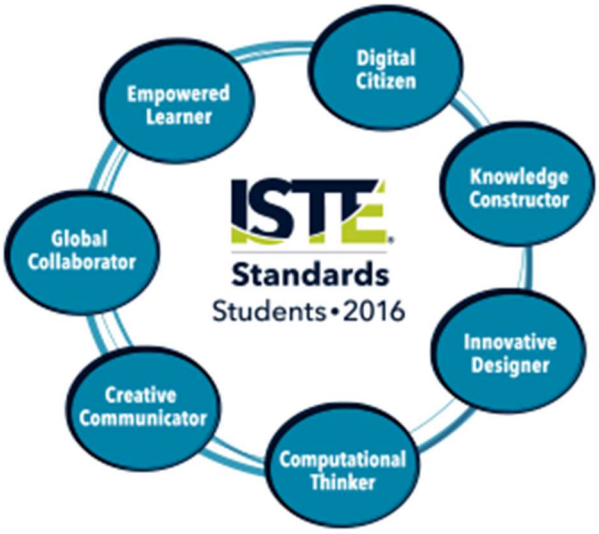
Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical. Students:

a. cultivate and manage their digital identity and reputation and are aware of the permanence of their actions in the digital world.
b. engage in positive, safe, legal and ethical behavior when using technology, including social interactions online or when using networked devices.
c. demonstrate an understanding of and respect for the rights and obligations of using and sharing intellectual property.
d. manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online.

3. Knowledge Constructor

Students critically curate a variety of resources using digital tools to construct knowledge, produce creative artifacts and make meaningful learning experiences for themselves and others. Students:

a. plan and employ effective research strategies to locate information and other resources for their intellectual or creative pursuits.
b. evaluate the accuracy, perspective, credibility and relevance of information, media, data or other resources.
c. curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions.
d. build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.



4. Innovative Designer

Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions. Students:

a. know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.
b. select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.
c. develop, test and refine prototypes as part of a cyclical design process.
d. exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.

5. Computational Thinker

Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions. Students:

a. formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.
b. collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.
c. break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
d. understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

6. Creative Communicator

Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals. Students:

a. choose the appropriate platforms and tools for meeting the desired objectives of their creation or communication.
b. create original works or responsibly repurpose or remix digital resources into new creations.
c. communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.
d. publish or present content that customizes the message and medium for their intended audiences.

7. Global Collaborator

Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally. Students:

a. use digital tools to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning.
b. use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.
c. contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.
d. explore local and global issues and use collaborative technologies to work with others to investigate solutions

Appendix B

Computer Science Teacher Association (CSTA) K-12 Computer Science Standards

Progression of Computer Science Teachers Association (CSTA) K-12 Computer Science Standards, Revised 2017

Concept	Subconcept	Level 1A (Ages 5-7) *By the end of Grade 2, students will be able to...*	Level 1B (Ages 8-11) *By the end of Grade 5, students will be able to...*	Level 2 (Ages 11-14) *By the end of Grade 8, students will be able to...*	Level 3A (Ages 14-16) *By the end of Grade 10, students will be able to...*
Computing Systems	Devices	**1A-CS-01** Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use. *(P1.1)*	**1B-CS-01** Describe how internal and external parts of computing devices function to form a system. *(P7.2)*	**2-CS-01** Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. *(P3.3)*	**3A-CS-01** Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. *(P4.1)*
Computing Systems	Hardware & Software	**1A-CS-02** Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). *(P7.2)*	**1B-CS-02** Model how computer hardware and software work together as a system to accomplish tasks. *(P4.4)*	**2-CS-02** Design projects that combine hardware and software components to collect and exchange data. *(P5.1)*	**3A-CS-02** Compare levels of abstraction and interactions between application software, system software, and hardware layers. *(P4.1)*
Computing Systems	Troubleshooting	**1A-CS-03** Describe basic hardware and software problems using accurate terminology. *(P6.2, P7.2)*	**1B-CS-03** Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. *(P6.2)*	**2-CS-03** Systematically identify and fix problems with computing devices and their components. *(P6.2)*	**3A-CS-03** Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. *(P6.2)*
Networks & The Internet	Network Communication & Organization		**1B-NI-04** Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination. *(P4.4)*	**2-NI-04** Model the role of protocols in transmitting data across networks and the Internet. *(P4.4)*	**3A-NI-04** Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing. *(P4.1)*
Networks & The Internet	Cybersecurity	**1A-NI-04** Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access. *(P7.3)*	**1B-NI-05** Discuss real-world cybersecurity problems and how personal information can be protected. *(P3.1)*	**2-NI-05** Explain how physical and digital security measures protect electronic information. *(P7.2)*	**3A-NI-05** Give examples to illustrate how sensitive data can be affected by malware and other attacks. *(P7.2)*
				2-NI-06 Apply multiple methods of encryption to model the secure transmission of information. *(P4.4)*	**3A-NI-06** Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. *(P3.3)*
					3A-NI-07 Compare various security measures, considering tradeoffs between the usability and security of a computing system. *(P6.3)*
					3A-NI-08 Explain tradeoffs when selecting and implementing cybersecurity recommendations. *(P7.2)*
Data & Analysis	Storage	**1A-DA-05** Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. *(P4.2)*	*Continuation of standard 1A-DA-05*	**2-DA-07** Represent data using multiple encoding schemes. *(P4.0)*	**3A-DA-09** Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. *(P4.1)*
					3A-DA-10 Evaluate the tradeoffs in how data elements are organized and where data is stored. *(P3.3)*
Data & Analysis	Collection, Visualization, & Transformation	**1A-DA-06** Collect and present the same data in various visual formats. *(P7.1, P4.4)*	**1B-DA-06** Organize and present collected data visually to highlight relationships and support a claim. *(P7.1)*	**2-DA-08** Collect data using computational tools and transform the data to make it more useful and reliable. *(P6.3)*	**3A-DA-11** Create interactive data visualizations using software tools to help others better understand real-world phenomena. *(P4.4)*
Data & Analysis	Inference & Models	**1A-DA-07** Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. *(P4.1)*	**1B-DA-07** Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea. *(P7.1)*	**2-DA-09** Refine computational models based on the data they have generated. *(P5.3, P4.4)*	**3A-DA-12** Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. *(P4.4)*
Algorithms & Programming	Algorithms	**1A-AP-08** Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. *(P4.4)*	**1B-AP-08** Compare and refine multiple algorithms for the same task and determine which is the most appropriate. *(P6.3, P3.3)*	**2-AP-10** Use flowcharts and/or pseudocode to address complex problems as algorithms. *(P4.4, P4.1)*	**3A-AP-13** Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. *(P5.2)*
Algorithms & Programming	Variables	**1A-AP-09** Model the way programs store and manipulate data by using numbers or other symbols to represent information. *(P4.4)*	**1B-AP-09** Create programs that use variables to store and modify data. *(P5.2)*	**2-AP-11** Create clearly named variables that represent different data types and perform operations on their values. *(P5.1, P5.2)*	**3A-AP-14** Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. *(P4.1)*
Algorithms & Programming	Control	**1A-AP-10** Develop programs with sequences and simple loops, to express ideas or address a problem. *(P5.2)*	**1B-AP-10** Create programs that include sequences, events, loops, and conditionals. *(P5.2)*	**2-AP-12** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. *(P5.1, P5.2)*	**3A-AP-15** Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. *(P5.2)*
					3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. *(P5.2)*
	Practices	*P1. Fostering an Inclusive Computing Culture* *P2. Collaborating Around Computing*	*P3. Recognizing and Defining Computational Problems* *P4. Developing and Using Abstractions*	*P5. Creating Computational Artifacts* *P6. Testing and Refining Computational Artifacts*	*P7. Communicating About Computing*

Progression of Computer Science Teachers Association (CSTA) K-12 Computer Science Standards, Revised 2017

Concept	Subconcept	Level 1A (Ages 5-7) *By the end of Grade 2, students will be able to...*	Level 1B (Ages 8-11) *By the end of Grade 5, students will be able to...*	Level 2 (Ages 11-14) *By the end of Grade 8, students will be able to...*	Level 3A (Ages 14-16) *By the end of Grade 10, students will be able to...*
Algorithms & Programming *(continued)*	Modularity	**1A-AP-11** Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. *(P3.2)*	**1B-AP-11** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. *(P3.2)*	**2-AP-13** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. *(P3.2)*	**3A-AP-17** Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. *(P3.2)*
			1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. *(P5.3)*	**2-AP-14** Create procedures with parameters to organize code and make it easier to reuse. *(P4.1, P4.3)*	**3A-AP-18** Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. *(P5.2)*
	Program Development	**1A-AP-12** Develop plans that describe a program's sequence of events, goals, and expected outcomes. *(P5.1, P7.2)*	**1B-AP-13** Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. *(P1.1, P5.1)*	**2-AP-15** Seek and incorporate feedback from team members and users to refine a solution that meets user needs. *(P2.3, P1.1)*	**3A-AP-19** Systematically design and develop programs for broad audiences by incorporating feedback from users. *(P5.1)*
		1A-AP-13 Give attribution when using the ideas and creations of others while developing programs. *(P7.3)*	**1B-AP-14** Observe intellectual property rights and give appropriate attribution when creating or remixing programs. *(P7.3)*	**2-AP-16** Incorporate existing code, media, and libraries into original programs, and give attribution. *(P4.2, P5.2, P7.3)*	**3A-AP-20** Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. *(P7.3)*
		1A-AP-14 Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops. *(P6.2)*	**1B-AP-15** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. *(P6.1, P6.2)*	**2-AP-17** Systematically test and refine programs using a range of test cases. *(P6.1)*	**3A-AP-21** Evaluate and refine computational artifacts to make them more usable and accessible. *(P6.3)*
			1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. *(P2.2)*	**2-AP-18** Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts. *(P2.2)*	**3A-AP-22** Design and develop computational artifacts working in team roles using collaborative tools. *(P2.4)*
		1A-AP-15 Using correct terminology, describe steps taken and choices made during the iterative process of program development. *(P7.2)*	**1B-AP-17** Describe choices made during program development using code comments, presentations, and demonstrations. *(P7.2)*	**2-AP-19** Document programs in order to make them easier to follow, test, and debug. *(P7.2)*	**3A-AP-23** Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. *(P7.2)*
Impacts of Computing	Culture	**1A-IC-16** Compare how people live and work before and after the implementation or adoption of new computing technology. *(P7.0)*	**1B-IC-18** Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices. *(P7.1)*	**2-IC-20** Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options. *(P7.2)*	**3A-IC-24** Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. *(P1.2)*
			1B-IC-19 Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. *(P1.2)*	**2-IC-21** Discuss issues of bias and accessibility in the design of existing technologies. *(P1.2)*	**3A-IC-25** Test and refine computational artifacts to reduce bias and equity deficits. *(P1.2)*
					3A-IC-26 Demonstrate ways a given algorithm applies to problems across disciplines. *(P3.1)*
	Social Interactions	**1A-IC-17** Work respectfully and responsibly with others online. *(P2.1)*	**1B-IC-20** Seek diverse perspectives for the purpose of improving computational artifacts. *(P1.1)*	**2-IC-22** Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact. *(P2.4, P5.2)*	**3A-IC-27** Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields. *(P2.4)*
	Safety, Law, & Ethics		**1B-IC-21** Use public domain or creative commons media, and refrain from copying or using material created by others without permission. *(P7.3)*		**3A-IC-28** Explain the beneficial and harmful effects that intellectual property laws can have on innovation. *(P7.3)*
		1A-IC-18 Keep login information private, and log off of devices appropriately. *(P7.3)*		**2-IC-23** Describe tradeoffs between allowing information to be public and keeping information private and secure. *(P7.2)*	**3A-IC-29** Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users. *(P7.2)*
					3A-IC-30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics. *(P7.3)*
	Practices	*P1. Fostering an Inclusive Computing Culture* *P2. Collaborating Around Computing*	*P3. Recognizing and Defining Computational Problems* *P4. Developing and Using Abstractions*	*P5. Creating Computational Artifacts* *P6. Testing and Refining Computational Artifacts*	*P7. Communicating About Computing*

Suggested Citation: Computer Science Teachers Association (2017). CSTA K-12 Computer Science Standards, Revised 2017. Retrieved from http://www.csteachers.org/standards.
The K–12 Computer Science Framework, led by the Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative in partnership with states and districts, informed the development of this work. View the framework at http://k12cs.org.

Appendix C

K12 Computer Science Framework

The framework is a high-level guide describing Computer Science (CS) for all students. The framework will empower students to be informed citizens who can critically engage in CS-related discussions; develop as learners, users, and creators of CS knowledge and artifacts; better understand the role of computing in the world around them; and learn, perform, and express themselves in other subjects and interests. At its center are core concepts and practices. View the framework at https://k12cs.org/navigating-the-practices/.

LITERACY STANDARDS FOR GRADES 6-12:

HISTORY/SOCIAL STUDIES, SCIENCE, AND TECHNICAL SUBJECTS

College and Career Readiness Anchor Standards for Reading

The Grades 6-12 standards on the following pages define what students should understand and be able to do by the end of each grade span. They correspond to the College and Career Readiness (CCR) anchor standards below by number. The CCR and grade-specific standards are necessary complements—the former providing broad standards, the latter providing additional specificity—that together define the skills and understandings that all students must demonstrate.

Key Ideas and Details

1. Read closely to determine what the text says explicitly and to make logical inferences from it; cite specific textual evidence when writing or speaking to support conclusions drawn from the text.
2. Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.
3. Analyze how and why individuals, events, or ideas develop and interact over the course of a text.

Craft and Structure

4. Interpret words and phrases as they are used in a text, including determining technical, connotative, and figurative meanings, and analyze how specific word choices shape meaning or tone.
5. Analyze the structure of texts, including how specific sentences, paragraphs, and larger portions of the text (e.g., a section, chapter, scene, or stanza) relate to each other and the whole.
6. Assess how point of view or purpose shapes the content and style of a text.

Integration of Knowledge and Ideas

7. Integrate and evaluate content presented in diverse formats and media, including visually and quantitatively, as well as in words. *
8. Delineate and evaluate the argument and specific claims in a text, including the validity of the reasoning as well as the relevance and sufficiency of the evidence.
9. Analyze how two or more texts address similar themes or topics in order to build knowledge or to compare the approaches the authors take.

Range of Reading and Level of Text Complexity

10. Read and comprehend complex literary and informational texts independently and proficiently.

*See College and Career Readiness Anchor Standards for Writing, "Research to Build and Present Knowledge," on page 60 for additional standards relevant to gathering, assessing, and applying information from print and digital sources.

Appendix D

College and Career Readiness Anchor Standards for Writing

The Grades 6-12 standards on the following pages define what students should understand and be able to do by the end of each grade span. They correspond to the College and Career Readiness (CCR) anchor standards below by number. The CCR and grade-specific standards are necessary complements—the former providing broad standards, the latter providing additional specificity—that together define the skills and understandings that all students must demonstrate.

Text Types and Purposes*

1. Write arguments to support claims in an analysis of substantive topics or texts using valid reasoning and relevant and sufficient evidence.
2. Write informative/explanatory texts to examine and convey complex ideas and information clearly and accurately through the effective selection, organization, and analysis of content.
3. Write narratives to develop real or imagined experiences or events using effective technique, well-chosen details, and well-structured event sequences.

Production and Distribution of Writing

4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
5. Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach.
6. Use technology, including the Internet, to produce and publish writing and to interact and collaborate with others.

Research to Build and Present Knowledge

7. Conduct short as well as more sustained research projects based on focused questions, demonstrating understanding of the subject under investigation.
8. Gather relevant information from multiple print and digital sources, assess the credibility and accuracy of each source, and integrate the information while avoiding plagiarism.
9. Draw evidence from literary or informational texts to support analysis, reflection, and research.

Range of Writing

10. Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of tasks, purposes, and audiences.

*These broad types of writing include many subgenres

Bibliography

Alabama Course of Study: Digital Literacy and Computer Science. Alabama State Department of Education, 2018. www.alsde.edu/sec/sct/COS/Final%202018%20Digital%20Literacy%20and%20Computer%20Science%20COS[5206].pdf

"CS Discoveries 2018: Vocab." *Code.org,* 2017, curriculum.code.org/csd/vocab/. Accessed 3 January 2019.

"CSTA K-12 Computer Science Standards." *Computer Science Teachers Association,* 2017, www.csteachers.org/standards. Accessed 4 January 2019.

"Glossary: CS Fundamentals." *Code.org*, 2017, code.org/curriculum/docs/k- 5/glossary. Accessed 4 January 2019.*Interactive Design Foundation.* The Interactive Design Foundation, 2016, www.interaction-design.org/literature/article/5-stages- in-the-design-thinking-process. Accessed 3 January 2019.

"ISTE Standards for Students." *ISTE (International Society for Technology in Education),* 2016, www.iste.org/standards/for-students. Accessed 4 January 2019.

K-12 Computer Science Framework. Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative, k12cs.org. Accessed 4 January 2019.

Merriam-Webster. Merriam Webster, 2019. www.merriam-webster.com/. Accessed 3 January 2019.

"Pseudocode." *Whatis.com*, 2019, whatis.techtarget.com/search/query?q=pseudocode. Accessed 4 January 2019.

Glossary

Abstraction – The process of withdrawing or removing details to highlight essential properties (e.g., the color of a pixel represented at one level as three bytes (red, green, and blue), or at a lower level as a sequence of bits). The act of representing essential features without including the background details or explanations.

Acceptable Use/Usage Policy (AUP) – A document defining constraints and practices that a user must agree to for access to an organization's network and/or the Internet. Many organizations require that employees or students sign an acceptable use policy before being granted a access to the network.

Addressing – Unique identification of components on a network.

Algorithm – A list of steps to finish a task; a set of instructions that can be performed with or without a computer.

Application Programming Interface (API) – Code that allows two software programs communicate with one another; an external software library that provides a collection of features (implemented as functions or methods) that offer reusable functionality in a program (e.g., OpenGL is an external library that provides an API that can be used for programming of computer graphics).

Array – An indexable collection of values (e.g., a row, a column, or a collection of integers representing student grades).

Artificial Intelligence (AI) – The capability of a machine to imitate intelligent human behavior.

ASCII (American Standard Code for Information Interchange) – A character set for representing numbers, punctuation, and letters in the English alphabet (e.g., the number 65 corresponds to the letter "A").

Assistive technologies – Any item, piece of equipment, or product system that is used to increase, maintain, or improve functional capabilities of a person with a disability.

Attribution – The ascribing of a work (as of literature or art) to a particular author or artist.

Automation – The use of a combination of mechanical and computer-based control systems to perform actions without the need of human oversight.

Bandwidth – The bit-rate measure of the transmission capacity over a network communication system; or, the carrying capacity of a channel or the data transfer speed of that channel.

Binary – A discrete numbering system that can represent information using only two options (0 or 1, on or off, yes or no, true or false) to allow for digital processing.

Biometric – The process by which a person's unique physical and other traits are detected and recorded by an electronic device or system as a means of confirming identity.

Bit – A contraction of "**B**inary dig**IT**." A bit is the single unit of information in a computer, typically represented as a 0 or 1.

Block-based programming language – Any predefined code that lets users create programs by manipulating "blocks" or graphical programming elements, rather than writing code using text with specific syntax rules, sometimes called visual coding, drag-and-drop programming, or graphical programming blocks.

Boolean – A variable data type or expression that can be set to either true or false.

Bug – An error in a program that prevents the program from running as expected.

Byte – A contraction of "**B**inar**Y** **TE**rm." A group of bits, usually eight, processed as a single unit of data.

Censorship – The act of examining media for the purpose of suppressing parts deemed objectionable on moral, political, military, or other grounds.

Cipher – A method of transforming a text in order to conceal its meaning; or, a coded message that requires a key to decode. The cipher cannot be decoded without the key.

Citation – The attribution of a reference to a book, paper, author, or other item in fixed, tangible form, especially in a scholarly work.

Client-server computing – A distributed network architecture with shared responsibilities between servers (computers that provides resources and services) and clients (those that request the resources/services).

Cloud-based computing – Applications, services, or resources made available to users on demand via the Internet from a remote computing provider's server.

Code – One or more command(s) or algorithm(s) designed to be carried out by a computer using a programming language. See also: **Program**

Comment – A note in the source code of a computer program that helps explain the code to those who read it.

Command – An instruction given by a user telling a computer to do something, such as run a single program or a group of linked programs.

Compression – The process of reducing the size of a computational artifact (e.g., a file) using a digital tool which implements an algorithm that recognizes repetitive data and removes redundancy across the artifact.

Computer science – The study of computers and algorithmic processes including their principles, hardware and software designs, applications, networks, and impact on society.

Computer system – A collection of one or more computers or computing devices, together with their hardware and software, integrated for the purpose of accomplishing shared tasks. Although a computer system can be limited to a single computer or computing device, it more commonly refers to a collection of multiple, connected computers, computing devices, and hardware.

Computational artifact – Something created by a human using a computer which can be, but is not limited to, a program, an image, an audio clip, a video, a presentation, or a Web page file. The computational artifact could solve a problem, show creative expression, or provide a viewer with new insight or knowledge.

Computational thinking – a problem-solving process that includes (but is not limited to) the following characteristics: Formulating problems in a way that enables us to use a computer and other tools to help solve them; Logically organizing and analyzing data; Representing data through abstractions such as models and simulations; Automating solutions through algorithmic thinking (a series of ordered steps); Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources; Generalizing and transferring this problem solving process to a wide variety of problems.

Conditionals/Compound conditionals ("If" statements) – Statements that run only when certain conditions exist. Often called a selection or "if" statement in a programming language, represented as an expression that evaluates to a Boolean value.

Control – The power to direct the course of actions. In programming, the use of elements of programming code to direct which actions take place and the order in which they take place. A programming (code) structure that implements control. Selection ("if" statements) and loops are examples of control structures.

Cookies – A small file or part of a file stored on a user's computer, created and subsequently read by a website server, and containing personal information (such as a user identification code, customized preferences, or a record of pages visited).

Copyright – A legal right created by the law of a country that grants the creator of an original work exclusive right for its use and distribution.

Creative Commons – A collection of public copyright licenses that enable the free distribution of an otherwise copyrighted work, used when an author wants to give people the right to share, use, and build upon a work that they have created.

Cyberbullying – Using electronic communication to intimidate, humiliate, or threaten another person.

Cybersecurity – The protection against access to, or alteration of, computing resources, through the use of technology, processes, and training.

Data – Raw facts that are collected and used for reference or analysis. Data can be digital or nondigital and can be in many forms, including numbers, text, show of hands, images, sounds, or video. Information used as a basis for reasoning, discussion, or calculation.

Data mining – The process of traversing through large data sets to identify patterns and establish relationships in order to solve problems through data analysis.

Database – A collection of data organized for search and retrieval.

Debug – To find and fix errors in an algorithm or program.

Decomposition (Decomposing) – The process of separating a whole into related parts or elements.

Decryption – The process of taking encoded or encrypted text or other data and converting it back into text (often called plaintext) that a human or computer can read and understand.

Design thinking – A methodology used to solve complex problems and find desirable solutions using logic, imagination, intuition, and systemic reasoning to explore possibilities of what could be and to create desired outcomes that benefit the end user.

Digital identity – Information on an entity used by computer systems to represent an external person, organization, application, or device.

Digital footprint – The collected information about an individual across multiple digital sources.

Digital globalization – The unrestricted flow of electronic data and products.

Digital literacy – The ability to use information and communication technologies to find, evaluate, create, and communicate information. Digital Literacy requires both cognitive and technical skills.

Digital permanence – The history and development of digital storage techniques, specifically quantifying the expected lifetime of data stored on various digital media and the factors which influence the perpetuity of digital data.

Domain Name System (DNS) server – An Internet service that translates a domain name to the correct IP address (e.g., amazon.com to 72.21.215.90).

Emerging technology – A new technology that is currently being developed, or will be developed in the near future.

Encryption – The process of converting information or data into a text (often called ciphertext) that is not readable by a human.

Equitable access – Robust and reliable access to current and emerging technologies and digital resources, with connectivity for all.

Extraction – The process of retrieving relevant information from data sources (like a database) in a specific pattern.

File Transfer Protocol (FTP) – A network protocol that provides the capability to upload and download data within a network.

Flaming – The use of digital emotional abuse to invoke certain emotions and responses such as rage, sadness, humiliation, self-doubt and more.

Flowchart – A diagram of the sequence of movements or actions of people or things involved in a complex system or activity.

Function – A named piece of code that can be called over and over again, sometimes called *procedures* or *methods*; a segment of code that includes the steps performed in a specified process.

Geolocation – The process or technique of identifying the physical location of a person or device by means of digital information processed via the Internet.

Hacking – Using a computer to create or explore some new idea; more commonly, to gain illegal access to a computer.

Hexadecimal – A number system with a base of 16, compared to the decimal number system (base 10) or binary number system (base 2), which uses the letters A through F to represent 10 through 15 in decimal (e.g., hexadecimal 10 is equal to 16 in decimal).

Hierarchical classification – A classification system where entries are arranged based on some order of rank structure.

Hyperlink – A link from an HTML file to another location or file, typically activated by clicking on a highlighted word or image on the screen.

Hypertext Transfer Protocol (HTTP) – An Internet protocol that controls the transfer of web data over the Internet.

Hypertext Transfer Protocol Secure (HTTPS) – An Internet protocol used to transmit web data securely via encryption.

Information – Data that has been processed into a useful form.

Ideation – The capacity for forming ideas.

Infographic – A visual image or group of images combined with simple text to represent complex data in a simplified way.

Initialize – To set something (such as a computer program counter) to a starting position, value, or configuration.

Input – A device or component that allows information to be received by a computer.

Intellectual property – A work or invention that is the result of creativity, such as a piece of writing or a design, which one owns and for which one may apply for a patent, copyright, or trademark.

Internet of Things (IoT) – The ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between objects and other Internet-enabled devices and systems.

Internet Protocol (IP) address – A collection of numbers (often four sets of numbers separated by a period, such as 72.21.215.90) that is used to uniquely identify each device connected to a computer network, such as the Internet.

Iteration (Loop) – A repetitive action or command typically created with programming loops. Loop is the action of doing something over and over again.

Loop - See Iteration; Control

Keyword – Main or significant term used to search the Internet for content; also used to represent the words that comprise a computer programming language.

Malware – Software designed to negatively impact a computer's normal functioning.

Metadata – Information that describes other information, such as descriptive data, organizational descriptions, and procedural information regarding the creation and technical specifications of the information. For example, an image file may contain metadata regarding the time and place of the picture creation as well as technical specifications about the camera used.

Multimedia – Using, involving, or encompassing several media (e.g., images, sound, video).

Netiquette – (Inter**NET** et**IQUETTE**). The correct or acceptable way of communicating on the Internet.

Net neutrality – The principle that Internet service providers should enable access to all content and applications regardless of the sources, and without favoring or blocking particular products or websites.

Network – An interconnected system of computers, peripherals, terminals, and databases connected by communication lines.

Operating system – Software that controls the operation of a computer and directs the processing of programs.

Output – Any device or component that transmits information from a computer.

Packet – A unit of data routed between an origin and a destination on a network.

Packet sniffing – Intercepting and reading the information within network packets that are sent as plaintext (not encrypted).

Parameter – a numerical or other measurable factor forming one of a set that defines a system or sets the conditions of its operation.

Password – A string of characters used for authentication to prove identity in order to approve access.

Peer-to-peer computing – Two or more computers with similar access privileges and responsibilities that communicate directly in order to share resources and services, rather than going through an intermediate server.

Prototype –A first full-scale and usually functional form of a new design.

Perseverance – Continued effort to do or achieve something despite difficulties, failure, or opposition.

Personal security – Actions which reduce the risk of threats and protect the user from hurtful disruptions in the patterns of daily life.

Phishing – The fraudulent practice of sending emails purporting to be from reputable companies in order to induce individuals to reveal personal information, such as passwords and credit card numbers.

Pixel – A contraction of "**PIC**ture **EL**ement." Any of the small, discrete elements that together constitute an image (as on a television or computer screen).

Portability – The ability of a user to export data, information, or software entered into or created by a software application or computing platform so it may be used in other applications or platforms.

Predictive modeling – A process that uses data mining and probability to forecast outcomes.

Problem domain – The area of expertise or application that is being explored to the exclusion of all other topics.

Problem-solving process – Problem definition > Problem Analysis > Generate possible solutions > Analyze solutions > Selecting appropriate solution(s) > Evaluate solution.

Program – An algorithm that has been coded into a form that can be run by a machine.

Programming (Coding) – The art of envisioning, designing, and implementing a computer program using some computational language.

Programming language – A vocabulary and set of syntax rules for instructing a computer or computing device to perform specific tasks.

Pseudocode – A detailed yet readable description of what a computer program or algorithm must do, expressed in a formally-styled natural language rather than programming language. (See Conditionals)

Queue – A data structure that consists of a list of records arranged so that records are added at one end and removed from the other. Sometimes described as First In First Out (FIFO).

Ransomware – A type of malware designed to block access to data until a specific demand (usually financial) is met.

Recurring standards – Key practices or concepts that appear at grade levels along the K - 12 continuum with progressive complexity. Rather than repeating these standards at multiple grade levels in this document, they are included on the Digital Literacy and Computer Science Course of Study Recurring Standards pages.

Router – Device or software that determines the path that data packets travel from source to destination.

Scalability – The ability of a process to maintain functionality and integrity as the scope and size of the process increases.

Selection – Using a Boolean condition to determine which of two parts of an algorithm is used.

Sequencing – Performing tasks in logical order.

Server – Computers on a network that provide access to resources and/or services to clients.

Simple Mail Transfer Protocol (SMTP) – An Internet standard that defines the protocol for sending electronic mail.

Simulation –The production of a digital model of something, especially for the purpose of study.

Social engineering – The management and manipulation of human beings in accordance with their place and function in society.

Software piracy – The illegal copying, distribution, or use of software.

Spoofing – To deceive or hoax; often used to indicate identity deception of email or network addresses.

Stack – A memory or a section of memory in a computer for temporary storage in which the last item stored is the first retrieved. Sometimes described as Last In First Out (LIFO).

Switch – A high-speed device that receives incoming data packets and redirects them to their destination on a local area network (LAN).

Syntax – An arrangement of items that abides by a prescribed set of rules.

System – A set of connected things or parts forming a complex whole, in particular; a set of principles or procedures according to which something is done; an organized scheme or method systems.

Table – A collection of arrays, which create rows and columns for ready reference; a database is represented by a collection of tables that are used to store information.

Topology – A description of the manner in which various network entities (computers, routers, printers, etc.) are arranged and connected.

Two-factor authentication – A security mechanism that requires two types of credentials for authentication and is designed to provide an additional layer of validation, minimizing security breaches (e.g., a password, and a text message confirmation).

Unplugged activity – An activity that teaches the fundamentals of computer science without digital tools, using physical implements such as card activities, strings, crayons, or puzzles.

Variable – An element, feature, or factor that is liable to change; in a programming language, a symbolic representation of some state or property of the program.

Version control – The consistent management of historical changes made to a digital artifact by collaborators.

Virus – Programming code that has been written to cause corruption of data on a computer; often attached to an executable file that spreads from one file to another once the program is executed.

Website – A collection of interlinked web pages on the World Wide Web.